Bayesian Gait Optimization for Bipedal Locomotion

Roberto Calandra[1], Nakul Gopalan[2], André Seyfarth[3],
Jan Peters[1,4], and Marc Peter Deisenroth[5,1]

[1] TU Darmstadt, Dept. of Computer Science, Intelligent Autonomous Systems Lab, Germany
[2] Brown University, Dept. of Computer Science, USA
[3] TU Darmstadt, Dept. of Sport Science, Lauflabor Locomotion Lab, Germany
[4] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[5] Imperial College London, Dept. of Computing, UK

Abstract. One of the key challenges in robotic bipedal locomotion is finding gait parameters that optimize a desired performance metric, such as robustness or energy efficiency. Typically, gait optimization requires extensive robot experiments and specific expert knowledge. Instead, we propose to apply data-driven machine learning to automate and speed up the process of gait optimization. In particular, we use Bayesian optimization to efficiently find gait parameters that optimize the desired performance metric. As a proof of concept we demonstrate that Bayesian optimization is near-optimal in a classical stochastic optimal control framework. Moreover, we validate our approach to Bayesian gait optimization on a low-cost but sensitive real bipedal walker and show that good walking gaits can be efficiently found by Bayesian optimization.

Keywords: Bayesian optimization, gait optimization, bipedal locomotion

1 Introduction

Bipedal walking and running are versatile and fast locomotion gaits. 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 paper, we focus on *gait optimization*, i.e., finding good parameters for the gait of a robotic biped.

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 is either an educated guess by a human expert or a systematic search, such as grid search. As a result, gait optimization may require considerable expert knowledge, engineering effort and time-consuming experiments. Additionally, the effectiveness of the resulting gait is restricted by the assumptions made during the controller design process, regarding the environment, the hardware and the performance criteria. Therefore, a change in the environment (e.g., different floor surfaces), a variation in the hardware response (e.g., decline in performances of the hardware, replacement of a motor or differences in the calibration) or the choice of a performance criterion (e.g., walking speed, energy efficiency, robustness), which differs from the
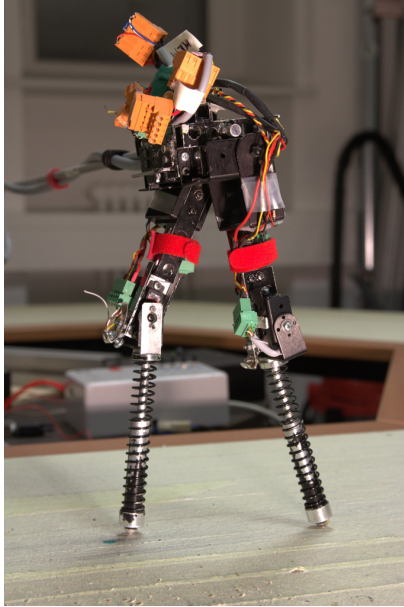
Fig. 1: The bio-inspired dynamical bipedal walker *Fox*. Using Bayesian optimization, we found reliable and fast walking gaits with a velocity of up to 0.45 m/s.

one used during the controller design process, often require searching for new, more appropriate, gait parameters.

The search for appropriate 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. Therefore, it is a valuable and principled approach to designing controllers and reduces the need for engineering expert knowledge. To date, automatic gait optimization methods have been used for designing efficient gaits for locomotion. Optimization methods used in the past for gait optimization include gradient descent methods [1] and genetic algorithms [2]. However, gradient descent based methods [1] might not find the optimal solutions for objective function with multiple local minima, and the computation of the gradient is required. Furthermore, most global optimization approaches are impractical to apply to sensitive robots as they require a large number of interactions with the real robot. For example, in genetic algorithms multiple sets of parameters from the population must be evaluated for each iteration, therefore, requiring impractical number of evaluations [2]. Since a large number of interactions can wear the robot out, extensive experiments may be economically infeasible or require an impractical amount of time. Hence, it is essential to reduce the number of interactions required to find good parameters.

To overcome this practical limitation on the number of possible interactions, we propose to use *Bayesian optimization* for efficient bipedal gait optimization. Bayesian optimization is a state-of-the-art global optimization method [3,4,5] 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 has been successfully applied to sensor-set selection [6] and gait optimization for quadrupeds [7] and snake robots [8]. Bayesian optimization makes efficient use of the past interactions by learning a probabilistic model of the function to optimize. Subsequently, the learned model is used for finding optimal parameters *without* the need to evaluate the expensive function again. By exploiting the learned model, Bayesian optimization, therefore, often requires fewer interactions than other optimization methods [4]. 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 model. More-

over, unlike most optimization methods, it can re-use any collected interaction data set, e.g., whenever we want to change the performance criterion.

In this paper, we demonstrate that Bayesian optimization is a promising approach for gait optimization. In Section 3.1, as a proof of concept, we apply Bayesian optimization to a well-studied stochastic optimal control task, i.e., stochastic Linear-Quadratic Regulation (LQR) [9], where a known analytical optimal solution is available. We demonstrate that Bayesian optimization successfully finds near-optimal solutions for the stochastic LQR problem quickly, reproducibly and reliably. In Section 3.2, we show that Bayesian optimization can be used for imitation of trajectories in the context of bipedal walking. Given a reference trajectory we find controller parameters that result in a gait that closely resembles the reference trajectory. In Section 3.3, we apply Bayesian optimization to gait optimization for robotic bipedal locomotion. Experimental results on the bio-inspired biped *Fox* (Figure 1) demonstrate that Bayesian optimization finds good gait parameters in a small number of experiments. Moreover, the learned controller results in a better gait compared to previous hand-crafted controllers. The use of an efficient gait optimization method for bipedal locomotion greatly alleviates the need for extensive parameter search and reduces the requirement of expert knowledge.

2 Efficient Gait Optimization

The search for appropriate parameters for a controller and/or trajectory representation can be formulated as an optimization problem, such as the minimization

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}} \, f(\boldsymbol{\theta}) \tag{1}$$

of an *objective function* f with respect to the parameters $\boldsymbol{\theta}$. In the case of gait optimization, $\boldsymbol{\theta}$ are the parameters of the gait controller, while the objective function f is a performance criterion, such as the walking speed, energy consumption or robustness. Note that 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.** When evaluating the objective function f the value of the function $f(\boldsymbol{\theta})$ is available, but not the gradient information $\mathrm{d}f(\boldsymbol{\theta})/\mathrm{d}\boldsymbol{\theta}$ with respect to the parameters. The use of gradient information is generally desirable in local optimization as it leads to faster convergence than zero-order methods. Thus, it is common 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 the whole family of efficient gradient descent-based methods (e.g., gradient descent, conjugate gradient, LBFGS [10]) undesirable for our task.

2. **Stochastic objective function.** The evaluation of the objective function is inherently stochastic due to noisy measurements and variable initial conditions. Therefore, any suitable optimization method needs to take into consideration that two evaluations of the same parameters $\boldsymbol{\theta}$ can yield two different values $f_1(\boldsymbol{\theta}) \neq f_2(\boldsymbol{\theta})$.

3. **Global solution.** Ideally, we strive to find the global minimum of the objective function. However, no assumption can be made about the presence of multiple local minima or about the convexity of the objective function.

All these characteristics make this family of problems a very challenging optimization task. A classical way of dealing with this family of problems is to evaluate the objective function f at an evenly-spaced grid in the parameter space. Sequentially, the grid search is refined in the most promising intervals of the space. Another possibility is to use random search, which can perform well [11], e.g., when the objective function has an intrinsic lower dimensionality. However, both methods typically require an impractical number of function evaluations/robot interactions to find good gait parameters. In contrast, Bayesian optimization [3] naturally deals with this family of optimization problems and finds solutions in a small number of evaluations of the objective function.

2.1 Bayesian Optimization

Bayesian optimization, as summarized in Algorithm 1, is an iterative model-based global optimization method [3,4,5,12,13]. After each evaluation of the objective function f, a model of f is built (line 3 of Algorithm 1). In particular, the model maps parameters $\boldsymbol{\theta}$ to corresponding function evaluation $f(\boldsymbol{\theta})$. From the resulting model the *response surface* $\hat{f}(\boldsymbol{\theta})$ is computed (line 4) and used for a "virtual" optimization process

---

Algorithm 1: Bayesian optimization

---

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

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}}\ \hat{f}(\boldsymbol{\theta})\,. \tag{2}$$

In this context, "virtual" indicates that optimizing the response surface $\hat{f}(\boldsymbol{\theta})$ with respect to the parameters $\boldsymbol{\theta}$ does not need interactions with the real system, but only evaluations of the learned model. Only when a new set of parameters $\boldsymbol{\theta}^*$ has been selected from the virtual optimization process of the response surface $\hat{f}(\boldsymbol{\theta})$, they are evaluated on the real objective function f (line 7). The new data $\{\boldsymbol{\theta}^*, f(\boldsymbol{\theta}^*)\}$ is used to update the model of the objective function (line 8).

A variety of different models, such as linear functions or splines [4], have been used in the past to map $\boldsymbol{\theta} \mapsto f(\boldsymbol{\theta})$. However, 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 a change in the system, e.g., after replacing a motor or changing the walking surface. In this paper, we use Gaussian processes (GPs) as the probabilistic model for the Bayesian optimization.
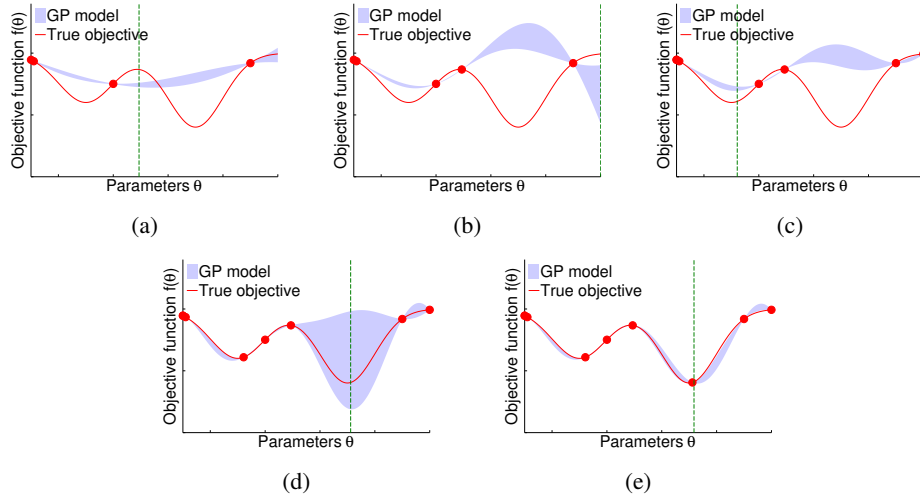
Fig. 2: Example of the Bayesian optimization process for minimizing an unknown 1-D objective function f (red curve). The 95% confidence of the model prediction is represented by the blue area. The optimization is initialized with 4 previously evaluated parameter $\boldsymbol{\theta}$. The value of the next parameter to be evaluated is represented by the green dashed line. At each iteration, the model is updated using all the previously evaluated parameters (red dots). After a few iterations, Bayesian optimization found the global minimum of the unknown objective function.

When using a probabilistic model, the response surface $\hat{f}(\boldsymbol{\theta})$ is a probability distribution and cannot directly be optimized. Instead, the *acquisition function* $\alpha(\cdot)$ is used for the virtual optimization of the probabilistic GP. The purpose of the acquisition function is two-fold. First, it maps the GP onto a single surface, the *acquisition surface* $\alpha(\boldsymbol{\theta})$ to be optimized.[6] Second, the GP expresses model uncertainty, which is used to trade off exploration and exploitation. Thereby, the minimization of the objective function from Equation (1) can be rephrased as the minimization of the acquisition surface

$$\underset{\boldsymbol{\theta} \in \mathbb{R}^d}{\text{minimize}}\, \alpha(\boldsymbol{\theta}). \tag{3}$$

As summarized in Algorithm 1, in Bayesian optimization, a GP model $\boldsymbol{\theta} \mapsto p(f(\boldsymbol{\theta}))$ is learned from the parameters $\boldsymbol{\theta}$ to the corresponding measurements $f(\boldsymbol{\theta})$ of the objective function (line 3 of Algorithm 1). This model is used to predict the response surface $\hat{f}(\boldsymbol{\theta})$ (line 4 of Algorithm 1) and the corresponding acquisition surface $\alpha(\boldsymbol{\theta})$ (line 5 of Algorithm 1), once the response surface $\hat{f}(\boldsymbol{\theta})$ is mapped through the acquisition function α. Using a global optimization technique, the minimum $\boldsymbol{\theta}^* = \text{argmin}_\theta\, \alpha(\boldsymbol{\theta})$ of the acquisition surface α is computed (line 6 of Algorithm 1) without any evaluation of the objective function, e.g., no robot interaction, see Equation (3). The current minimum $\boldsymbol{\theta}^*$ is

---

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

evaluated (line 7 of Algorithm 1) and, together with the resulting measurement $f(\boldsymbol{\theta}^*)$, added to the dataset $\mathbb{D}$ (line 8 of Algorithm 1). Additionally, past evaluations can be used to initialize the dataset $\mathbb{D}$ (line 1 of Algorithm 1), as well as the prior of the GP model (line 2 of Algorithm 1).

Figure 2 illustrates the Bayesian optimization process for a 1-D function. The horizontal axis represents the parameter space. The red curve shows the true, but unknown, objective function f and the blue area represents the 95% confidence bound of the GP model of f. The GP model is trained on a small data set, represented by the red dots. From this model the acquisition function is computed. The minimum of the acquisition function determines the next parameter set $\boldsymbol{\theta}$ to be evaluated (dashed green line). Subsequently, the GP model of the objective function is updated, and the process is restarted. After a few iterations, Bayesian optimization found the global minimum.

2.2 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 [14]. Such a GP is a distribution over functions

$$f(\boldsymbol{\theta}) \sim \mathrm{GP}\left(m_f, k_f\right) \tag{4}$$

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

$$k(\boldsymbol{\theta}_p, \boldsymbol{\theta}_q) = \sigma_f^2 \exp(-\tfrac{1}{2}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)^T \boldsymbol{\Lambda}^{-1}(\boldsymbol{\theta}_p - \boldsymbol{\theta}_q)) + \sigma_w^2 \delta_{pq}$$

with $\boldsymbol{\Lambda} = \mathrm{diag}([l_1^2, ..., l_D^2])$. Here, l_i are the characteristic length-scales, σ_f^2 is the variance of the latent function f and σ_w^2 the measurement noise variance. The GP predictive distribution at a test input $\boldsymbol{\theta}_*$ is

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

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

Given n training inputs $\boldsymbol{X} = [\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n]$ and corresponding training targets $\boldsymbol{y} = [f(\boldsymbol{\theta}_1), ..., f(\boldsymbol{\theta}_n)]$, we define the training data set $\mathbb{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$. Moreover, $\boldsymbol{K}$ is the matrix composed as $K_{ij} = k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, $k_{**} = k(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*)$ and $\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{\theta}_*)$. In our experiments, we compute the hyperparameters of the covariance function by maximum a posteriori (MAP) estimates [14].

2.3 Acquisition Function

A number of acquisition functions $\alpha(\boldsymbol{\theta})$ exist, such as probability of improvement [3], expected improvement [15], upper confidence bound [16] and entropy-based improvements [17]. In this paper, we use the *upper confidence bound (UCB)* where the acquisition surface is defined as

$$\alpha(\boldsymbol{\theta}) = \mu(\boldsymbol{\theta}) - \kappa\, \sigma(\boldsymbol{\theta}), \tag{7}$$

where κ is a free parameter that trades off exploration and exploitation. We determine κ automatically according to the GP-UCB [18,19] algorithm, which also allows to compute regret bounds. An extensive comparison of other acquisition functions with the biped considered in Section 3.3 can be found in [20].

2.4 Optimizing the Acquisition Surface

Once the acquisition surface in Equation (7) 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 (3) is different from the original global optimization problem defined in Equation (1). First, the measurements in Equation (3) are noise free because the objective function in Equation (7) is an analytical model. Second, there is no restriction in terms of how many evaluations we can perform: Evaluating the acquisition surface only requires to evaluate the model, but no interactions with the physical system (e.g., the robot). Third, we can compute the derivatives of any order, either with finite differences or analytically. Therefore, we are no longer restricted to the use of zero-order optimization methods. As a result, any global optimizer that fulfills these characteristics can be used. In particular, in our experiments we used DIRECT [21] to find the approximate global minimum, followed by LBFGS [10] to refine it.

3 Experimental Setup & Results

In this section, we present the experiments performed and results obtained to validate Bayesian optimization for automatic gait optimization. First, we evaluate Bayesian optimization on a classical stochastic optimal control problem: a discrete-time stochastic linear-quadratic regulator (LQR). Since an optimal solution to the stochastic LQR system can be computed analytically, we evaluate the quality of the solution found by Bayesian optimization to this baseline. Second, we apply Bayesian optimization to a trajectory imitation problem in the context of bipedal walking. Given a reference trajectory, we demonstrate that Bayesian optimization finds suitable parameters of rhythmic motor primitives (RMPs) to replicate the trajectory. We consider the case of demonstrated gait trajectories of a simulated biped. Third, we present and discuss the experimental results of Bayesian optimization applied to gait optimization for bipedal locomotion on the robot shown in Figure 1.

3.1 Proof of Concept: 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

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}_t \boldsymbol{x}_t + \boldsymbol{B}_t \boldsymbol{u}_t + \boldsymbol{w}_t, \qquad t = 0, 1, ..., N-1, \tag{8}$$

and a quadratic cost

$$J = \boldsymbol{x}_N^T \boldsymbol{Q}_N \boldsymbol{x}_N + \sum_{t=0}^{N-1} \left(\boldsymbol{x}_t^T \boldsymbol{Q}_t \boldsymbol{x}_t + \boldsymbol{u}_t^T \boldsymbol{R}_t \boldsymbol{u}_t \right), \tag{9}$$

Table 1: Performance of Bayesian optimization compared to the exact solution for the stochastic LQR problem.

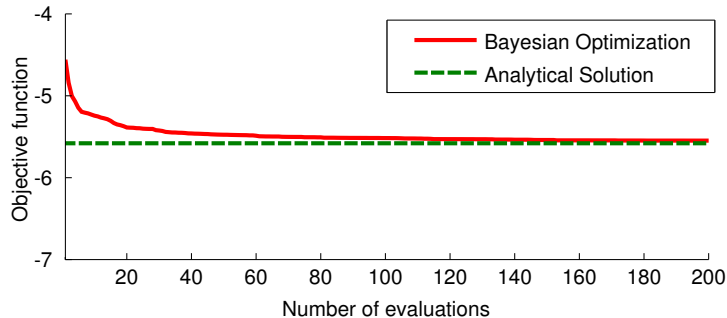|	
Cost incurred by the analytical solution	-5.57 ± 0.01
Cost incurred by Bayesian optimization	-5.54 ± 0.01

Fig. 3: Average over 50 experiments of best parameters find during the minimization process for 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.

where the noise $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$ and the matrices $\boldsymbol{A}_t$, $\boldsymbol{B}_t$, $\boldsymbol{Q}_t \geq \boldsymbol{0}$ and $\boldsymbol{R}_t > \boldsymbol{0}$ are given and, in this paper, assumed to be time invariant. The objective is to find controls $\boldsymbol{u}_0, \ldots, \boldsymbol{u}_{N-1}$ that minimize Equation (9). The control signal $\boldsymbol{u}_t$ is a linear function of the state $\boldsymbol{x}_t$, computed for each time step as

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

where $\boldsymbol{L}_t$ is a gain matrix. An analytical optimal solution to minimize the quadratic cost J exists for the stochastic linear-quadratic regulator [9].

To assess the performance of Bayesian optimization, we consider a stochastic LQR system with $\boldsymbol{x} \in \mathbb{R}^2$, $\boldsymbol{u} \in \mathbb{R}^4$. The stationary gain matrix $\boldsymbol{L} \in \mathbb{R}^{4 \times 2}$ 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 $\boldsymbol{L}$. For Bayesian optimization, we define the objective function as

$$f(\boldsymbol{\theta}) = \log(J/t)\,, \tag{10}$$

where the parameters $\boldsymbol{\theta}$ to optimize are the stationary gain matrix $\boldsymbol{L} \in \mathbb{R}^{4 \times 2}$. To initialize Bayesian optimization, 15 uniformly randomly sampled gain matrices $\boldsymbol{L}$ were used. Moreover, the initial state $\boldsymbol{x}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and the matrices $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{Q}$ and $\boldsymbol{R}$ were fixed.

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 1 shows the mean value for the objective function and its standard deviation for both the analytical solutions
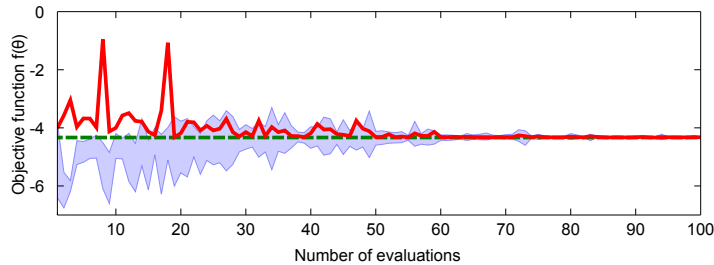
Fig. 4: Example of Bayesian optimization for a stochastic LQR. The objective value function (red curve) and the 95% confidence of the model prediction (blue area) are shown during the optimization process, additionally, the analytical solution (green dashed line) is shown as a reference.

and the ones obtained through Bayesian optimization. We conclude that Bayesian optimization finds near-optimal solutions for the stochastic LQR problem. Additionally, as shown in Figure 3, the average over the 50 experiments of the best parameters find so far in the optimization process suggests that Bayesian optimization reliably quickly finds a near-optimal solution. In Figure 4, an example of the minimization process of Bayesian optimization for the stochastic LQR problem is shown. The objective function is shown as a function of the number of evaluations. Each evaluation requires to compute the objective function f in Equation (10) for the current parameters $\boldsymbol{\theta} = \boldsymbol{L}$. The analytical minimum is shown by the green dashed line, the shaded area shows the 95% confidence bound of the predicted objective function $p(f(\boldsymbol{\theta}))$ for the parameters selected in the ith evaluation. The red line shows the actual measured function value $f(\boldsymbol{\theta})$. Initially, the model was relatively uncertain. With an increasing number of experiments the model became more certain, and the optimization process converged to the optimal solution.

We conclude that Bayesian optimization can efficiently find gain matrices $\boldsymbol{L}$ 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 is not applicable for limited time horizons N, and the discrete time Riccati equation, which can be applied, does not produce a stationary solution.

3.2 Bayesian Optimization for Trajectory Imitation

In the following, we apply Bayesian optimization to learning gaits for bipedal robots based on trajectory imitation. Given a reference trajectory, the objective is to find gait parameters such that the biped's trajectory closely resembles the desired reference trajectory. Gait trajectories are modeled by rhythmic motor primitives. The parameters of the rhythmic motor primitives are typically found by imitation learning [22]. In this paper, we pose this type of trajectory imitation as a Bayesian optimization problem to find the rhythmic motor primitives parameters.
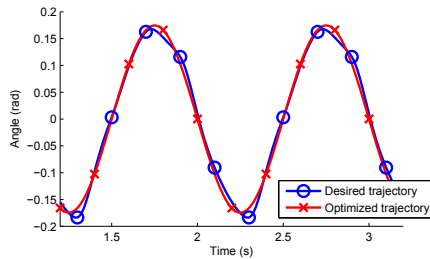
Rhythmic Motor Primitives (RMPs) are parametrizable dynamical systems that model and generate rhythmic trajectories [23]. RMPs have been used to model and learn

bipedal trajectories [22,24] and other rhythmic trajectories, such as drumming [25] and ball paddling [26]. An RMP models a rhythmic trajectory as a modulated limit cycle

$$\tau^2 \ddot{\boldsymbol{q}} = \underbrace{\alpha_z(\beta_z(\boldsymbol{g} - \boldsymbol{q}) - \tau\dot{\boldsymbol{q}})}_{\text{Attractor function}} + \underbrace{\boldsymbol{\theta}\boldsymbol{\psi}r}_{\text{Forcing function}} \,, \qquad (11)$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ can be the joint angles of a robot and their first and second-order derivatives. The attractor function is a limit cycle with timing constants α_z and β_z. The time period of the rhythmic action is τ and can be extracted by frequency analysis of the demonstrations. The amplitude signal $\boldsymbol{r}$ is used to modulate or scale the amplitude of the learned trajectory. The parameter $\boldsymbol{g}$ is the baseline of the rhythmic trajectory.

The *forcing function* modulates the *attractor function* to generate the desired trajectory. The forcing function consists of weight vectors $\boldsymbol{\theta}$ and nonlinear basis function ψ. To model a trajectory using RMPs, we optimize the weight vectors that modulate the attractor function, such that the RMP generates the desired reference trajectory.

Fig. 5: Gait imitation using Bayesian optimization. Example of desired trajectory τ including random noise (blue circle curve) compared with the trajectory generated by the RMP with optimized parameters (red crosses curve). The two curves are almost identical.

The biped used in simulation is an under-actuated three link biped (two links for limbs and one for torso) with five degree of freedom, two of which are actuated. The dynamics are given in [27]. The demonstrated trajectories τ for the limbs were assumed sinusoidal between $+10\,^\circ$ to $-10\,^\circ$, such that at each time instant they were equal in magnitude but opposite in sign. The torso's desired trajectory was assumed constant, bending forward at $+30\,^\circ$. We used RMPs with 5 basis functions to model each of the trajectories. In this set-up, we optimized only the RMP weight vectors $\boldsymbol{\theta}$ in Equation (11). For optimizing the RMP weights we defined the objective

$$f(\boldsymbol{\theta}) = \exp\left(\|\boldsymbol{\tau} - \text{RMP}(\boldsymbol{\theta})\|^2\right) \,, \qquad (12)$$

which penalizes the distance between the trajectory generated by the model $\text{RMP}(\boldsymbol{\theta})$ and a noisy demonstrated trajectory $\boldsymbol{\tau}$. Equation (12) was evaluated using 10 cycles of the trajectory. Bayesian optimization converged after about 50 evaluations. The resulting trajectory, generated by the optimized RMP parameters, closely resembled the desired reference trajectory as shown in Figure 5. Using the generated parameters the biped walked smoothly.

While other approaches (such as least squares and locally weighted regression) exist to solve trajectory imitation for RMPs, the result presented suggests that also Bayesian optimization is suitable for trajectory imitation. Given a trajectory, using Bayesian optimization we can learn the parameters of an RMP to replicate it.
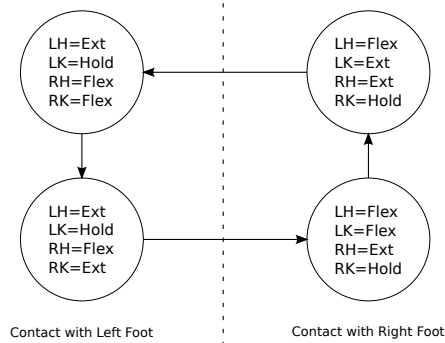
Fig. 6: *Fox* controller is a finite state machine with four states. Each of the four joints, left hip (LH), left knee (LK), right hip (RH) and right knee (RK), can perform one of three actions: flexion (Flex), extension (Ext) or holding (Hold). When a joint reaches the maximum extension or flexion, 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 $\boldsymbol{\theta}$.
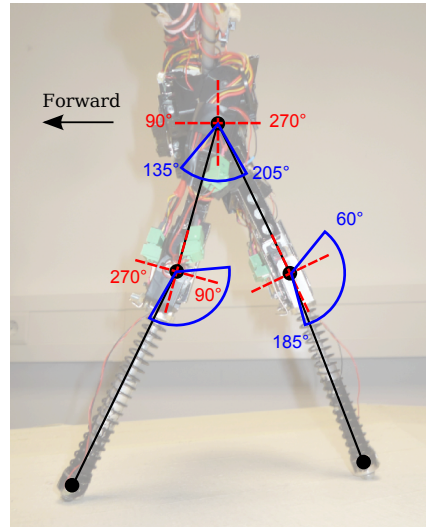

Fig. 7: Hip and knee angle reference frames (red dashed) and rotation bounds (blue solid). The hip joint angles' range lies between $135°$ forward and $205°$ backward. The knee angles range from $185°$ when fully extended to $60°$ when flexed backward.

3.3 Gait Optimization for a Bio-Inspired Biped

In the following set-up, we consider the case where a reference trajectory is no longer available. Instead, gait parameters for a biped are learned directly to maximize walking speed and robustness. In this section, we introduce the hardware of the bipedal robot *Fox*, see Figure 1, used to evaluate Bayesian gait optimization. Moreover, we present experimental results of the gait optimization and analyze the quality of the learned gaits.

Hardware and Controller Description To validate our Bayesian gait optimization approach we used the 2-D dynamic walker *Fox*, shown in Figure 1. 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 [28]. *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 in Figure 7. The touch sensors return 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)*, shown in Figure 6, with four states: two for the swing phases of each leg [29]. 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 thresholds based on the angles of the joints.

For the optimization process, we identified eight parameters of the controller which are crucial for the resulting gait. These gait parameters consist of four thresholds 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 some motors could be ineffective with a different set of motors (e.g., if one or more motors are replaced), due to slightly different mechanical properties. Therefore, gait optimization techniques are essential for this robot.

Gait Optimization Results We applied Bayesian optimization to find suitable parameters for a walking gait of *Fox*. The objective function f to be minimized in the Bayesian optimization was

$$f(\boldsymbol{\theta}) = -\frac{1}{N} \sum_{i=1}^{N} V_i(\boldsymbol{\theta}) \,, \tag{13}$$

i.e., the negative mean walking velocities V_i over $N = 3$ experiments with the robot for a given set of gait parameters $\boldsymbol{\theta}$. Minimizing the performance criterion in Equation (13) maximizes the walking distance in the given time horizon. Moreover, this criterion does not only guarantee a fast walking gait but also reliability, since the gait must be robust to noise and the initial configurations across multiple experiments. Each experiment was initialized from similar initial configurations and consisted of the first 12 seconds starting from the moment when the foot of the robot touched the ground. To initialize Bayesian optimization, three uniformly randomly sampled parameter sets were used.

In Figure 8, 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 in the first step. Large parts of the first 60 experiments were spent to learn that the control signals applied on 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 were typically capable of walking but were rather unstable and fell after few steps. After 80 experiments, the model became more accurate (the function evaluations shown in red lied within the 95% confidence bound of the prediction), and Bayesian optimization found a stable walking gait. The resulting gait[7] was evaluated for a longer period of time, and it proved sufficiently robust to walk continuously for 2 minutes without falling, while achieving a mean velocity of 0.45 m/s. This mean velocity was close to the maximum velocity this hardware set-up can achieve [28]. Notably, the parameters obtained trough Bayesian optimization that correspond to the values of the thresholds were slightly asymmetrical for the two legs. The superior performances of asymmetri-

---

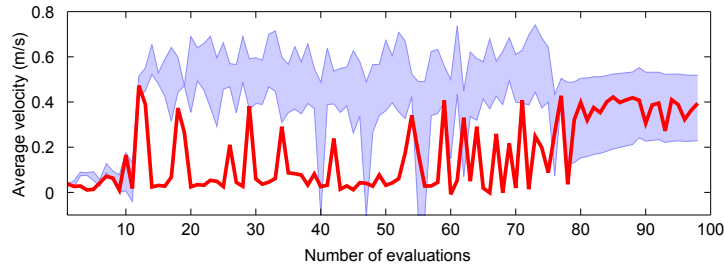[7]Videos are available at `http://www.ias.tu-darmstadt.de/Research/Fox`.

Fig. 8: Average walking speed during the gait optimization process of *Fox* using Bayesian optimization. The objective value function (red curve) and the 95% confidence of the model prediction (blue area) are shown during the optimization process. Three evaluations are used to initialize Bayesian optimization and are not shown in the plot. After 80 evaluations, Bayesian optimization finds an optimum corresponding to a stable walking gait with an average speed of 0.45 m/s.

cal parameters, was already observed during previous hand-tuning of gait parameters and probably originated from the smaller radius of the walking circle for the inner leg.

From our experience with the biped *Fox*, hand-tuning the gait parameters can be a very time-consuming process. Using a (uniform) grid search is infeasible as the number of required experiments would be N^8 where 8 is the number of free parameters that we consider and N is the resolution along each parameter dimension. In the most basic case, when we evaluate each parameters only two points, the final number of evaluations would be $2^8 = 256$, which is already twice the number of evaluations Bayesian optimization needed. Additionally, only a small part of the parameter space leads to walking gaits and the influence of the parameters is not trivial. Hence, more than two points for each free parameter would be required. Expert manual parameter search typically yielded inferior gaits compared to the ones obtained by Bayesian optimization, in both walking velocity and robustness. Additionally, Bayesian optimization sped up the parameter search from days to hours.

4   Conclusion

Gait optimization for bipedal locomotion is a time-consuming and complex task. Manual gait optimization is an empirical process, which requires extensive experience and knowledge. Automatic optimization methods circumvent the need for expert knowledge, but might instead require a larger number of robot interactions. In a context such as bipedal locomotion, where interacting with the robot can be expensive in terms of wearing out or time, these automatic methods become impractical. In this paper, we proposed to use Bayesian optimization to alleviate both these issues by automatically optimizing gaits in only a small number of interactions with the robot.

As a proof of concept, we have shown that Bayesian optimization applied to a stochastic LQR problem can find near-optimal stationary solutions. Moreover, we have

demonstrated that Bayesian optimization can be successfully applied for trajectory imitation. Given a desired reference trajectory, Bayesian optimization found parameters for rhythmic motor primitives that accurately reproduced it. Finally, we applied Bayesian optimization to gait optimization for a real bio-inspired dynamic bipedal walker. Even in the presence of severe noise, our approach automatically found good gaits in a small number of experiments with the bipedal robot. The resulting performance was superior to manually designed gaits. From a practical perspective, Bayesian optimization allowed us to find good gait parameters in hours, whereas manual parameter search required days.

In practice, Bayesian optimization has some limitations. First, Bayesian optimization is currently limited to optimizing 10–20 parameters. The reason for this limitation is that model building with high-dimensional parameters spaces but only sparse data is very challenging. Second, the goodness of the optimization strongly depends on the quality of the learned model. In future, we will explore Bayesian optimization for higher-dimensional problems and studies of multiple acquisition functions and improvements of the expressiveness of the GP model. Moreover, we will develop a continuation of efficient bipedal gait design, such as the evaluation of various gait performance criteria (especially robustness) and comparisons of learned gaits with human gaits.

References

1. Tedrake, R., Zhang, T.W., Seung, H.S.: Learning to walk in 20 minutes. In: Proceedings of the Fourteenth Yale Workshop on Adaptive and Learning Systems,. (2005)
2. 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
3. Kushner, H.: 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
4. Jones, D.R.: A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization **21**(4) (2001) 345–383
5. Osborne, M.A., Garnett, R., Roberts, S.J.: Gaussian processes for global optimization. In: 3rd International Conference on Learning and Intelligent Optimization (LION3). (2009) 1–15
6. Garnett, R., Osborne, M.A., Roberts, S.J.: Bayesian optimization for sensor set selection. In: Information Processing in Sensor Networks, ACM (2010) 209–219
7. Lizotte, D., Wang, T., Bowling, M., Schuurmans, D.: Automatic gait optimization with Gaussian process regression. In: Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI). (2007) 944–949

8. Tesch, M., Schneider, J., Choset, H.: Using response surfaces and expected improvement to optimize snake robot gait parameters. In: Intelligent Robots and Systems (IROS), IEEE (2011) 1069–1074
9. Bertsekas, D.P.: Dynamic Programming and Optimal Control. 3rd edn. Athena Scientific (2007)
10. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing **16**(5) (1995) 1190–1208
11. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. Journal of Machine Learning Research (JMLR) **13** (2012) 281–305
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. 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)
14. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press (2006)
15. Mockus, J., Tiesis, V., Zilinskas, A.: The application of Bayesian methods for seeking the extremum. Towards Global Optimization **2** (1978) 117–129
16. Cox, D.D., John, S.: Sdo: A statistical method for global optimization. Multidisciplinary Design Optimization: State of the Art (1997) 315–329
17. Hennig, P., Schuler, C.J.: Entropy search for information-efficient global optimization. Journal of Machine Learning Research **13** (2012) 1809–1837
18. Auer, P.: Using confidence bounds for exploitation-exploration trade-offs. Journal of Machine Learning Research (JMLR) **3** (2003) 397–422
19. Srinivas, N., Krause, A., Kakade, S., Seeger, M.: Gaussian process optimization in the bandit setting: No regret and experimental design. In: Proceedings of International Conference on Machine Learning (ICML). (2010) 1015–1022
20. Calandra, R., Seyfarth, A., Peters, J., Deisenroth, M.P.: An experimental comparison of Bayesian optimization for bipedal locomotion. In: Proceedings of 2014 IEEE International Conference on Robotics and Automation (ICRA). (2014)
21. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. Journal of Optimization Theory and Applications **79**(1) (1993) 157–181
22. Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., Kawato, M.: Learning from demonstration and adaptation of biped locomotion. Robotics and Autonomous Systems **47**(2) (2004) 79–91
23. Ijspeert, A., Nakanishi, J., Schaal, S.: Learning attractor landscapes for learning motor primitives. In: Advances in Neural Information Processing Systems (NIPS). (2003)
24. Gopalan, N., Deisenroth, M.P., Peters, J.: Feedback error learning for rhythmic motor primitives. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). (2013)
25. Pongas, D., Billard, A., Schaal, S.: Rapid synchronization and accurate phase-locking of rhythmic motor primitives. In: Intelligent Robots and Systems (IROS), IEEE (2005) 2911–2916
26. Kober, J., Peters, J.: Learning motor primitives for robotics. In: International Conference on Robotics and Automation (ICRA). (2009)
27. Grizzle, J., Abba, G., Plestan, F.: Asymptotically stable walking for biped robots: Analysis via systems with impulse effects. IEEE Transactions on Automatic Control (2001)
28. Renjewski, D.: An engineering contribution to human gait biomechanics. PhD thesis, TU Ilmenau (2012)
29. Renjewski, D., Seyfarth, A.: Robots in human biomechanicsa study on ankle push-off in walking. In: Bioinspiration & Biomimetics. (2012)