

# An Experimental Comparison of Bayesian Optimization for Bipedal Locomotion

Roberto Calandra<sup>1</sup>, André Seyfarth<sup>2</sup>, Jan Peters<sup>1,3</sup> and Marc Peter Deisenroth<sup>4,1</sup>

**Abstract**—The design of gaits and corresponding control policies for bipedal walkers is a key challenge in robot locomotion. Even when a viable controller parametrization already exists, finding near-optimal parameters can be daunting. The use of automatic gait optimization methods greatly reduces the need for human expertise and time-consuming design processes. Many different approaches to automatic gait optimization have been suggested to date. However, no extensive comparison among them has yet been performed. In this paper, we present some common methods for automatic gait optimization in bipedal locomotion, and analyze their strengths and weaknesses. We experimentally evaluated these gait optimization methods on a bipedal robot, in more than 1800 experimental evaluations. In particular, we analyzed Bayesian optimization in different configurations, including various acquisition functions.

## I. INTRODUCTION

Bipedal locomotion is not robust, fast or energetically efficient [11] compared to wheeled or quadrupedal locomotion [25]. However, bipedal locomotion in robotics possesses various advantages: 1) high versatility with a wide range of gaits, from walking in rough terrains to fast running; 2) bipedal robots can naturally make use of environments designed for humans (e.g., stairs) without requiring architectonic modifications.

Key challenges in bipedal locomotion include balance control, foot placement, and gait optimization. In this paper, we focus on black-box gait optimization, i.e., finding good parameters for the gait of a biped, without the need of a dynamics model. Hence, we assume that a suitable controller to generate the desired gait has already been designed, but that appropriate gait parameters for the controller still need to be found. Due to the partially unpredictable effects and correlations among the gait parameters, gait optimization is often an empirical, time-consuming and strongly robot-specific process. In practice, gait optimization often translates into a trial-and-error process where choosing the parameters becomes either an educated guessing by a human expert or a systematic search, such as grid search. As a result, gait optimization may require considerable expert experience, engineering efforts and time-consuming experiments. Additionally, the effectiveness of the resulting gait parameters is limited by the circumstances and human insights speculated during the search process.

Therefore, a change in the environment (e.g., different floor surfaces), a variation in the hardware response (e.g., performance decline, substitution of a component or differences in the calibration) or the choice of a performance criterion (e.g., walking speed, energy efficiency, robustness), which differs from the one used during the controller design process, can require searching for new and more appropriate gait parameters.

By formulating the search for appropriate gait parameters as an optimization problem, it is possible to automate gait optimization. Automatic gait optimization is a valuable and principled approach to designing controllers and reduces the need for engineering expert knowledge. To date, various automatic gait optimization methods have been used in locomotion to design efficient gaits including: gradient descent methods [27], Bayesian optimization [18], [28], [5], genetic algorithms [6], particle swarm optimization [20] and others [12]. In the context of robotics, and specifically gait optimization, the number of experiments that can be performed on a real system is small. Each experiment can be costly, requires a long time, and inevitably leads to wear and tear of the robot’s hardware. Hence, it is crucial for the chosen optimization method to limit the number of experiments to perform while reliably finding near-optimal parameters. In the gait optimization literature, the results of the proposed methods are frequently presented without evaluating alternative methods. Additionally, each proposed method typically is evaluated on customized hardware. Therefore, any comparison inferred by a reader can give only a limited insight into the performance of different gait optimization methods. Furthermore, only rarely the characteristics, advantages and limitations of each method are clearly presented.

In this paper, we present an experimental evaluation and

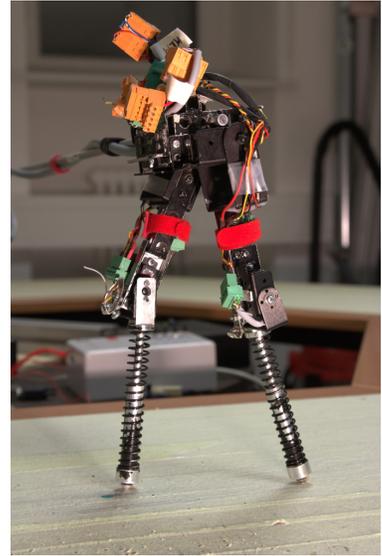


Fig. 1: The bio-inspired dynamical bipedal walker *Fox* used for the comparative evaluation of the gait optimization methods.

<sup>1</sup>Intelligent Autonomous Systems Lab, Department of Computer Science, Technische Universität Darmstadt, Germany

<sup>2</sup>Laufabor Locomotion Lab, Department of Sport Science, Technische Universität Darmstadt, Germany

<sup>3</sup>Max Planck Institute for Intelligent Systems, Tübingen, Germany

<sup>4</sup>Department of Computing, Imperial College London, UK

comparison of common automatic optimization methods in the context of gait optimization. In Section II, we formalize gait search as an optimization problem and introduce some popular optimization methods: grid search, pure random search, Bayesian optimization and gradient-descent-based methods. Furthermore, we discuss the properties and characteristics of the different methods. In Section III, we present the experimental set-up and the results obtained using the bio-inspired dynamical bipedal walker *Fox* shown in Figure 1. We conclude in Section IV with some final remarks and possible future directions for gait optimization. Clearly, the experimental results obtained with the optimization methods considered in this paper are more generally applicable in robotics, for example, in grasping, imitation learning and robot-human interaction.

## II. METHODS FOR GAIT OPTIMIZATION

The search for appropriate parameters of a controller can be formulated as an optimization problem, such as the minimization

$$\underset{\theta \in \mathbb{R}^D}{\text{minimize}} f(\theta) \quad (1)$$

of an objective function  $f(\cdot)$  with respect to the parameters  $\theta$ . In the case of gait optimization, the objective function  $f$  encodes a performance criterion, such as walking speed, energy efficiency or robustness, while  $\theta$  are the parameters of the gait controller. Hence, evaluating the objective function  $f$  for a given set of parameters requires a physical interaction with the robot.

The considered gait optimization problem has the following properties:

- 1) **Zero-order objective function.** Each evaluation of the objective function  $f$  returns only the value of the function  $f(\theta)$ , but no gradient information  $\nabla_{\theta} f = \text{d}f(\theta) / \text{d}\theta$  with respect to the parameters. The use of first-order optimization methods, which make use of gradient information, is generally desirable in optimization as it leads to faster convergence than zero-order methods. Thus, it is common for zero-order objective functions to approximate the gradient using finite differences. However, finite differences requires evaluating the objective function  $f$  multiple times. Since each evaluation requires interactions with the robot, the number of robot experiments quickly becomes excessive, rendering first-order methods (e.g., the whole family of gradient descent-based methods) unsuitable for our task.
- 2) **Stochastic objective function.** The evaluation of the objective function is inherently stochastic due to noisy measurements, variable initial conditions and system uncertainties (e.g., slack). Therefore, any suitable optimization method needs to take into consideration that two evaluations of the same parameters  $\theta$  can yield two different values  $f_1(\theta) \neq f_2(\theta)$ .
- 3) **Global optimization.** Ideally, we strive to find the global minimum of the objective function in (1). However, no assumption can be made about the presence of

Method	Order optimizer	Global optimizer	Re-usability evaluations	Stochasticity assumption
Grid Search	Zero-order	Global	Limited	No*
Pure Random Search	Zero-order	Global	Yes	No*
Gradient-descent Family	First-order	Local	No	No
Bayesian Optimization	Zero-order	Global	Yes	Yes

TABLE I: Comparison of various gait optimization methods. (\*) extensions exist for stochastic optimization, but they are impractical in terms of number of evaluations required.

multiple local minima or the convexity of the objective function.

These characteristics make this family of problems a challenging optimization task. Table I shows the methods commonly used to solve such a family of problems in the context of gait optimization. In the following, we introduce and analyze these optimization methods.

### A. Grid Search

Grid search is a widely adopted exhaustive optimization method, which evaluates the parameters along a (typically) regular grid. Each evaluation of the parameters is carried out independently of all the others evaluations. Therefore, grid search is a global zero-order optimization method as it can optimize non-convex objective functions and does not require gradients  $\nabla_{\theta} f$ . As a result, the underlying objective function  $f$  does not need to be differentiable or even continuous.

Due to the curse of dimensionality [9], the required number of evaluations grows exponentially with respect to the number of parameters. Given the parameters  $\theta \in \mathbb{R}^D$ , and assuming a grid with  $p$  evaluations along each parameter's covariate, the number of evaluations  $n$  required is

$$n = p^D, \quad (2)$$

which can rapidly become infeasible.

Since each evaluation of the objective function can be performed independently of the others, grid search can be easily parallelized. Hence, grid search is attractive in contexts where it is possible to parallelize the evaluations (e.g., using multiple robots on which to perform experiments). Classical grid search does not explicitly consider the case of stochastic objective functions, in which case it can fail to converge. Extensions exist that can deal with stochastic optimization [10], but they are impractical due to the need to evaluate each set of parameters multiple times, therefore, requiring an even higher numbers of evaluations.

Once the optimization is complete, grid search does not naturally allow for adding new evaluations. New evaluations would require to either redefine the grid and lose the past evaluations, or fill the gaps from the previous grid and typically lose the regular structure of the grid.

### B. Pure Random Search

Pure random search [3] selects the parameters  $\theta$  to evaluate by sampling from a bounded uniform distribution. Pure random search maintains most of the positive properties of grid search: zero-order global optimization, straightforward

parallelization. Despite its simplicity pure random search is an attractive approach due to the statistical guarantees of global convergence. Given  $n$  evaluations and an optimum having a relative space  $a$  (i.e., size of the space of the optimum with respect to the size of the overall parameters space), the probability  $S$  of finding the global optimum is

$$S = 1 - (1 - a)^n. \quad (3)$$

Typically, such probabilistic convergence guarantees require an impractical (i.e., tending to infinite) number of evaluations. However, pure random search often performs surprisingly well in finding good approximate solutions, even compared to more complicated and modern optimization approaches, e.g., in high-dimensional problems where many irrelevant dimensions exist [1].

Similarly to grid search, pure random search does not explicitly consider the case of stochastic objective functions, in which case it can fail to converge. Also for pure random search, there are extensions which can deal with stochastic optimization [7], but they are impractical due to the even higher numbers of evaluations required. Adding new evaluations, if an optimization proved unsatisfactory, can be easily done by sampling new parameters and evaluating them as well.

### C. Gradient-descent Family

Gradient-descent-based optimization is a wide family of optimization methods. They are based on gradient descent (GD), sometimes called “steepest descent”, which is a first-order iterative optimization method. At each iteration  $t$ , a set of parameters  $\theta$  is selected based on the update rule

$$\theta_{t+1} = \theta_t + \epsilon_t \nabla_{\theta_t} f, \quad (4)$$

where the learning rate  $\epsilon_t$  (also called step size) is a free parameter and  $\nabla_{\theta_t} f$  is the gradient of the objective function with respect to the parameters  $\theta_t$ . The selection of  $\epsilon_t$  is an important choice that can considerably affect the optimization process. While a constant  $\epsilon_t$  can be used, it would lead to an inefficient optimization, given that it would have to be tuned based on the particular specifics of the problem we want to optimize. A second possibility is the use of an adaptive step size (e.g., one that decreases through time). An appropriate adaptive learning rate toward the end of the optimization stabilizes the optimization process in a near-optimal minimum, rather than zig-zagging around. A common choice is to select an optimal  $\epsilon$  for each iteration using a linesearch. Other methods belonging to the gradient-descent family are conjugate gradients and L-BFGS [4].

There are important limitations of gradient-descent methods for the formulated optimization problem. 1) Computing the gradient  $\nabla_{\theta_t} f$  is undesirable in our case as it increases the number of evaluations required. 2) Gradient-descent methods are local: since they follow the gradient, they only find the minimum in whose basin of attraction the optimization started. Hence, the initialization of the parameters  $\theta_0$  becomes crucial and global optima are rarely found. Gradient-descent methods are occasionally used for global

---

### Algorithm 1: Bayesian optimization

---

```

1  $\mathbb{T} \leftarrow$  if available:  $\{\theta, f(\theta)\}$ 
2 Prior  $\leftarrow$  if available: Prior of the GP model
   while optimize do
3   Train GP model from  $\mathbb{T}$ 
4   Compute response surface  $\hat{f}(\cdot)$ 
5   Compute acquisition surface  $\alpha(\cdot)$ 
6   Find  $\theta^*$  that optimizes  $\alpha(\cdot)$ 
7   Evaluate  $f$  at  $\theta^*$ 
8   Add  $\{\theta^*, f(\theta^*)\}$  to  $\mathbb{T}$ 

```

---

optimization by running multiple optimizations initialized by different parameters  $\theta_0$ . However, this technique clearly requires many more evaluations, and therefore is impractical. Since each new set of parameters  $\theta_{t+1}$  depends only on the previous set of parameters  $\theta_t$ , gradient-descent methods do not allow for any re-use of past evaluations.

### D. Bayesian Optimization

Bayesian optimization is an iterative model-based global optimization method [16], [14], [21], [2]. In iterative model-based global optimization methods, after each evaluation of the objective function  $f$ , a model that maps parameters  $\theta$  to corresponding function evaluations  $f(\theta)$  is built. From the resulting model the response surface  $\hat{f}(\cdot)$  is predicted and used for a “virtual” optimization process

$$\underset{\theta \in \mathbb{R}^D}{\text{minimize}} \hat{f}(\theta). \quad (5)$$

In this context, “virtual” indicates that optimizing the response surface  $\hat{f}(\cdot)$  with respect to the parameters  $\theta$  does not need interactions with the real system, but only evaluations of the learned model. Only when a new set of parameters  $\theta^*$  has been selected from the virtual optimization process of the response surface  $\hat{f}$ , they are evaluated on the real objective function  $f$ . The resulting value of the objective function, together with the corresponding parameters  $\{\theta^*, f(\theta^*)\}$ , are used to update the model of the objective function.

A variety of different models, such as linear functions or splines [14], have been used in the past to model  $f$ . In Bayesian optimization, probabilistic models are used. The use of a probabilistic model allows to model noisy observations and to explicitly take the uncertainty about the model itself into account. Additionally, such a probabilistic framework allows to use priors that encode available expert knowledge or information from related systems, such as optimal parameter priors to changes in the system, e.g., after replacing a motor or changing the walking surface. In this paper, we use Gaussian processes (GPs) as probabilistic model for Bayesian optimization.

When using a probabilistic model, the response surface  $\hat{f}(\cdot)$  in Equation (5) is a probability distribution. Therefore, the response surface  $\hat{f}$  cannot be optimized directly. Instead, an **acquisition function**  $\alpha(\cdot)$  is necessary for the virtual optimization of the probabilistic GP. The purpose of

the acquisition function is two-fold: 1) It maps the GP onto a single surface called the acquisition surface  $\alpha(\boldsymbol{\theta})$ , which is subsequently optimized<sup>1</sup>. Thereby, the minimization of the objective function from Equation (1) can be rephrased as the minimization of the acquisition surface

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}} \alpha(\boldsymbol{\theta}). \quad (6)$$

2) The GP expresses model uncertainty, which is used to trade off exploration and exploitation in the optimization. In Bayesian optimization, as summarized in Algorithm 1, a GP model  $\boldsymbol{\theta} \mapsto f(\boldsymbol{\theta})$  is learned from the data set  $\mathbb{T} = \{\boldsymbol{\theta}, f(\boldsymbol{\theta})\}$  composed by the parameters  $\boldsymbol{\theta}$  and the corresponding measurements  $f(\boldsymbol{\theta})$  of the objective function (Line 3 of Algorithm 1). This model is used to predict the response surface  $\hat{f}$  (Line 4 of Algorithm 1) and the corresponding acquisition surface  $\alpha(\boldsymbol{\theta})$  (Line 5 of Algorithm 1), once the response surface  $\hat{f}(\cdot)$  is mapped through the acquisition function  $\alpha(\cdot)$ . Using a global optimizer, the minimum  $\boldsymbol{\theta}^*$  of the acquisition surface  $\alpha(\boldsymbol{\theta})$  is computed (Line 6 of Algorithm 1) without any evaluation of the objective function, e.g., no robot interaction is required, see Equation (6). This minimum  $\boldsymbol{\theta}^*$  is evaluated on the robot (Line 7 of Algorithm 1) and, together with the resulting measurement  $f(\boldsymbol{\theta}^*)$ , added to the dataset  $\mathbb{T}$  (Line 8 of Algorithm 1). Additionally, past evaluations can be used to initialize the dataset  $\mathbb{T}$  (Line 1 of Algorithm 1), as well as the prior of the GP model (Line 2 of Algorithm 1).

1) *Gaussian Process Model for Objective Function:* To create the model that maps  $\boldsymbol{\theta} \mapsto f(\boldsymbol{\theta})$ , we make use of Bayesian non-parametric Gaussian Process regression [22]. Such a GP is a distribution over functions

$$f \sim \text{GP}(m_f, k_f), \quad (7)$$

fully defined by a prior mean  $m_f$  and a covariance function  $k_f$ . As prior mean, we choose<sup>2</sup>  $m_f \equiv \mathbf{0}$ , while the chosen covariance function  $k_f$  is the squared exponential with automatic relevance determination and Gaussian noise

$$k_f(\boldsymbol{\theta}_p, \boldsymbol{\theta}_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)^T \boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)\right) + \sigma_w^2 \delta_{pq} \quad (8)$$

with  $\boldsymbol{\Lambda} = \text{diag}([l_1^2, \dots, l_D^2])$ . Here,  $l_i$  are the characteristic length-scales,  $\sigma_f^2$  is the variance of the latent function  $f(\cdot)$  and  $\sigma_w^2$  the noise variance. The GP predictive distribution is

$$p(f(\boldsymbol{\theta})|\mathbb{T}, \boldsymbol{\theta}) = \mathcal{N}(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta})), \quad (9)$$

where the mean  $\mu(\boldsymbol{\theta})$  and the variance  $\sigma(\boldsymbol{\theta})$  are respectively computed as

$$\mu(\boldsymbol{\theta}) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}, \quad \sigma^2(\boldsymbol{\theta}) = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*. \quad (10)$$

Given  $n$  training inputs  $\mathbf{X} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n]$  and corresponding training targets  $\mathbf{y} = [f(\boldsymbol{\theta}_1), \dots, f(\boldsymbol{\theta}_n)]$ , we define the training

<sup>1</sup>The correct notation would be  $\alpha(\hat{f}(\boldsymbol{\theta}))$ , but we use  $\alpha(\boldsymbol{\theta})$  for notational convenience.

<sup>2</sup>The use of a more informative prior allows to make use of expert knowledge.

data set  $\mathbb{T} = \{\mathbf{X}, \mathbf{y}\}$ . Moreover,  $\mathbf{K}$  is the matrix composed of  $K_{ij} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$ ,  $k_{**} = k(\boldsymbol{\theta}, \boldsymbol{\theta})$  and  $\mathbf{k}_* = k(\mathbf{X}, \boldsymbol{\theta})$ .

A practical issue, for both GP modeling and Bayesian optimization, is the choice of the hyperparameters of the GP model. The hyperparameters of a GP model are the parameters of the covariance function, in our case, the characteristic length-scales  $l_i$ , the variance of the latent function  $\sigma_f^2$  and the noise variance  $\sigma_w^2$ . In gait optimization, these hyperparameters are often fixed a priori [18]. There are suggestions [17] that fixing the hyperparameters can considerably speed up the convergence of Bayesian optimization. However, manually choosing the value of the hyperparameters requires extensive expert knowledge about the system that we want to optimize, which is often an unrealistic assumption. Additionally, it is unclear if fixing the hyperparameters provide an advantage that balances the risk of incorrectly approximate the real hyperparameters. An alternative common approach, which we employ in our experiments, is to automatically select the hyper-parameters by optimizing with respect to the marginal likelihood [22].

2) *Acquisition Function:* A number of acquisition functions  $\alpha(\cdot)$  exists, such as probability of improvement [16], expected improvement [19], upper confidence bound [8] and entropy-based improvements [13]. Experimental results [13] suggest that expected improvement on specific families of artificial functions performs better on average than probability of improvement and upper confidence bound. However, these results do not necessarily hold true for real-world problems such as gait optimization, where the objective functions are more complex to model. Probability of improvement [18], expected improvement [28] and upper confidence bound [5] have all been previously employed in gait optimization. However, no experimental comparison has been carried out in gait optimization yet, and it is still unclear why one of them should be chosen over the others.

a) *Probability of Improvement (PI):* Introduced by Kushner [16], the acquisition function PI is defined as

$$\alpha(\boldsymbol{\theta}) = \Phi\left(\frac{T - \mu(\boldsymbol{\theta})}{\sigma(\boldsymbol{\theta})}\right), \quad (11)$$

where  $\Phi(\cdot)$  is the normal cumulative distribution function and  $T$  the target value. The target value  $T$  is often the minimum of all explored data plus, optionally, a positive constant (for a study of its effects, see [17]). PI is a function bounded by the interval  $[0, 1]$ . Hence, since the normal cumulative distribution function is monotonically increasing, to minimize PI is sufficient to minimize

$$\alpha(\boldsymbol{\theta}) = \frac{T - \mu(\boldsymbol{\theta})}{\sigma(\boldsymbol{\theta})}. \quad (12)$$

Intuitively, PI computes the probability (cumulative distribution) of the response surface in  $\boldsymbol{\theta}$  to be better than the target value  $T$ .

b) *Expected Improvement (EI):* Mockus [19] introduced EI, which can be seen as an extension of probability

of improvement. EI is formulated as

$$\alpha(\boldsymbol{\theta}) = \sigma[u\Phi(u) + \phi(u)]; \quad u = \frac{T - \mu(\boldsymbol{\theta})}{\sigma(\boldsymbol{\theta})}, \quad (13)$$

where  $\phi(\cdot)$  is the normal probability density function.

c) *Upper Confidence Bound (UCB)*: UCB [8] is defined as

$$\alpha(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta}) - \kappa\sigma(\boldsymbol{\theta}). \quad (14)$$

The choice of the free parameter  $\kappa$  is crucial as it determines the trade-off rate between exploration and exploitation. GP-UCB [26] suggests to automatically select  $\kappa$  according to

$$\kappa = \sqrt{2 \log \left( \frac{n^{D/2+2\pi^2}}{3\delta} \right)}, \quad (15)$$

where  $\delta$  is the number of past evaluations of the objective function  $f$  and  $D$  the dimensionality of the parameters  $\boldsymbol{\theta}$ . Additionally, automatically selecting  $\kappa$  allows to estimate regret bounds [26].

3) *Optimizing the Acquisition Surface*: Once the acquisition surface  $\alpha(\boldsymbol{\theta})$  is computed (Line 5 of Algorithm 1), it is still necessary to find the parameters  $\boldsymbol{\theta}^*$  of its minimum (Line 6 of Algorithm 1). To find this minimum, we use a standard global optimizer. Note that the global optimization problem in Equation (6) is different from the original global optimization problem defined in Equation (1). First, the measurements in Equation (6) are noise free because the objective function in Equation (14) is an analytical model. Second, there is no restriction in terms of how many evaluations we can perform. Evaluating the acquisition surface only requires interactions with the model, but not with a physical system, such as a robot. Third, we can compute the derivatives of any order, either with finite differences or analytically. Therefore, we are no longer restricted to zero-order optimization methods, and any global optimizer can be used. In our experiments we used DIRECT [15] to find the approximate global minimum, followed by L-BFGS [4] to refine it.

### III. COMPARATIVE EVALUATION ON THE *Fox* ROBOT

In this section, we first introduce the bio-inspired dynamical bipedal walker *Fox* used as our robotic evaluation platform. Afterward, we describe the experimental set-up and give details for each of the optimization methods employed. Finally, we present the comparison of the experimental evaluations and discuss it.

#### A. Experimental Set-up

To validate our Bayesian gait optimization approach we used the 2-D dynamic walker *Fox*, shown in Figure 1. This robot consists of a rudimentary trunk, two legs, made of a rigid segment connected by a knee joint to a telescopic leg spring, and two spherical feet with touch sensors [23].

*Fox* is equipped with low-cost metal-gear DC motors at both hip and knee joints. Together they drive four actuated degrees of freedom. Moreover, there are six sensors on the robot: two on the hip joints, two on the knee joints, and one under each foot. The sensors on the hip and knee joints return voltage measurements

corresponding to angular positions of the leg segments. The touch sensor under each foot returns binary ground contact signals. The walker is mounted on a boom that enforces planar, circular motion. An additional sensor in the boom measures the angular position of the walker, i.e., the position of the walker on the circle.

The controller of the walker is a **finite state machine (FSM)** with four states (see Figure 3): two for the swing phases of each leg [24]. These states control the actions performed by each of the four actuators, which were extension, flexion or holding of the joint. The transition between states is regulated by the contact sensors and thresholds based on the angles of the joints. For the optimization process, we identified four parameters of the controller crucial for the resulting gait; More parameters would make a comparison with grid search unfeasible, due to the excessive number of evaluations required. These four gait parameters consist of thresholds values of the FSM (two for each leg), and directly influence the position of the hips. The remaining parameters were chosen such that only a small set of parameters space would lead to a properly walking gait. To achieve a stable gait in this parameters space it was necessary to make use of the inertia of the legs to compensate the low torques applied.

#### B. Performance Metric

A common metric used in gait optimization is the average walking velocity  $\bar{v} = \Delta x / \Delta t$ , with  $\Delta x$  being the walked distance during the time  $\Delta t$ . However, this metric can be misleading since fast gaits can also be sensitive to intrinsic perturbations that lead to falls. Therefore, the objective function  $f$  to be minimized was defined as

$$f(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^N \bar{v}_i(\boldsymbol{\theta}), \quad (16)$$

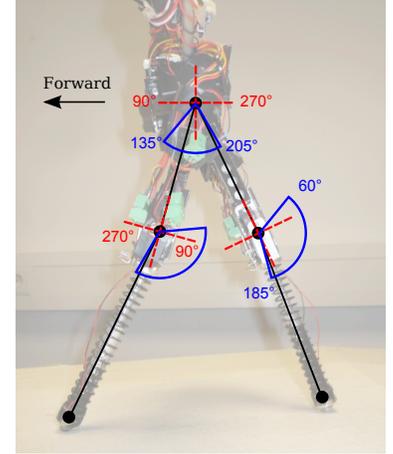


Fig. 2: Hip and knee angle reference frames (red dashed) and rotation bounds (blue solid). The hip joint angles' range lies between  $135^\circ$  forward and  $205^\circ$  backward. The knee angles range from  $185^\circ$  when fully extended to  $60^\circ$  when flexed backward.

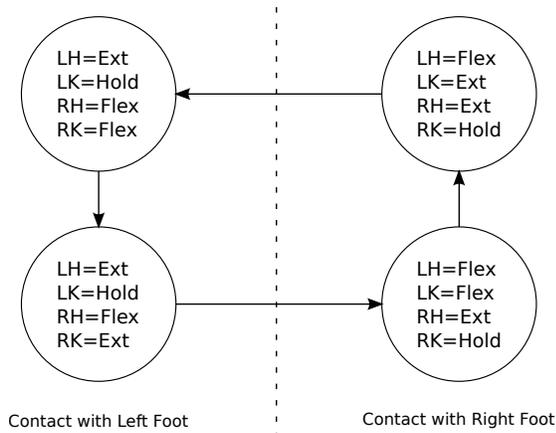


Fig. 3: *Fox'* controller is a finite state machine with four states. Each of the four joints, left hip (LH), left knee (LK), right hip (RH) and right knee (RK), can perform one of three actions: flexion (Flex), extension (Ext) or holding (Hold). When a joint reaches the maximum extension or flexion, its state is transparently changed to holding. The transition between the states and the torque applied during flexion and extension are determined by the controller parameters  $\theta$ .

i.e., the negative mean of the average walking velocity  $\bar{v}$  over  $N = 3$  experiments on the robot for a given set of gait parameters  $\theta$ . Minimizing the performance criterion in Equation (16) maximizes the walking distance. This criterion does not only guarantee a fast walking gait, but also reliability since the gait must be robust to noise and initial configurations across multiple experiments. The chosen parameters space is sufficiently large that only a small percentage of the possible parameters values can achieve stable walking, while for most of the configurations the robot falls down after one or two steps. Each experiment was initialized from similar initial configurations, and each experiment consisted of 12 seconds starting from the moment when the foot of the robot first touched the ground. For grid search optimization, we used 3 evaluations along each of the four dimensions for a total of 81 evaluations (from Equation (2)). For comparability, we performed for all other methods the same number of evaluations. To initialize Bayesian optimization, we used the first three evaluations from pure random search (i.e., uniformly randomly sampled sets of parameters), thus, leaving 78 evaluations to be selected. Performing a whole cycle of optimization (i.e.,  $78 \times 3 = 234$  evaluations) consists of 46 minutes of robot walking time and typically required between 6 and 9 hours of hands-on experimental time.

### C. Experimental Results

The maximum walking speed of *Fox* evaluated during the gait optimization process for the different methods is shown in Figure 4a. The optimization process of GP-UCB is limited to 57 evaluations due to a mechanical failure that forcefully interrupted the experiment. Values of the objective function below 0.1 m/s indicate that the robot fell down after a single step. Values between 0.1 and 0.15 m/s indicate that the

robot could walk for multiple steps but showed systematic falls. Between 0.15 m/s and 0.25 m/s only occasional falls occurred. Above 0.25 m/s the achieved gait was robust and did not present any fall. From the results, we see that both grid search and random search performed poorly, finding a maximum that can only barely walk. We notice that Bayesian optimization using any of the different acquisition functions performed considerably better. Bayesian optimization using PI and GP-UCB achieved robust gaits with similar walking speed, while GP-UCB being slightly faster in finding the maximum. On the other hand, Bayesian optimization using EI did not lead to robust gaits. The reason of this result were the inaccuracies of the model of the underlying objective function. In fact, the automatically selected hyperparameters had overly long length-scales (see Equation (8)), which resulted in an inappropriate model and, thus, evaluating parameters of little interest.

This result is unexpected as EI is considered a versatile acquisition function, and there are experimental results [13], which suggest that EI on specific families of artificial functions performs better than GP-UCB and PI. We speculate that these results do not necessarily apply to complex real-world objective functions, such as the one we optimized. For example, if the objective function does not possess a single natural length-scale, but multiple length-scales. In particular, during the empirical evaluation, we observed that EI behaved excessively greedily, exploring the parameters space insufficiently. In turn, this insufficient exploration resulted in overly long length-scales and an inappropriate GP model. Therefore, we hypothesize that in case of an objective function which is complicated to model, the EI acquisition function might not perform as well as for easier artificial functions.

As a second comparison, we studied the effects of manually fixing hyperparameters. Thereby, we manually fixed the hyperparameters of the GP models to reasonable values, based on our expert knowledge. Figure 4b shows the performance of the different acquisition functions when using the manually fixed hyperparameters. From these results, it can be observed that all the different acquisition functions, when using the fixed hyperparameters, found similar sub-optimal solutions. The reason is that for all three acquisition functions with fixed hyperparameters, one parameter reached only a sub-optimal value. This observation suggests that, at least for that one parameter, the wrong length-scales prevented the creation of an accurate model and, therefore, the optimization process was hindered. As a confirmation of this hypothesis, using all the evaluations performed, we trained a GP model and automatically selected the hyperparameters using the marginal likelihood. The resulting values of the hyperparameters were, in some cases, half of the manually selected values, therefore, suggesting that the chosen length-scales were a rough approximation of the real ones. Both GP-UCB and PI using fixed hyperparameters performed worse than the respective cases with automatic hyperparameters selection because the longer length-scales limited the exploration of these acquisition functions. In contrast, for EI

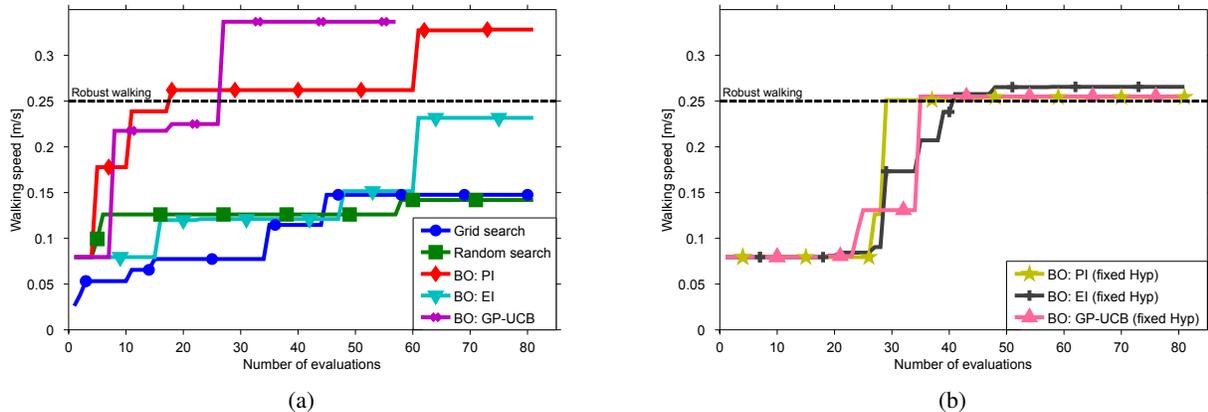


Fig. 4: The maximum walking speed of *Fox* evaluated during the gait optimization process. (a) Bayesian optimization performed better than both grid and random search. BO using the GP-UCB acquisition function performed best, achieving a fast and robust gait in less than 30 evaluations. (b) Bayesian optimization of various acquisition functions are shown for fixed hyperparameters. The use of imprecise fixed hyperparameters led to sub-optimal solutions for all the acquisition functions.

Method	Maximum during optimization
Grid search	0.148
Pure random search	0.142
BO: PI	0.328
BO: EI	0.232
BO: GP-UCB	<b>0.337</b>
BO: PI (fixed Hyp)	0.254
BO: EI (fixed Hyp)	0.266
BO: GP-UCB (fixed Hyp)	0.255

TABLE II: Maximum average walking speeds [m/s] found by the different optimization methods. Bayesian optimization using GP-UCB with automatic hyperparameters selection found the best maximum of all methods. Nevertheless, the maximum obtained using PI is qualitatively similar.

the use of fixed hyperparameters was beneficial. In fact, the fixed length-scales were smaller than the automatically selected ones and, therefore, more exploration was performed. The hyperparameters of the GP model directly influence the amount of exploration performed by the acquisition functions. Hence, fixing the hyperparameters using expert knowledge can be an attractive choice, since forcing the right amount of exploration can speed up the optimization process. However, the presented experimental results also show that a poor choice of hyperparameters can potentially harm the optimization process by limiting the exploration and leading to sub-optimal solution.

In Table II, we present the comparison of the optimum found by all the evaluated methods. Variations of up to 0.04 m/s can depend on the presence of noise in the experiments. Additionally, it should be noticed that the real noise of the objective function is not Gaussian. In fact, configurations of the parameters that produce periodic falls after a single step or stable gaits, typically behave consistently across various experiments and, therefore, present smaller noise (0.01 m/s). For parameters that produce unstable gaits with occasional falls the noise can be larger, typically up

to 0.04 m/s. A visual representation between two of the parameters of the parameter space as predicted using the data collected from all the over 1800 evaluations is shown in Figure 5. It can be noticed that this space is complex and non-convex, and, therefore, unsuitable for local optimization methods (e.g., gradient-based methods).

The GP modeling capabilities are often overlooked when evaluating Bayesian optimization’s performances, with all the emphasis on the use of different acquisition functions. Following the results of our experimental evaluation, we speculate that for complex objective functions there exists a strict and yet unexplored connection between the exploration properties of the acquisition function and the capabilities of GP modeling. The performance of an acquisition function depends on the capabilities of properly modeling the function, and vice-versa a proper modeling take place only when the acquisition function evaluate relevant parameters.

#### IV. DISCUSSION & CONCLUSION

Gait optimization is a key research topic relevant for efficient bipedal locomotion. However, due to the presence of ad-hoc solutions and different hardware, the proposed optimization methods had not yet been thoroughly compared to other methods. In this paper, we presented a survey of some gait optimization methods and discussed their strengths and limitations. To compare the different gait optimization methods, we performed over 1800 evaluations on a real bipedal walker. Experimental results show that Bayesian optimization performed considerably better than grid or random search. In the context of Bayesian optimization, we firstly compared different acquisition functions. While GP-UCB had the best performances, EI greedily chose the parameters to evaluate and therefore performed poorly. Secondly, we compared the manually fixing hyperparameters against automatically selecting them. The results showed that manually fixing the hyperparameters can strongly influence the outcome of the optimization process. These experimental

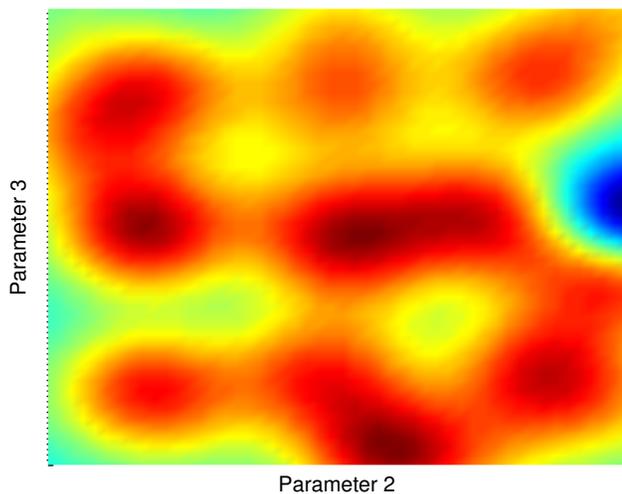


Fig. 5: Intensity map of the model of the parameters space, computed using all the evaluations performed from the different optimization methods, along two of the parameters. The presence of multiple local minima motivate the needs of using global optimization methods.

results show interesting insights of the different acquisition functions and into the difficulties of modeling complex objective functions.

Bayesian optimization is a promising method for efficient optimization, especially in fields like locomotion where only few evaluations can be performed before wearing out the hardware. Future research in Bayesian optimization should focus on obtaining more accurate and appropriate response surface models. Since the experimental noise is often not constant, a logical extension would be the use of heteroscedastic GP models. Additionally, further studies are required to fully understand the joint influence of different acquisition functions and models, in the case of hard to model complex objective functions.

#### ACKNOWLEDGMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreements #270327 (CompLACS) and #600716 (CoDyCo) and from the Department of Computing, Imperial College London.

#### REFERENCES

- [1] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)*, 13:281–305, 2012.
- [2] E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.
- [3] S. H. Brooks. A discussion of random methods for seeking maxima. *Operations Research*, 6(2):244–251, 1958.
- [4] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.
- [5] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth. Bayesian gait optimization for bipedal locomotion. In *Proceedings of Learning and Intelligent Optimization Conference (LION8)*, 2014.

- [6] S. Chernova and M. Veloso. An evolutionary approach to gait learning for four-legged robots. In *Intelligent Robots and Systems (IROS)*, volume 3, pages 2562–2567. IEEE, 2004.
- [7] Y. L. Chia and P. W. Glynn. Limit theorems for simulation-based optimization via random search. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 23(3):16, 2013.
- [8] D. D. Cox and S. John. SDO: A statistical method for global optimization. *Multidisciplinary design optimization: state of the art*, pages 315–329, 1997.
- [9] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality, 2000.
- [10] K. B. Ensor and P. W. Glynn. Stochastic optimization via grid search. *Lectures in Applied Mathematics - American Mathematical Society*, 33:89–100, 1997.
- [11] S. M. Gatesy and A. A. Biewener. Bipedal locomotion: effects of speed, size and limb posture in birds and humans. *Journal of Zoology*, 224(1):127–147, 1991.
- [12] T. Geng, B. Porr, and F. Wörgötter. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *The International Journal of Robotics Research*, 25(3):243–259, 2006.
- [13] P. Hennig and C. J. Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research (JMLR)*, 13:1809–1837, 2012.
- [14] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, 2001.
- [15] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.
- [16] 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.
- [17] D. J. Lizotte, R. Greiner, and D. Schuurmans. An experimental methodology for response surface optimization methods. *Journal of Global Optimization*, 53(4):699–736, 2012.
- [18] 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.
- [19] J. Mockus, V. Tiesis, and A. Zilinskas. The application of Bayesian methods for seeking the extremum. *Towards Global Optimization*, 2:117–129, 1978.
- [20] 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.
- [21] M. A. Osborne, R. Garnett, and S. J. Roberts. Gaussian processes for global optimization. In *3rd International Conference on Learning and Intelligent Optimization (LION3)*, pages 1–15, 2009.
- [22] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [23] D. Renjewski. *An engineering contribution to human gait biomechanics*. PhD thesis, TU Ilmenau, 2012.
- [24] D. Renjewski and A. Seyfarth. Robots in human biomechanics - a study on ankle push-off in walking. *Bioinspiration & Biomimetics*, 7(3):036005, 2012.
- [25] S. Seok, A. Wang, M. Y. Chuah, D. Otten, J. Lang, and S. Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3307–3312. IEEE, 2013.
- [26] N. Srinivas, A. Krause, S. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In J. Fürnkranz and T. Joachims, editors, *Proceedings of International Conference on Machine Learning (ICML)*, pages 1015–1022, Haifa, Israel, June 2010. Omnipress.
- [27] R. Tedrake, T. W. Zhang, and H. S. Seung. Learning to walk in 20 minutes. In *Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems*, 2005.
- [28] M. Tesch, J. Schneider, and H. Choset. Using response surfaces and expected improvement to optimize snake robot gait parameters. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1069–1074. IEEE, 2011.