

Bayesian Optimization for Learning Gaits under Uncertainty

An experimental comparison on a dynamic bipedal walker

Roberto Calandra · André Seyfarth ·
Jan Peters · Marc Peter Deisenroth

Received: date / Accepted: date

Abstract Designing gaits and corresponding control policies is a key challenge in robot locomotion. Even with a viable controller parameterization, finding near-optimal parameters can be daunting. Typically, this kind of parameter optimization requires specific expert knowledge and extensive robot experiments. Automatic black-box gait optimization methods greatly reduce the need for human expertise and time-consuming design processes. Many different approaches for automatic gait optimization have been suggested to date, such as grid search and evolutionary algorithms. In this article, we thoroughly discuss multiple of these optimization methods in the context of automatic gait optimization. Moreover, we extensively evaluate Bayesian optimization, a model-based approach to black-box optimization under uncertainty, on both simulated problems and real robots. This evaluation demonstrates that Bayesian optimization is particularly suited for robotic applications, where it is crucial to find a good set of gait parameters in a small number of experiments.

Keywords Gait optimization · Bayesian optimization · Robotics · Locomotion

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). MPD has been supported by an Imperial College Junior Research Fellowship.

Roberto Calandra
Intelligent Autonomous Systems, TU Darmstadt, Germany
E-mail: calandra@ias.tu-darmstadt.de

André Seyfarth
Laufflabor Locomotion Laboratory, TU Darmstadt, Germany
E-mail: seyfarth@sport.tu-darmstadt.de

Jan Peters
Intelligent Autonomous Systems, TU Darmstadt, Germany
and Max Planck Institute for Intelligent Systems, Tübingen, Germany
E-mail: mail@jan-peters.net

Marc Peter Deisenroth
Department of Computing, Imperial College London, United Kingdom
E-mail: m.deisenroth@imperial.ac.uk

1 Introduction

Bipedal walking and running are versatile and fast locomotion gaits. However, despite its high mobility, bipedal locomotion is rarely used in real-world robotic applications. Key challenges in bipedal locomotion include balance control, foot placement, and gait optimization. In this article, we focus on **gait optimization**, i.e., finding good parameters for the gait controller of a robotic biped.

Due to the partially unpredictable effects of and interactions among the gait parameters, gait optimization is often an empirical, time-consuming and strongly robot-specific process. In practice, gait parameter optimization often translates into a trial-and-error process, which requires an educated guess by a human expert or a systematic but time-consuming parameter search. As a result, gait optimization may require considerable expert knowledge, engineering effort and time-consuming experiments. Additionally, the effectiveness of the resulting gait strongly depends on the conditions assumed during the controller design process: A change in these conditions, often requires searching for new, more appropriate, gait parameters. Such changes include changes in the environment (e.g., different surfaces), a variation in the hardware response (e.g., hardware wear and tear, replacement of a motor or differences in the calibration) or a different performance criterion (e.g., walking speed, energy efficiency, robustness). Hence, to deploy walking robots in the real world, it is essential to reduce the dependence on expert knowledge and automate the gait optimization process.

The search for gait parameters can be formulated as an optimization problem. Such a problem formulation, in conjunction with an appropriate optimization method, allows to automate the search for optimal gait parameters and reduces the need for engineering expert knowledge. To date, automatic gait optimization methods have been used for designing efficient gaits in locomotion [1, 2, 3]. Commonly, these methods only find locally optimal solutions and do not take sources of uncertainty (e.g., measurement noise) into account. Moreover, many optimization methods require a high number of function evaluations to find a good solution. Since each function evaluation requires an experiment with the robot, standard optimization methods are time-consuming and will eventually cause severe wear and tear on the robot, rendering these methods economically impractical. In practice, it is often essential to keep the number of robot experiments small.

To overcome this practical constraint on the number of possible interactions with the robot, we propose to use **Bayesian optimization** for efficient bipedal gait optimization. Bayesian optimization is a state-of-the-art global optimization

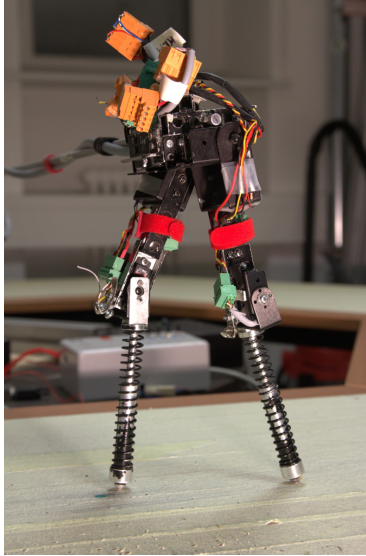


Fig. 1: The bio-inspired dynamical bipedal walker *Fox*. Using Bayesian optimization, we found reliable and fast walking gaits with a velocity of up to 0.45 m/s.

method [4,5,6,7] that can be applied to problems where it is vital to optimize a performance criterion while keeping the number of evaluations of the system small, e.g., when an evaluation requires an expensive interaction with a robot. Bayesian optimization makes efficient use of past interactions (experiments) by learning a probabilistic surrogate model of the function to optimize. Subsequently, the learned surrogate model is used for finding optimal parameters *without* the need to evaluate the expensive (true) function. By exploiting the learned model, Bayesian optimization, therefore, often requires fewer interactions (i.e., evaluations of the true objective function) than other optimization methods [5]. Bayesian optimization can also make good use of prior knowledge, such as expert knowledge or data from related environments or hardware, by directly integrating it into the prior of the learned surrogate model. Moreover, unlike most optimization methods, it can re-use any collected interaction data set, e.g., whenever we want to change the performance criterion. Bayesian optimization has been successfully applied to sensor-set selection [8], gait optimization for quadrupeds [9] and snake robots [10] and automatic algorithm configuration [11,12].

This paper builds upon and extends our previous work on Bayesian optimization for robotics [13,14]. In [13], Bayesian optimization was applied to gait optimization of a bipedal robot. Three acquisition functions and the effect of fixed versus automatic hyperparameter selection were analyzed. Our extensive evaluation with more than 1,800 experiments with the robot shown in Fig. 1 highlights the practicality and exposes strengths and weaknesses of commonly used acquisition functions. In [14], we considered a more challenging set-up with a higher number of parameters. We successfully applied Bayesian optimization for automatic gait optimization of up to eight parameters on a bipedal robotic walker.

In this article, we additionally formalize the problem of automatic gait optimization and discuss the practicality of commonly used optimization methods. Furthermore, we analyze *a posteriori* the quality of the models learned. The results of this analysis motivate the need for an efficient global optimization algorithm and give insights into the effects and interaction between the parameters.

2 Parameter Optimization under Uncertainty in Robotics

In this section, we formalize automatic parameter optimization under uncertainty in the context of robotics. Moreover, we discuss classical optimization methods and related work in the context of gait optimization.

2.1 Parameter Optimization in Robotics

The search for good controller parameters θ^* can be formulated as an optimization problem, such as the maximization

$$\theta^* \in \arg \max_{\theta \in \mathbb{R}^d} f(\theta), \quad (1)$$

of an objective function $f(\cdot)$ with respect to controller parameters $\theta \in \mathbb{R}^d$.

In robotics, the objective function f often encodes a single performance criterion, such as precision, speed, energy efficiency, robustness or a mixture of them.

| Method | Order optimizer | Stochasticity assumption | Global optimizer | Re-usability evaluations |
|-------------------------|-----------------|--------------------------|------------------|--------------------------|
| Grid Search | Zero-order | No* | Global | Limited |
| Pure Random Search | Zero-order | No* | Global | Yes |
| Gradient-descent family | First-order | No* | Local | No |
| Bayesian Optimization | Zero-order | Yes | Global | Yes |
| Evolutionary Algorithms | Zero-order | No* | Global | No |
| Particle Swarm | Zero-order | No* | Global | No |

Table 1: Optimization methods in robotics: Properties of various optimization methods commonly used for optimization in robotics. As discussed in Section 2.1, the ideal optimizer for robotic applications should be global, zero-order, and assuming stochasticity.

(*) Extensions exist for the stochastic case, but they increase the number of experiments required.

In gait optimization, the relevant criteria is typically walking speed or possibly a mixture together with energy efficiency and robustness, while θ are the parameters of an existing gait. Optimizing analytically the objective function f is typically unfeasible since the relation between the controller parameters and the objective function is unknown. Hence, we need to use numerical black-box optimization where evaluating the objective function f for a given set of parameters requires a physical interaction with the robot.

The general parameter optimization problem in robotics (as well as our considered gait optimization task) possesses the following properties:

- **Zero-order objective function.** Each evaluation of the objective function f returns the value of the function $f(\theta)$, but no information about the gradient $\nabla_{\theta} f = df(\theta)/d\theta$ with respect to the parameters θ .
- **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 account for the fact that two evaluations of the same parameters can yield two different values.
- **Global optimization.** No assumption can be made about the number of local maxima or the convexity of the objective function f . However, ideally we seek the global maximum of the objective function.

These characteristics render this family of problems a challenging optimization task. Additionally, in the context of robotics the number of experiments that can be performed on a real system is small. Each experiment can be costly, require a long time, and it inevitably contributes to the wear and tear of the robot’s hardware. Therefore, the optimizer must be as experimentally-efficient as possible. As result, the capability of re-using past experiments (e.g., experiments with random parameters) is a desirable property to keep the number of experiments small.

2.2 Optimization Methods in Robotics

Commonly used algorithms in robotics include grid search, gradient descent, evolutionary algorithms and others. In the following, we present some of the most

common optimization methods and we discuss the main limitations that make these algorithms unsuitable for robotic applications. Table 1 shows a summary of the methods discussed.

Due to its ease of use, the most common optimization method in robotics is grid search. Its main limitation is that it is an exhaustive search method and, thus, requires many experiments. In fact, the number n of experiments grows exponentially with the number of parameters d as $n = p^d$ where p is the number of experiments along each parameter dimension. An alternative to grid search is pure random search [15], which possesses statistical guarantees of convergence and often outperforms grid search, e.g., in high-dimensional problems where many irrelevant dimensions exist [16]. Nonetheless, even pure random search requires a number of experiments that is impractical in many robotic applications.

Another family of optimization methods commonly used in robotics are first-order methods, such as gradient descent. The use of first-order optimization methods, which make use of gradient information, is generally desirable in optimization as they lead to faster convergence than zero-order methods. Thus, it is common in the case of 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 also first-order methods (e.g., the whole family of gradient descent) unsuitable for our task.

Particle swarm and evolutionary algorithms are two common global optimization methods, which make use of populations of particles (or individuals), which explore the parameter space. However, both methods typically need thousands or tens of thousands of experiments to find good solutions. Hence, they are not easily applicable to real robots.

2.3 Related Work in Robot Locomotion

To date, various automatic gait optimization methods have been used in locomotion to design gaits, including gradient descent methods [18, 3], evolutionary algorithms [1, 19], particle swarm optimization [20] and many others [21, 22, 2, 17].

We now discuss the approaches that use surrogate models to optimize robot locomotion. In [21], a surrogate model optimization is performed on a bipedal robot using a non probabilistic model. Their approach is in the spirit of Bayesian optimization. However, since the acquisition function can only use a deterministic prediction from the model there is no exploration-exploitation trade-off and the next point to evaluate is selected to greedily exploit the model. Performing pure exploitation typically leads to find only local and suboptimal solutions. In robot locomotion, Bayesian optimization has been applied to quadrupedal robot [9], snake robots [10] and bipedal walkers [13]. In [9], two gait optimization criteria are considered for a Sony AIBO

| Method | Locomotion |
|-------------------------|-------------|
| Grid Search | [2, 17] |
| Pure Random Search | - |
| Gradient-descent family | [18, 3] |
| Bayesian Optimization | [9, 10, 13] |
| Evolutionary Algorithms | [1, 19] |
| Particle Swarm | [20] |

Fig. 2: Related work in robot locomotion: Various optimization methods and the corresponding work in robot locomotion where they are applied.

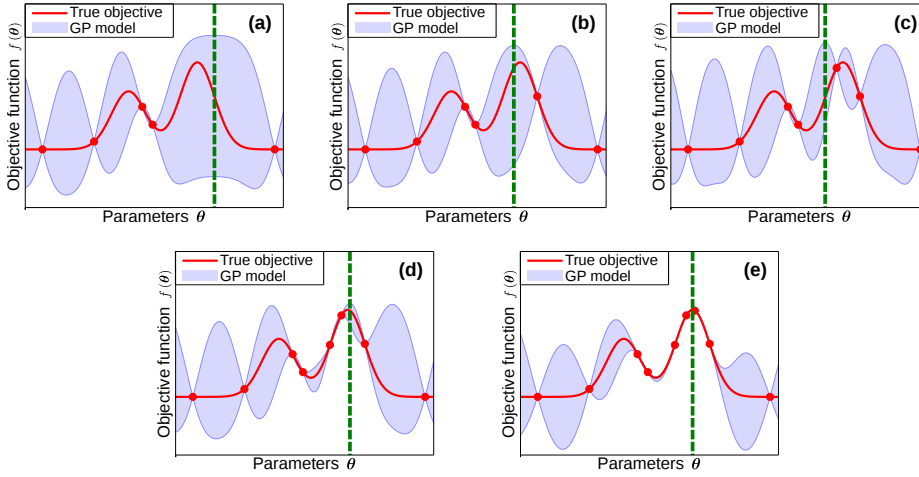


Fig. 3: Example of the Bayesian optimization process during the maximization of an unknown 1-D objective function f (red curve). The 95% confidence of the model prediction is represented by the blue area. The model is initialized with 5 previously evaluated parameters θ and the corresponding function values $f(\theta)$. The location of the next parameter to be evaluated is represented by the vertical green dashed line. At each iteration, the model is updated using all the previously evaluated parameters (red dots). Bayesian optimization quickly found the global maximum of the unknown objective function, after a few iterations.

ERS-7 quadrupedal robot: once with respect to the maximum walking speed and once for the maximum gait smoothness. As acquisition function the authors used Probability of improvement (which we discuss in Section 3.2), and as a model a standard Gaussian process. The hyperparameters of the GP were manually selected by a human expert at the beginning of the optimization. Correctly fixing the hyperparameters generally simplifies the optimization process and, therefore, speeds up the optimization [23]. Nonetheless, it requires a deep knowledge of the optimization task, which is typically an unrealistic assumption. In [10], Bayesian optimization is used to optimize the gait of a snake robot. The authors used expected improvement as acquisition function, which we discuss in Section 3.2.

3 Introduction to Bayesian Optimization

Bayesian optimization (BO) is a global optimization method [4, 6, 7] based on response surface (i.e., surrogate model). Bayesian optimization has been re-discovered multiple times by different communities and is also referred to as efficient global optimization (EGO) [24] and sequential kriging optimization (SKO) [25].

Response surface-based optimization methods iteratively create a data set $\mathbb{D} = \{\theta, f(\theta)\}$ of parameters θ and the corresponding function evaluations $f(\theta)$ [5]. This data set is used to build a model $\hat{f}(\cdot) : \theta \mapsto f(\theta)$, the **response surface**, that maps parameters θ to corresponding function evaluations $f(\theta)$. The response surface is subsequently used to replace the optimization of Eq. (1) with a “virtual”

Algorithm 1: Bayesian optimization

```

1 D ← if available:  $\{\boldsymbol{\theta}, f(\boldsymbol{\theta})\}$ 
2 Prior ← if available: Prior of the response surface
3 while optimize do
4   Train a response surface from D
5   Find  $\boldsymbol{\theta}^*$  that maximizes the acquisition surface  $\alpha(\boldsymbol{\theta})$ 
6   Evaluate  $f(\boldsymbol{\theta}^*)$  on the real system
7   Add  $\{\boldsymbol{\theta}^*, f(\boldsymbol{\theta}^*)\}$  to D

```

optimization process

$$\boldsymbol{\theta}^* \in \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^d} \hat{f}(\boldsymbol{\theta}) . \quad (2)$$

In this context, “virtual” indicates that optimizing the response surface $\hat{f}(\cdot)$ with respect to the parameters $\boldsymbol{\theta}$ only requires to evaluate the learned model, but not the true objective function f . Only when a new set of parameters $\boldsymbol{\theta}^*$ is determined by means of the virtual optimization process of the response surface \hat{f} , it is eventually evaluated on the real objective function f .

A variety of different models, such as linear functions or splines [5], have been used in the past to create the response surface that models f . In Bayesian optimization, probabilistic models are used. The use of a probabilistic model allows us to model noisy observations and to explicitly take the uncertainty about the model itself into account, which makes the probabilistic model more robust to the effect of model errors. A probabilistic framework also allows to use priors that encode available expert knowledge or information from related systems in a principled way. In the context of a walking robot, this knowledge could be a prior on the optimal parameters after a motor was replaced or the walking surface changed. The most common probabilistic model used in Bayesian optimization, and the one that we consider in this article, is a Gaussian process (GP) [26]. Nonetheless, other probabilistic models are possible, such as random forests [11].

When using a probabilistic model, the response surface $\hat{f}(\cdot)$ in Eq. (2) is a probability distribution. Therefore, the optimization of the response surface \hat{f} would result in a multi-objective optimization problem. Hence, an **acquisition function** $\alpha(\cdot)$ is used for the virtual optimization of the probabilistic model. The purpose of the acquisition function is two-fold: First, it scalarizes the response surface (which is a probability distribution) onto a single function, the acquisition surface $\alpha(\boldsymbol{\theta})$, such that it can be optimized¹. Thereby, the maximization of the response surface from Eq. (2) can be rephrased as the maximization of the acquisition surface

$$\boldsymbol{\theta}^* \in \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^d} \alpha(\boldsymbol{\theta}) . \quad (3)$$

Second, the GP expresses model uncertainty, which is used to trade off exploration and exploitation in the optimization. This trade off between exploration and exploitation, and therefore the model uncertainty, is extremely important for the optimization process when we have only few function evaluations. For an example of the optimization process of Bayesian optimization see Fig. 3.

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

Algorithm 1 summarizes the main steps of Bayesian optimization: A GP model for the (unknown) objective function $f : \boldsymbol{\theta} \mapsto f(\boldsymbol{\theta})$ is learned from the data set $\mathbb{D} = \{\boldsymbol{\theta}, f(\boldsymbol{\theta})\}$ composed of the parameters $\boldsymbol{\theta}$ and the corresponding measurements $f(\boldsymbol{\theta})$ of the true objective function (Line 4 of Algorithm 1). This model is used to predict the response surface \hat{f} and the corresponding acquisition surface $\alpha(\boldsymbol{\theta})$. Using a global optimizer the maximum $\boldsymbol{\theta}^*$ of the acquisition surface $\alpha(\boldsymbol{\theta})$ is determined (Line 5 of Algorithm 1) without any evaluation of the true objective function, e.g., no robot interaction is required, see Eq. (3). The parameters $\boldsymbol{\theta}^*$ are evaluated on the robot (Line 6 of Algorithm 1) and, together with the resulting measurement $f(\boldsymbol{\theta}^*)$, added to the data set \mathbb{D} (Line 7 of Algorithm 1). Note that the optimizer can be initialized by past evaluations for the data set \mathbb{D} (Line 1 of Algorithm 1), as well as by a prior of the GP model (Line 2 of Algorithm 1).

3.1 Gaussian Process Model for the Unknown Objective Function

To create the response surface model that maps $\boldsymbol{\theta} \mapsto f(\boldsymbol{\theta})$, we make use of Bayesian non-parametric GP regression [26]. A GP is a distribution over functions $f \sim \mathcal{GP}(m_f, k_f)$, fully defined by a prior mean m_f and a covariance function k_f . We assume a model where we observe noisy function values $y = f(\boldsymbol{\theta}) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$ is Gaussian noise. Both the prior mean m_f and the covariance function k_f are usually selected based on expert knowledge. Commonly used covariance functions include the squared exponential and Matérn covariance functions. In our experiments, we choose as prior mean $m_f \equiv \mathbf{0}^2$, while the chosen covariance function k_f is the squared exponential with automatic relevance determination and Gaussian noise

$$k_f(\boldsymbol{\theta}_p, \boldsymbol{\theta}_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)^T \boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)\right) + \sigma_\epsilon^2 \delta_{pq} \quad (4)$$

with $\boldsymbol{\Lambda} = \text{diag}([l_1^2, \dots, l_D^2])$. Here, l_i are the characteristic length-scales, σ_f^2 is the variance of the latent function $f(\cdot)$ and σ_ϵ^2 the noise variance. The explicit consideration of the measurement noise is important in robotic applications, although common optimization algorithms consider the measurements noise free.

Given n training inputs $\mathbf{X} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n]$ and corresponding training targets $\mathbf{y} = [y_1, \dots, y_n]$, we define the training data set $\mathbb{D} = \{\mathbf{X}, \mathbf{y}\}$. Hence, the GP predictive distribution is

$$p(f(\boldsymbol{\theta})|\mathbb{D}, \boldsymbol{\theta}) = \mathcal{N}\left(\mu(\boldsymbol{\theta}), \sigma^2(\boldsymbol{\theta})\right), \quad (5)$$

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

$$\mu(\boldsymbol{\theta}) = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{y}, \quad \sigma^2(\boldsymbol{\theta}) = k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{k}_*, \quad (6)$$

respectively, where \mathbf{K} is the matrix with $K_{ij} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, $k_{**} = k(\boldsymbol{\theta}, \boldsymbol{\theta})$ and $\mathbf{k}_* = k(\mathbf{X}, \boldsymbol{\theta})$.

A practical issue in Bayesian optimization and GP modeling is the selection of the hyperparameters of the GP model. The hyperparameters of a GP model are the parameters of the covariance function, i.e., the characteristic length-scales l_i , the

² A more informative prior can be used if expert knowledge is available.

variance of the latent function σ_f^2 and the noise variance σ_ϵ^2 . In gait optimization, these hyperparameters are often fixed a priori [9]. In [23], it is suggested that fixing the hyperparameters can considerably speed up the convergence of BO. However, manually tuning the hyperparameters requires extensive expert knowledge about the system that we want to optimize, which is often not available. Therefore, in this article we automatically select the hyperparameters by optimizing the marginal likelihood [26].

3.2 Acquisition Function

A number of acquisition functions $\alpha(\cdot)$ have been proposed, such as probability of improvement [4], expected improvement [27], upper confidence bound [28] and the recent entropy-based improvements [29]. All acquisition functions incorporate both the mean μ and the variance σ^2 of the GP prediction and result in different trade-offs between exploration and exploitation. Experimental results [29] suggest that expected improvement on specific families of artificial functions performs better on average than probability of improvement and upper confidence bound. However, these experiments required a good prior knowledge of the objective functions (e.g., the correct covariance function to use). This assumption does not necessarily hold for real-world problems, such as gait optimization. Probability of improvement [9], expected improvement [10] and upper confidence bound [13] have all been previously employed in gait optimization. However, only one experimental comparison has been carried out in gait optimization [14], and it is still unclear whether one of them should be preferred.

Probability of Improvement (PI). Introduced by Kushner [4], the acquisition function PI is defined as

$$\alpha(\boldsymbol{\theta}) = \Phi\left(\frac{\mu(\boldsymbol{\theta}) - T}{\sigma(\boldsymbol{\theta})}\right), \quad (7)$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and T the target value. The target value T is often the maximum of all explored data plus, optionally, a positive constant (for a study of its effects, see [23]). PI is a function bounded by the interval $[0, 1]$. Hence, since the normal cumulative distribution function is monotonically increasing, to maximize PI it is sufficient to maximize

$$\alpha(\boldsymbol{\theta}) = (\mu(\boldsymbol{\theta}) - T)/\sigma(\boldsymbol{\theta}). \quad (8)$$

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

Expected Improvement (EI). Mockus [27] introduced EI, which can be considered an extension of probability of improvement. The EI acquisition function is

$$\alpha(\boldsymbol{\theta}) = \sigma(\boldsymbol{\theta})[u\Phi(u) + \phi(u)]; \quad u = (\mu(\boldsymbol{\theta}) - T)/\sigma(\boldsymbol{\theta}), \quad (9)$$

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

Upper Confidence Bound (UCB). UCB [28] is defined as

$$\alpha(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta}) + \kappa\sigma(\boldsymbol{\theta}). \quad (10)$$

The choice of the free parameter κ is crucial as it determines the trade-off rate between exploration and exploitation. A special case of UCB is GP-UCB [30] where κ is automatically computed according to

$$\kappa = \sqrt{2 \log \left(\frac{n^{d/2+2} \pi^2}{3\delta} \right)}, \quad (11)$$

where n is the number of past evaluations of the objective function f , $\delta \in (0, 1)$ is a parameters and d the dimensionality of the parameters $\boldsymbol{\theta}$. Automatically selecting κ using GP-UCB allows to estimate regret bounds [30].

3.3 Optimizing the Acquisition Surface

Once the acquisition surface $\alpha(\boldsymbol{\theta})$ is computed, it is necessary to find its maximum

$$\boldsymbol{\theta}^* \in \arg \max_{\boldsymbol{\theta} \in \mathbb{R}^d} \alpha(\boldsymbol{\theta}). \quad (12)$$

This is still a global optimization problem, but considerably easier compared to the original global optimization problem defined in Eq. (1):

- The measurements in Eq. (12) are noise free since the objective function in Eq. (10) is an analytical model. This allows to use also global optimization algorithms, which do not consider stochasticity.
- There is no experimental restriction on how often we evaluate α : Evaluating the acquisition surface only requires interactions with the model, but not with a physical system, such as a robot. Thus, evaluating α only requires computations. Hence, optimization methods that require thousands or millions of evaluations can be employed to find the global maximum of α .
- We can compute the gradients of α of any order, either with finite differences or analytically and, therefore, use first or second-order optimization methods.

Therefore, virtually any global optimizer can be used to find the maximum $\boldsymbol{\theta}^*$ of α . Common choices are DIRECT [31] to find an approximate global maximum followed by L-BFGS [32] or CMA-ES [33] to refine it. In our experiments, we use DIRECT and L-BFGS.

4 Evaluation and Comparisons

We experimentally compare Bayesian optimization with different acquisition functions and other baseline optimization methods. First, we perform a feasibility study on a simulated stochastic linear-quadratic regulator. We compare the solution found by Bayesian optimization with the optimal solution of this classical stochastic optimal control problem. Second, we perform an experimental comparison on a real bipedal robot where we find gait parameters that maximize the walking speed in real and noisy conditions.

Table 2: Performance of Bayesian optimization compared to the exact solution for the stochastic LQR problem.

| | |
|---|-----------------|
| Utility incurred by the analytical solution | 5.57 ± 0.01 |
| Utility incurred by Bayesian optimization | 5.54 ± 0.01 |

4.1 Stochastic Linear-Quadratic Regulator

The linear-quadratic regulator is a classical stochastic optimal control problem. The discrete-time stochastic LQR problem consists of a linear dynamical system

$$\mathbf{x}_{t+1} = \mathbf{A}_t \mathbf{x}_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t, \quad t = 0, 1, \dots, N-1, \quad (13)$$

and a quadratic cost

$$J = \mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N + \sum_{t=0}^{N-1} \left(\mathbf{x}_t^T \mathbf{Q}_t \mathbf{x}_t + \mathbf{u}_t^T \mathbf{R}_t \mathbf{u}_t \right), \quad (14)$$

where $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ is Gaussian system noise and the matrices $\mathbf{R}_t > \mathbf{0}$, $\mathbf{Q}_t \geq \mathbf{0}$, \mathbf{A}_t , \mathbf{B}_t are given and assumed to be time invariant. The objective is to find the control sequence $\mathbf{u}_0, \dots, \mathbf{u}_{N-1}$ that minimizes Eq. (14). The control signal \mathbf{u}_t is a linear function of the state \mathbf{x}_t , computed for each time step as

$$\mathbf{u}_t = \mathbf{L}_t \mathbf{x}_t,$$

where \mathbf{L}_t is a gain matrix. Using the algebraic Riccati equation, the optimal gain matrix \mathbf{L}_t can be computed such that the quadratic cost J for the stochastic linear-quadratic regulator is minimized [34].

To assess the performance of Bayesian optimization, we consider a stochastic LQR system with $\mathbf{x} \in \mathbb{R}^2$, $\mathbf{u} \in \mathbb{R}^4$. The stationary gain matrix $\mathbf{L} \in \mathbb{R}^{4 \times 2}$ defines a set of 8 free parameters to be determined by Bayesian optimization. We compare our solution with the corresponding analytical solution for the stationary gain matrix \mathbf{L} . For Bayesian optimization, we define the objective function as the utility

$$f(\boldsymbol{\theta}) = -\log(J/N), \quad (15)$$

where the parameters $\boldsymbol{\theta}$ to optimize are the stationary gain matrix $\mathbf{L} \in \mathbb{R}^{4 \times 2}$. We initialized Bayesian optimization with 15 uniformly randomly sampled gain matrices \mathbf{L} . The initial state \mathbf{x}_0 was sampled from a standard normal $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

We performed 50 independent experiments: For each experiment, we selected the best parameters found after 200 steps of Bayesian optimization. These parameters were then evaluated on the stochastic LQR system 100 times. Table 2 shows the mean value for the objective function and its standard deviation for both the analytical solutions and the ones obtained through Bayesian optimization. The table shows that BO finds near-optimal solutions for the stochastic LQR problem. Additionally, as shown in Fig. 4, the average over the 50 experiments of the best parameters found so far in the optimization process suggests that Bayesian optimization reliably quickly finds a near-optimal solution.

Fig. 5 shows an example of the maximization process of BO for the stochastic LQR problem. The objective function is displayed as a function of the number of evaluations. Each evaluation requires to compute the objective function f in Eq. (15) for the current parameters $\theta = \mathbf{L}$. The analytical maximum is shown by the green dashed line, the shaded area shows the 95% confidence bound of the predicted objective function $p(f(\theta))$ for the parameters selected in the i th evaluation. The red line shows the actual measured function value $f(\theta)$. Initially, the model was relatively uncertain. With an increasing number of experiments the model became more confident, and the optimization process converged to the optimal solution.

We conclude that BO can efficiently find gain matrices \mathbf{L} that solve the stochastic LQR problem. Additionally, with Bayesian optimization it is possible to find stationary solutions for cases with a short time horizon N where no analytical optimal solution is available: The algebraic Riccati equation does not yield the optimal solution for finite-horizon problems, and the discrete-time Riccati equation, does not yield stationary solutions.

4.2 Gait Optimization of a Bio-Inspired Biped

To validate our Bayesian gait optimization approach, we used the dynamic bipedal walker *Fox*, shown in Fig. 1. The walker is mounted on a boom that enforces planar, circular motion. This robot consists of a trunk, two legs made of rigid segments connected by knee joints to telescopic leg springs, and two spheric feet with touch sensors [35]. *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, as shown

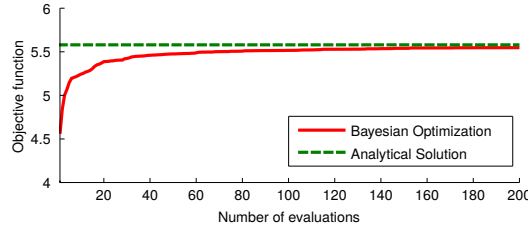


Fig. 4: Average over 50 experiments of the best parameters found during the maximization process of a stochastic LQR using Bayesian optimization. The average objective value function (red curve) during the optimization process and the average analytical solution (green dashed line) are shown.

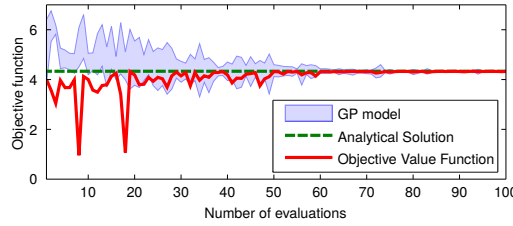


Fig. 5: Example of Bayesian optimization of a stochastic LQR. The objective value function (red curve) and the 95% confidence of the model prediction (blue area) are shown during the optimization process; the analytical solution (green dashed line) is shown as a reference.

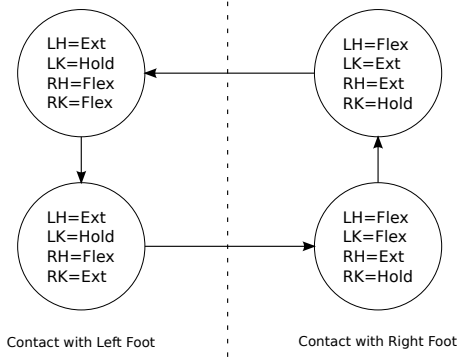


Fig. 6: The *Fox* controller is a finite state machine with four states. Each of the four joints, left hip (LH), left knee (LK), right hip (RH) and right knee (RK), can perform one of three actions: flexion (Flex), extension (Ext) or holding (Hold). When a joint reaches the maximum extension or flexion, its state is changed to holding. The transition between the states and the control signals applied during flexion and extension are determined by the controller parameters θ .

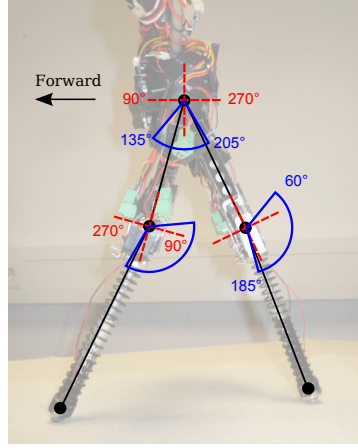


Fig. 7: Hip and knee angle reference frames (red dashed) and rotation bounds (blue solid). The hip joint angles' range lies between 135° forward and 205° backward. The knee angles range from 185° when fully extended to 60° when flexed backward.

in Fig. 7. The touch sensors return binary ground contact signals. 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)**, shown in Fig. 6, with four states: two for the swing phases of each leg [36]. These states control the actions performed by each of the four actuators, which were extension, flexion or holding of the joint. The transitions between the states are regulated by thresholds based on the angles of the joints. For the optimization process, we identified eight parameters of the controller that are crucial for the resulting gait. These gait parameters consist of four threshold values of the FSM (two for each leg) and the four control signals applied during extension and flexion (separately for knees and hips). It is important to notice that a set of parameters that proved to be efficient with a set of motors could be ineffective with a different set of motors (e.g., if one motor is replaced), due to slightly different mechanical properties. Therefore, automatic and fast gait optimization techniques are essential for this robot.

4.2.1 Gait Optimization Comparison

In the first experiment we compare the performances of Uniform Random Search, Grid Search and Bayesian optimization with the three different acquisition functions introduced in Section 3.2. For BO, we also consider the two separate cases when the hyperparameters are automatically optimized, and when they are manually set by an expert. In this experiment, we optimize the four threshold parameters

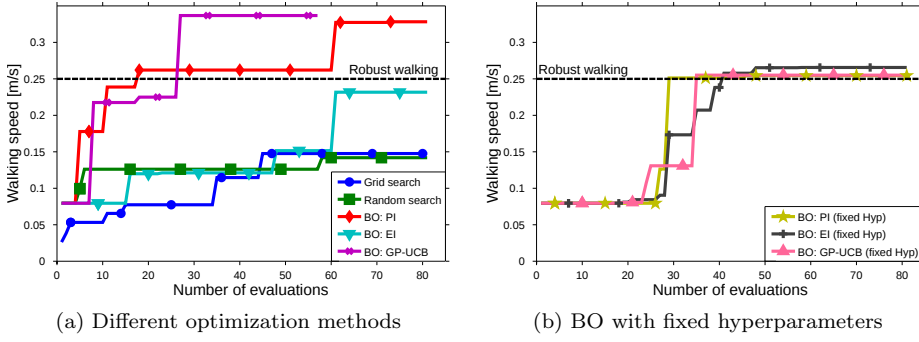


Fig. 8: The maximum walking speed of *Fox* evaluated during the gait optimization process. (a) BO performed better than both grid and random search. BO with the GP-UCB acquisition function performed best, achieving a fast and robust gait in less than 30 experiments. (b) BO of various acquisition functions for fixed hyperparameters. Manually fixed hyperparameters led to sub-optimal solutions for all the acquisition functions.

of the controller θ with respect to the objective function

$$f(\theta) = \frac{1}{N} \sum_{i=1}^N V_i(\theta), \quad (16)$$

i.e., the average walking velocities V_i in the given time horizon over $N = 3$ experiments. 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. Based on preliminary experiments we knew that the variance of the walking velocities across multiple experiments was not constant for all parameters (i.e., homoscedastic), but heteroscedastic. This was particularly pronounced for parameters that lead to unstable, but occasionally very fast gaits. Since standard GP models assume homoscedastic noise, averaging over N experiments proved helpful to reduce (but not eliminate) the heteroscedasticity of the objective function, such that we could still use a homoscedastic model. Each experiment was initialized from similar initial configurations and lasted 12 seconds, starting from the moment when the foot of the robot initially touched the ground. To initialize BO, three uniformly randomly sampled parameter sets were used.

The maximum walking speed of *Fox* evaluated during the gait optimization process for the different methods is shown in Fig. 8a. 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 immediately. Values between 0.1 m/s and 0.15 m/s indicate that the robot could walk for multiple steps but showed systematic falls thereafter. 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 cause a fall. From the results we see that both grid search and random search performed poorly, finding a maximum, such that the robot could only limp. We noticed that Bayesian optimization performed considerably better with all acquisition functions. BO using PI and GP-UCB achieved robust gaits with a similar walking speed, while GP-UCB was

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

Fig. 9: Maximum average walking speeds [m/s] found by the different optimization methods. BO using GP-UCB with automatic hyperparameter selection found the best maximum of all methods. The maximum obtained by PI is qualitatively similar.

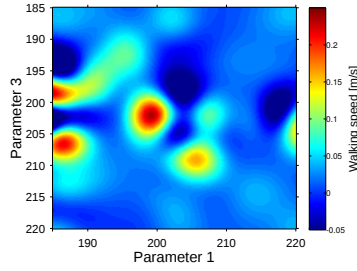


Fig. 10: Two-dimensional slice of the mean of the response surface computed using all the experiments performed. Avoiding the local maxima requires global optimization.

slightly faster in finding the best gait parameter settings. On the other hand, using EI did not lead to robust gaits. The reason of this result were the inaccuracies of the model of the underlying objective function: The automatically selected hyperparameters had overly long length-scales (see Eq. (4)), 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 [29], which suggest that EI on specific families of synthetic 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. In particular, during the empirical evaluation, we observed that EI explored the parameter space insufficiently well. In turn, this insufficient exploration resulted in overly long length-scales and an inappropriate GP model. Therefore, we hypothesize that EI suffers more from inaccurate models than PI and UCB, which tend to explore more aggressively.

As a second comparison, we studied the effects of manually fixing the GP hyperparameters to reasonable values, based on our expert knowledge. Fig. 8b shows the performance of BO with the different acquisition functions with fixed hyperparameters. All acquisition functions found similar sub-optimal solutions with fixed hyperparameters since 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, thus, the optimization process was hindered. To confirm this hypothesis, using all the evaluations performed, we trained a GP model and optimized the hyperparameters using the marginal likelihood. Some of the resulting values of the hyperparameters were half of the manually selected values, which suggests 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 hyperparameter selection because the longer length-scales limited the exploration of these acquisition functions. In contrast, for EI the use of fixed hyperparameters was beneficial since 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

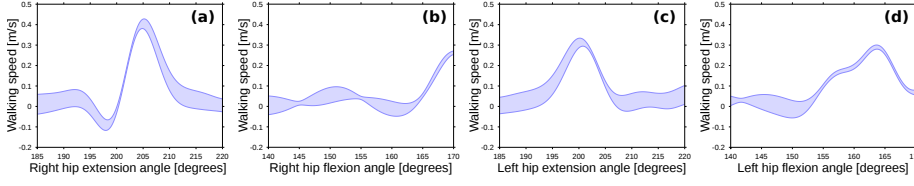


Fig. 11: Slices of the response surface along each parameter dimension. The response surface was trained with the data collected across all the experiments. Only a small zone of each slice leads to a walking gait. The symmetry between parameters of the left and right leg is clear, see (a)/(c) and (b)/(d). The visible shift of about 5 degrees in the optimal parameters is due to the constrained circular walk.

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 harm the optimization process by limiting the exploration and leading to sub-optimal solutions.

Parameters corresponding to the best function value observed do not always correspond to the real optimum, due to the presence of (measurement) noise. On the other hand using the best parameters obtained from the model should take into consideration the noise, and, therefore, achieve better results on average. However, using the parameters obtained from the model assumes sufficient correctness of the model, which might not always be the case. When possible, at the end of each trial, we additionally evaluated both the parameters with the best function value observed so far, and the best parameters obtained from the model. The results consistently showed that these two sets of parameters were almost identical and led to a similar performance, therefore, validating the goodness of the model. Table 9 compares the optimum found by all evaluated methods.

During the experiments, we observed and empirically estimated that variations of up to 0.04 m/s can depend on the presence of noise in the experiments. Additionally, it should be noted that the real noise of the objective function is neither Gaussian nor homoscedastic. Configurations of the parameters that produce periodic falls after a single step or stable gaits, typically behave consistently across various experiments and result in a smaller noise (0.01 m/s). On the other hand for parameters that produce unstable gaits with occasional falls the noise can be larger, typically up to 0.04 m/s. A two-dimensional slice of the response surface computed using the data collected from all the over 1800 evaluations is shown in Fig. 10. This optimization surface is complex and non-convex, and, therefore, unsuitable for local optimization methods (e.g., gradient-based methods). In Fig. 11, similar results are shown by taking slices of the parameter space. Notably, a symmetry between the parameters of the two legs is visible, except for a shift of about 5 degrees. This asymmetry can be explained by the smaller radius of the walking circle for the inner leg. This slight asymmetry also reflected in the best parameters found during the optimization.

Running a single trial for each experiment limits the reliability of our results. However, based on our experience with the system we believe that these results indicate a trend. In particular, the results obtained highlight the important issue that on a real system the performance of the different acquisition functions might

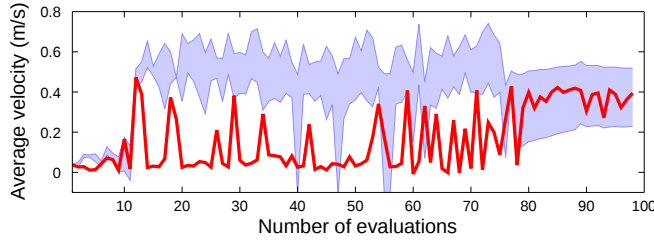


Fig. 12: Average walking speed during the gait optimization process of *Fox* using Bayesian optimization. For each evaluation the measurement of the objective function $f(\theta^*)$ (red curve) and the corresponding 95% confidence of the model prediction $\hat{f}(\theta^*)$ (blue area) are shown. Three evaluations are used to initialize BO (not shown in the plot). After 80 evaluations, BO finds an optimum corresponding to a stable walking gait with an average speed of 0.45 m/s.

differ compared to the typical analytical benchmarks. The main reason is that we can perfectly model the objective functions in these benchmarks, which is not the case for real-world systems that can violate typical assumptions of GP models (e.g., smoothness, homoscedasticity), such as on the *Fox* robot. In this context, our results suggest that GP-UCB is more robust to model inaccuracies.

The GP modeling capabilities are often overlooked when evaluating the performance of different acquisition functions in Bayesian optimization. Based on the results of our experimental evaluation, we speculate that there exists a fairly unexplored connection between the exploration properties of the acquisition function and the capabilities of GP modeling, i.e., the GP prior. The performance of an acquisition function depends on the quality of the GP model, and vice versa proper modeling takes place only when the acquisition function explores a sufficiently rich parameter set.

4.2.2 Including Control Signals in Gait Optimization

We further evaluated the performance of BO with GP-UCB (the best performing acquisition function in the previous experiment) by increasing the dimensionality of the optimization problem. To the four threshold parameters, we add other four parameters corresponding to the voltage provided to the motors, for a total of eight parameters. As objective function we again use the velocity defined in Eq. (16).

In Fig. 12, the Bayesian optimization process for gait learning is shown. Initially, the learned GP model could not adequately capture the underlying objective function. Average velocities below 0.1 m/s typically indicate a fall of the robot after the first step. Large parts of the first 60 experiments were spent on learning that the control signals applied at the hips had to be sufficiently high in order to swing the leg forward (i.e., against gravity and friction). Once this knowledge was acquired, the produced gaits typically led to walking but were rather unstable, and the robot fell after few steps. After 80 experiments, the model became more accurate (the function evaluations shown in red were within the 95% confidence bound of the prediction), and Bayesian optimization found a stable walking gait. The resulting gait³ was evaluated for a longer period of time, and it proved sufficiently

³ Videos are available at <http://www.ias.tu-darmstadt.de/Research/Fox>.

robust to walk continuously for 2 minutes without falling, while achieving an average velocity of 0.45 m/s. This average velocity was higher than the performance obtained by optimizing only four parameters and was close to the maximum velocity this hardware set-up can achieve [35]. The best parameters obtained through Bayesian optimization were slightly asymmetrical for the two legs.

From our experience with the biped *Fox*, hand-tuning the gait parameters can be a very time-consuming process of many days. Only around 1% of the considered parameter space leads to walking gaits, and the influence and the interaction of the parameters is not trivial. Moreover, expert manual parameter search yielded inferior gaits compared to the ones obtained by Bayesian optimization, in both walking velocity and robustness. Automatic gait parameter selection by means of Bayesian optimization sped up the parameter search from days to hours.

5 Conclusion

Automatic gait optimization is a key challenge that needs to be addressed to deploy bipedal walkers in real-world applications. The principal limitation of most common optimization methods is the need to be experimentally-efficient while accounting for various sources of uncertainty in each experiment, including measurement noise, model uncertainty and stochasticity of the robot and environment. In this article, we evaluated Bayesian optimization on a bio-inspired bipedal walker. Due to a probabilistic surrogate model, BO is an efficient solution to real-world gait optimization. Performing over 1800 experiments, we compared different variants of BO and observed that the GP-UCB acquisition function performed best.

References

1. Chernova, S., Veloso, M.: An evolutionary approach to gait learning for four-legged robots. In: Intelligent Robots and Systems (IROS). Volume 3., IEEE (2004) 2562–2567
2. Gibbons, P., Mason, M., Vicente, A., Bugmann, G., Culverhouse, P.: Optimisation of dynamic gait for small bipedal robots. In: Proc. 4th Workshop on Humanoid Soccer Robots (Humanoids 2009). (2009) 9–14
3. Kulk, J., Welsh, J.: Evaluation of walk optimisation techniques for the NAO robot. In: Humanoids 2011. (2011) 306–311
4. Kushner, H.J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering* **86** (1964) 97
5. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization* **21** (2001) 345–383
6. Osborne, M.A., Garnett, R., Roberts, S.J.: Gaussian processes for global optimization. In: Learning and Intelligent Optimization (LION). (2009) 1–15
7. Brochu, E., Cora, V.M., De Freitas, N.: 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)
8. Garnett, R., Osborne, M.A., Roberts, S.J.: Bayesian optimization for sensor set selection. In: Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks. IPSN '10, New York, NY, USA, ACM (2010) 209–219
9. Lizotte, D.J., Wang, T., Bowling, M., Schuurmans, D.: Automatic gait optimization with Gaussian process regression. In: International Joint Conference on Artificial Intelligence (IJCAI). (2007) 944–949
10. Tesch, M., Schneider, J., Choset, H.: Using response surfaces and expected improvement to optimize snake robot gait parameters. In: International Conference on Intelligent Robots and Systems (IROS), IEEE (2011) 1069–1074

11. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: *Learning and Intelligent Optimization (LION)*. Springer (2011) 507–523
12. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: *Advances in Neural Information Processing Systems (NIPS)*. (2012)
13. Calandra, R., Seyfarth, A., Peters, J., Deisenroth, M.P.: An experimental comparison of Bayesian optimization for bipedal locomotion. In: *International Conference on Robotics and Automation (ICRA)*. (2014)
14. Calandra, R., Gopalan, N., Seyfarth, A., Peters, J., Deisenroth, M.P.: Bayesian gait optimization for bipedal locomotion. In: *Learning and Intelligent Optimization (LION)*. (2014) 274–290
15. Brooks, S.H.: A discussion of random methods for seeking maxima. *Operations Research* **6** (1958) 244–251
16. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. *Journal of Machine Learning Research (JMLR)* **13** (2012) 281–305
17. Yamane, K.: Geometry and biomechanics for locomotion synthesis and control. In: *Modeling, Simulation and Optimization of Bipedal Walking*. Volume 18 of *Cognitive Systems Monographs*. Springer (2013) 273–287
18. Tedrake, R., Zhang, T., Seung, H.: Stochastic policy gradient reinforcement learning on a simple 3D biped. In: *International Conference on Intelligent Robots and Systems (IROS)*. (2004) 2849–2854
19. Tang, Z., Zhou, C., Sun, Z.: Humanoid walking gait optimization using ga-based neural network. In: *Advances in Natural Computation*. Volume 3611 of *Lecture Notes in Computer Science*. Springer (2005) 252–261
20. Niehaus, C., Röfer, T., Laue, T.: 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. Hemker, T., Stelzer, M., von Stryk, O., Sakamoto, H.: Efficient walking speed optimization of a humanoid robot. *International Journal of Robotics Research (IJRR)* **28** (2009) 303–314
22. Geng, T., Porr, B., Wörgötter, F.: Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *International Journal of Robotics Research (IJRR)* **25** (2006) 243–259
23. Lizotte, D.J., Greiner, R., Schuurmans, D.: An experimental methodology for response surface optimization methods. *Journal of Global Optimization* **53** (2012) 699–736
24. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13** (1998) 455–492
25. Huang, D., Allen, T.T., Notz, W.I., Zeng, N.: Global optimization of stochastic black-box systems via sequential kriging meta-models. *Journal of Global Optimization* **34** (2006) 441–466
26. Rasmussen, C.E., Williams, C.K.I.: *Gaussian Processes for Machine Learning*. The MIT Press (2006)
27. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. *Towards Global Optimization* **2** (1978) 117–129
28. Cox, D.D., John, S.: SDO: A statistical method for global optimization. *Multidisciplinary design optimization: state of the art* (1997) 315–329
29. Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. *Journal of Machine Learning Research (JMLR)* **13** (2012) 1809–1837
30. Srinivas, N., Krause, A., Kakade, S., Seeger, M.: Gaussian process optimization in the bandit setting: No regret and experimental design. In: *International Conference on Machine Learning (ICML)*, Omnipress (2010) 1015–1022
31. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications* **79** (1993) 157–181
32. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* **16** (1995) 1190–1208
33. Hansen, N., Ostermeier, A.: Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation* **9** (2001) 159–195
34. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*. 3rd edn. Athena Scientific (2007)
35. Renjewski, D.: An engineering contribution to human gait biomechanics. PhD thesis, TU Ilmenau (2012)
36. Renjewski, D., Seyfarth, A.: Robots in human biomechanics - a study on ankle push-off in walking. *Bioinspiration & Biomimetics* **7** (2012) 036005