Bayesian Optimization for Whole-Body Control of High Degrees of Freedom Robots through Reduction of Dimensionality

Citation for published version:
Yuan, K, Chatzinikolaidis, I & Li, Z 2019, 'Bayesian Optimization for Whole-Body Control of High Degrees of Freedom Robots through Reduction of Dimensionality' IEEE Robotics and Automation Letters, vol. 4, no. 3. DOI: 10.1109/LRA.2019.2901308

Digital Object Identifier (DOI):
10.1109/LRA.2019.2901308

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE Robotics and Automation Letters

Bayesian Optimization for Whole-Body Control of High Degrees of Freedom Robots through Reduction of Dimensionality

Kai Yuan, Iordanis Chatzinikolaidis, and Zhibin Li

***Abstract*—This paper aims to achieve automatic tuning of optimal parameters for whole-body control algorithms to achieve the best performance of high-DoF robots. Typically the control parameters at a scale up-to hundreds are often hand-tuned yielding sub-optimal performance. Bayesian Optimization (BO) can be an option to automatically find optimal parameters. However, for high dimensional problems, BO is often infeasible in realistic settings as we studied in this paper. Moreover, the data is too little to perform dimensionality reduction techniques such as Principal Component Analysis or Partial Least Square. We hereby propose an Alternating Bayesian Optimization (ABO) algorithm that iteratively learns the parameters of sub-spaces from the whole high-dimensional parametric space through interactive trials, resulting in sample efficiency and fast convergence. Furthermore, for the balancing and locomotion control of humanoids, we developed techniques of dimensionality reduction combined with the proposed ABO approach that demonstrated optimal parameters for robust whole-body control.**

***Index Terms*—Optimization and Optimal Control, Legged Robots, Humanoid and Bipedal Locomotion, Humanoid Robots**

I. Introduction

For robot locomotion, control parameters are essential for the stabilization [1], Inverse Dynamics and whole-body control [2] of legged robots, e.g. humanoids [3], [4] and quadrupeds [5]. However, the high-dimensional and often sensitive parameters need to be correctly chosen to guarantee stability and good performance, which could be manually tuned or automatically found by search algorithms. The former is time consuming and suboptimal due to the correlation of high-dimensional parameters, while the latter requires sophisticated search subject to the high-dimensionality of the problem that may be sample-inefficient or often impossible.

The influence of parameters on the performance of a task cannot be directly computed, because the evaluation needs to be quantified from the interaction between the robot and the environment. Therefore, the objective function for evaluating the performance can be treated as a black-box, and derivative-free searching algorithms can be useful to determine the optimal parameters. With increasing dimensionality of the parameters, random or grid search approaches are inefficient as the parametric space increases exponentially, and thus the amount of evaluations. Derivative free searching algorithms, such as Sequential Model-based Algorithm Configuration [6], evolutionary algorithm [7], and particle swarm methods [8], are able to find a suitable set of parameters. However, they are not well suited for expensive evaluations due to their sample-inefficient nature.

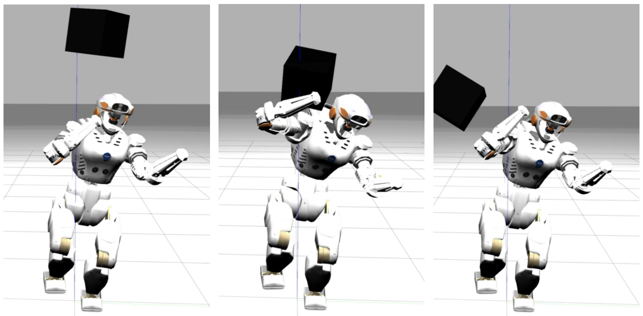


Fig. 1: Robust balancing against perturbations of NASA's Valkyrie using 36 *automatically* tuned control parameters.

Furthermore, automatic tuning of control parameters can be equivalently achieved by machine learning. Reinforcement learning can be used in robot control to tune optimal gait parameters [9], or to directly learn control policies for humanoid balancing control [10]. Alternatively, mathematical optimization can be effective in low dimensional problems such as optimizing the gait parameters considering the kinematics and dynamics constraints [11], [4].

Bayesian Optimization (BO) [12], a derivative-free, sample-efficient optimization algorithm, is suitable to find global optima for black-box optimization functions [13], and is widely used for hyper-parameter tuning [14]. In robotics, BO has been used to find gait parameters on real, physical hardware in [15] (5 dim./9 dim.), [16] (7 dim.), [17] (8 dim.), [18] (15 dim.). Additionally, domain knowledge has been applied to find suitable kernels for improving sample efficiency on hardware experiments [15], [19]. However, the dimensionality in these works was relatively low, and hence a direct implementation of BO performed well.

Despite its benefits, the capability of BO in finding global optima is limited by the high dimensionality, as the search space grows exponentially. Dimensionality reduction approaches, such as Principal-Component Analysis [20] or Partial Least Squares [21], would fail when the evaluation of the objective function is expensive, or the data is insufficient to find correlations in parameters. An approach to combine BO with LQR was suggested in [22], in which, rather than tuning the full state-feedback controller, only the lower dimensional LQR weights were tuned via BO. For humanoid Whole-Body Control, a Trial-and-Error learning algorithm in [23] dealt with model inaccuracies by learning repulsors to alter reference motions that prevents an unstable configuration of state space.

Recently, high-dimensional BO methods have been proposed: in [24] only a subset is optimized over by randomly dropping out parts of the parameters space; in [25] a local

This research is supported by the EPSRC CDT in Robotics and Autonomous Systems (EP/L016834/1), Future AI and Robotics for Space (EP/R026092/1), and Offshore Robotics for Certification of Assets (EP/R026173/1).

The authors are with the School of Informatics, the University of Edinburgh, UK. Corresponding author's email: kai.yuan@ed.ac.uk

search is performed near a priorly given location. However, the work in [24] does not consider the correlation between the parameters and randomly drops out parameters, while the solution in [25] relies on prior knowledge of the optimum which is mostly unavailable without pre-tuning.

To overcome these limitations, we propose a novel approach that applies domain knowledge for partitioning the parameter space, and is able to find an optimum without assuming the location of the optimum. We adopt the idea of Alternating Optimization [26] and iteratively optimize one partitioned parameter sub-space at a time while keeping the rest of the parameter space fixed.

Our proposed algorithm has similarities with Coordinate Descent approaches [27]. However, instead of a sequential search along the coordinates until convergence, our proposed method searches in sub-hyper-boxes of the parameter space and is independent on the convergence of previous iterations. In combination with BO, a global optimization algorithm, the alternating nature reduces the risks of local minima and allows parallelization, which are the two characteristics that are not provided by Coordinate Descent methods.

In this work, we show in simulation that the proposed algorithm is able to find an optimal hyper-parameter set for 36 parameters of a Whole-Body Quadratic Programming controller. As a further validation, our proposed algorithm is shown to be capable of finding the optima of 24 challenging objective functions from the COCO benchmarks [28]. Our contributions are summarised as follows:

- A novel Alternating Bayesian Optimization (ABO) algorithm that is able to optimize black-box objective functions with high-dimensional parameters.
- An automated parameter tuning framework for Whole-Body Control that can find the high-dimensional optimal parametric set from scratch within a few iterations.
- Evaluation of the versatility of the proposed ABO on the COCO benchmarking platform that shows consistent convergence of finding near global optima for challenging high-dimensional objective functions.

This paper is structured as follows. First, the BO problem for whole-body control is formulated in Section II. Second, our novel ABO algorithm and the automated tuning framework are presented in Section III. The results of the whole-body control and benchmarking on COCO are analyzed in Section IV, and the potential uses and possible limitations of the proposed algorithm are discussed in Section V. Finally, we conclude in Section VI.

II. Bayesian Optimization

To find the maximum of an expensive-to-evaluate objective function, BO performs three steps. First, a cheap-to-evaluate, surrogate objective function is built as a Gaussian Process (GP). Second, an acquisition function maximises over the GP in order to find a local maximum. The point found by maximising the acquisition function is a candidate for the global optimum. Third, this point is sampled on the actual expensive-to-evaluate objective function. Observing a point at the suspected optimal points reduces (eliminates if noiseless) the variance of the GP at that point, and therefore refines the GP. By continuously iterating so, a global optimum of the actual objective function can be found [14]. The pseudo code is given in **Algorithm 1**. The sampled objective value y_i consists of the true objective function $J(\boldsymbol{x}_i)$ and the noise ϵ_i. The notation $\boldsymbol{y}_{1:T}$ indicates the samples gathered for y_i at time step $i = 1, ..., T$.

Algorithm 1 Pseudo code for Bayesian Optimization

```
$\boldsymbol{y}_{1:T} \leftarrow J(\boldsymbol{x}_{1:T}) + \epsilon_{1:T}$,                    sample $T$ points
Initialise GP with $\mathcal{D}_{1:T} \leftarrow \{\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T}\}$
for $i = 1, 2, ..., N$ do
    $\boldsymbol{x}_t \leftarrow \mathrm{argmax}_{\boldsymbol{x}} a(\boldsymbol{x})$,          get next query point
    $y_t \leftarrow J(\boldsymbol{x}_t) + \epsilon_t$,                 sample query point $\boldsymbol{x}_t$
    Update GP with $\mathcal{D}_{1:t} \leftarrow \{\mathcal{D}_{1:t-1}, (\boldsymbol{x}_t, y_t)\}$
```

A. Gaussian Processes

A Gaussian Process is defined by a mean function $m(\boldsymbol{x})$ and a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')). \tag{1}$$

The prior mean $m(\boldsymbol{x})$ (not conditioned on data) is chosen to be a zero function $m(\boldsymbol{x}) = 0$ [29]. For the choice of a covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$, several kernels have been proposed [29]. Throughout this work, we used a Matérn kernel with the parameters[1] $\nu = 1.5$, $l = 1.0$:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} d}{l} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu} d}{l} \right), \tag{2}$$

where $d = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, gamma function $\Gamma(\cdot)$, modified Bessel function K_ν, and non-negative parameters ν, l.

B. Optimization Problem Formulation

The BO aims to find a set of parameters X that maximises the reward function $J(X)$, which keeps the objectives to be within a certain user-defined range. The reward function $J_{track}(X)$ for tracking performance is:

$$J_{track}(X) = r_C + r_F + r_T + r_P + r_H, \tag{3}$$

where the tracking objectives are Centre of Mass (COM) (r_C), foot (r_F), torso (r_T) and pelvis (r_P), and hand (r_H). Every tracking reward $r_X = \sum_{i=0}^{N} r_{x,i}$ consists of the sum of N rewards $r_{x,i}$. A reward $0 < r_{x,i} \leq 1$ at time step i for tracking a desired value x_{d_i} is given if the objective is in a certain range (determined by width κ):

$$r_{x,i} = \exp(-\kappa \|x_{d_i} - x_i\|^2). \tag{4}$$

The width $\kappa = -\ln(C)/\delta_{max}^2$ is calculated by the range δ_{max} and associated reward $C \doteq 0.001$ ($C \to 0$, because $C =$

[1] While BO requires user-choices, e.g., kernel and acquisition function and related hyper-parameters, our ABO in this study is robust towards these choices. The same results were achieved by different acquisition functions (Expected Improvement, Upper Confidence Bound, Probability of Improvement), and kernels (Radial Basis Function, Matérn, Rational Quadratic) using the default parameters of scikit-learn.

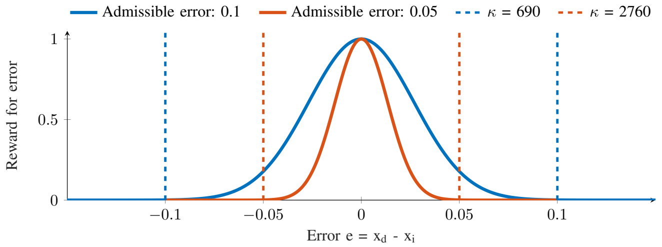


Fig. 2: Reward of error values for $\kappa = 690$ and $\kappa = 2760$.

0 and $\ln(0)$ are infeasible). Critical links, such as feet and COM have an error range $\delta_{max} = 0.05$ for both orientation in $[rad]$ and position in $[m]$ ($\kappa = 2760$, red curve, Fig. 2). An error range $\delta_{max} = 0.1$ for torso, pelvis, and hands yields a width parameter $\kappa = 690$ (blue curve, Fig. 2). The admissible error range is motivated by the physical shape of the support polygon, where $0.1m$ and $0.05m$ correspond to half of the foot length and width respectively, and the rotational errors approximately match the smallest torso joint limits.

A fall penalty is added if either the orientation of the torso θ_{rpy} exceeds a threshold $\boldsymbol{\delta}_{rpy_{max}}$, or the pelvis height z_c is below a threshold δ_z:

$$J_{fall} = \begin{cases} 0, & \text{if } \boldsymbol{\theta}_{rpy} \geq \boldsymbol{\delta}_{rpy_{max}} \text{ or } z_c < \delta_z, \\ 1, & \text{else.} \end{cases} \tag{5}$$

The overall reward function is:

$$J(X) = J_{track}(X) \cdot J_{fall}. \tag{6}$$

C. Acquisition Function

The goal of an acquisition function $a(\boldsymbol{x})$ is to have a fast-to-evaluate function at any given point $\boldsymbol{x}$ in order to decide where to sample next for the observation $(y_i, \boldsymbol{x}_i)$. The next sample point is chosen as the point $\boldsymbol{x} = \text{argmax}_{\boldsymbol{x}} a(\boldsymbol{x})$ that maximises the acquisition function. The mean $\mu(\boldsymbol{x}) = \mu(\boldsymbol{x}; \mathcal{D}, \theta)$ and variance $\sigma^2(\boldsymbol{x}) = \sigma^2(\boldsymbol{x}; \mathcal{D}, \theta)$ are calculated from previous observations $\mathcal{D} = \{\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}\}$ and hyper-parameters θ of the GP. In this paper, the upper confidence bound (UCB) [30] is used. It automatically trades off exploration and exploitation by weighing mean against variance:

$$a_{UCB}(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \alpha\sigma(\boldsymbol{x}), \tag{7}$$

where the trade-off parameter $\alpha \geq 0$. The UCB for maximisation can be seen as an algorithm that minimises the accumulated regret $R_T = \Sigma_{t=1}^{T} f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)$ with the unknown-optimal point $\boldsymbol{x}^*$ to a point of no-regret [31]:

$$\lim_{T \to \infty} R_T / T = 0. \tag{8}$$

III. Dimensionality Reduction Techniques

This section presents the principles for reducing the dimensionality of the optimization variables. First, we elaborate our proposed Alternating Bayesian Optimization (ABO) approach for finding high-dimensional parameters using a pseudo algorithm. Next, we introduce domain knowledge such as symmetry to reduce the dimensionality of parameters from more than 100 down to 36, and then group these 36 parameters by their correlation.

A. Core Concept and Formulation of the Algorithm

We propose a novel method (**Algorithm 2**) of finding the global optimum for the whole set of optimization variables $\boldsymbol{X}$. We partition $\boldsymbol{X} = [X_1, ..., X_K]$ into K groups, and successively optimize the objective function (6) on each partition X_j $(j = 1, ..., K)$ while keeping the other $K - 1$ partitions fixed. We define the crossed-out notation $\cancel{X_i}$ to indicate that the parameter group X_j $(j = 1, ..., K)$ is fixed.

First, an initial set of parameters $\boldsymbol{X}_0$ will be used for all optimization variables with maximum value $y_{max} = J(\boldsymbol{X}_0)$. Next the objective function $J(\boldsymbol{X})$ will be optimized via BO with all parameter groups being fixed except X_j. If using X_j results in a better value than the current maximum value y_{max}, then X_j will be used and y_{max} will be updated.

After iterating through all K groups for N times or when a termination criterion is met, a final BO step with N_{final} iterations is performed to locally optimize around the near optimal parameter set. This step aims to either fine-tune the parameter $\boldsymbol{X}_{sub}$ (if $J(\boldsymbol{X}_{final}) - J(\boldsymbol{X}_{sub}) < \delta$), or unstuck the optimization (if $J(\boldsymbol{X}_{final}) - J(\boldsymbol{X}_{sub}) > \delta$).

In **Algorithm 2**, ABO is used to find near-optimal parameters, and the final holistic BO will either globally fine-tune or unstuck the local optimum. For large variations between ABO (**Algorithm 2**, line 2) and holistic BO (**Algorithm 2**, line 9), which indicates a local maximum, the algorithm will be restarted with new initialisation $\boldsymbol{X}_0 = \boldsymbol{X}_{final}$.

Algorithm 2 Pseudo code for Alternating Bayesian Optimization

```
 1: $\boldsymbol{X} \leftarrow \boldsymbol{X}_0$, $y_{max} \leftarrow J(\boldsymbol{X}_0)$
 2: for $i = 1, 2, ..., N$ and not terminate do
 3:     for $j = 1, ..., K$ do
 4:         $X_j^+ \leftarrow \underset{X_j}{\text{argmax}} J(\cancel{X_1}, ..., X_j, ..., \cancel{X_K})$
 5:         if $J(X_1, ..., X_j^+, ..., X_K) > y_{max}$ then
 6:             $\boldsymbol{X} \leftarrow [X_1, ..., X_j^+, ...X_K]$
 7:             $y_{max} \leftarrow J([X_1, ..., X_j^+, ..., X_K])$
 8: $\boldsymbol{X}_{sub} \leftarrow \boldsymbol{X}$
 9: for $i = 1, 2, ..., N_{final}$ do
10:     $\boldsymbol{X}_{final} \leftarrow \text{argmax}_{\boldsymbol{X}} J(\boldsymbol{X})$
11: if $J(\boldsymbol{X}_{final}) - J(\boldsymbol{X}_{sub}) > \delta$ then
12:     Restart ABO with $\boldsymbol{X}_0 \leftarrow \boldsymbol{X}_{final}$
```

B. Reduction of Dimensionality for Whole-Body Control

The Quadratic Programming (QP) based Whole-Body controller optimizes physically feasible torques for tracking task-space references. The task priorities are represented by the weights as $w = [w_0, ..., w_n]$ in the objective function of the whole-body optimization problem[2] (9). The whole-body QP in [3] is adopted and the tasks J_{tasks} are:

$$J_{task} = \frac{1}{2}\|AX - b\|^2, \tag{9}$$

[2] The priority is determined via ABO. The reward function (6) is designed such that the weights will be optimized to keep the tracking error within the user-specified error range and as close to zero as possible.

where $A = [w_0 A_0, \cdots, w_n A_n]^T$, $b = [w_0 b_0, \cdots, w_n b_n]^T$, and the optimization variable $X = [\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{\lambda}]^T$ consisting of torque commands $\boldsymbol{\tau}$, joint accelerations $\ddot{\boldsymbol{q}}$ and ground reaction forces $\boldsymbol{\lambda}$. Rearranging (9) leads to the QP form:

$$\min_X X^T H X + f^T X \tag{10}$$

$$s.t. \quad A_{eq} X + B_{eq} = 0 \tag{11}$$

$$A_{ineq} X + B_{ineq} \geq 0. \tag{12}$$

The cost function (10) is a weighted sum over tracking objectives and regularization (cf. Table I). The Cartesian reference acceleration for tracking COM and body link trajectories is calculated from the desired position x_d, velocity $\dot{x}_d$, and acceleration $\ddot{x}_d$ via a PD law as:

$$\ddot{x} = \ddot{x}_d + K_P(x_d - x) + K_D(\dot{x}_d - \dot{x}). \tag{13}$$

The equations of motion form the equality constraints (11):

$$\begin{bmatrix} M(q) & -S & -J^T(q) \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\tau} \\ \boldsymbol{\lambda} \end{bmatrix} + h(q, \dot{q}) = 0, \tag{14}$$

with inertia matrix $M(q)$, selection matrix S, stacked Jacobian matrices $J^T(q)$ of the contact links, and nonlinear effects $h(q, \dot{q})$. Torque limits, friction constraints, and COP constraints are considered in the inequality constraints (12) and are implemented as proposed in [32].

C. Optimization Variables

By assuming symmetry between the left and right and grouping symmetric links, the optimization variables can be reduced to 36 (Table I). As shown in Table II, III, holistically optimizing 36 parameters altogether for the whole-body control yields suboptimal results, and is thus impractical.

We will show that grouping the optimization variables into three physically meaningful groups will result in faster convergence and better performance. The parameters $X = [X_W, X_{PD}, X_M]$ are categorised into objective function weights X_W (9), PD gains X_{PD} (13), and miscellaneous parameters X_M. Here, the miscellaneous set X_M describes the parameters required for the wrench constraints as in [32], including the foot geometry x_{front}, x_{back}, y_{left}, y_{right} with y_{side} = y_{left} =y_{right}, and friction constraints μ. Instead of using a constant foot size in X_M, the foot geometry can be treated as a tunable parameter for finding a suitable stability margin, where boundary limits are the actual foot dimensions of the robot. We found that leaving a stability margin for the actual foot geometry and X_M leads to higher robustness. Averaged over 10 trials, a stability margin of $2cm$ was found.

IV. Results

This section presents three key results: 1) the ability of ABO to find optimal parameters for whole-body control of the Valkyrie robot; 2) comparison of ABO with three other parameter search algorithms; 3) a further evaluation study of ABO on the COCO benchmarking suite [28].

A. Comparison Methodology

ABO is compared with three other parameter search algorithms: holistic BO, alternating random search, and BO using dropout [24]. The holistic BO approach optimizes over the whole parametric space (**Algorithm 1**), while ABO iteratively optimizes over its sub-spaces (**Algorithm 2**). The alternating random search method is similar to our ABO approach, but the next sample point is uniformly and randomly sampled from the search space instead of being found by an optimized acquisition function.

BO using dropout achieves dimensionality reduction by randomly dropping out dimensions and optimizing over d parameters instead of the full, high-dimensional parameter space. The work in [24] proposed two fill-in strategies for the dropped dimensions: dropout-random and dropout-copy. The third method, dropout-mix, performed similar to dropout-copy and is not shown in our comparison (Fig. 5b) for clarity purposes. We implemented both fill-in strategies with a sampled dimension of $d = 8$ and $d = 16$.

For ABO, every BO iteration consists of 30 BO evaluations with UCB as acquisition function $a_{UCB}(\boldsymbol{x})$ using the trade-off parameter $\alpha = 3$. For the low-dimensional parameters X_M, only 10 BO evaluations are conducted. Thus, for the whole-body control, 70 objective function evaluations are conducted per iteration (30 for X_W, X_{PD}, 10 for X_M). The holistic, random search, and dropout algorithms use the same evaluation budget (70 per iteration) as ABO.

B. Optimizing Hyper-parameters for Whole-Body Control

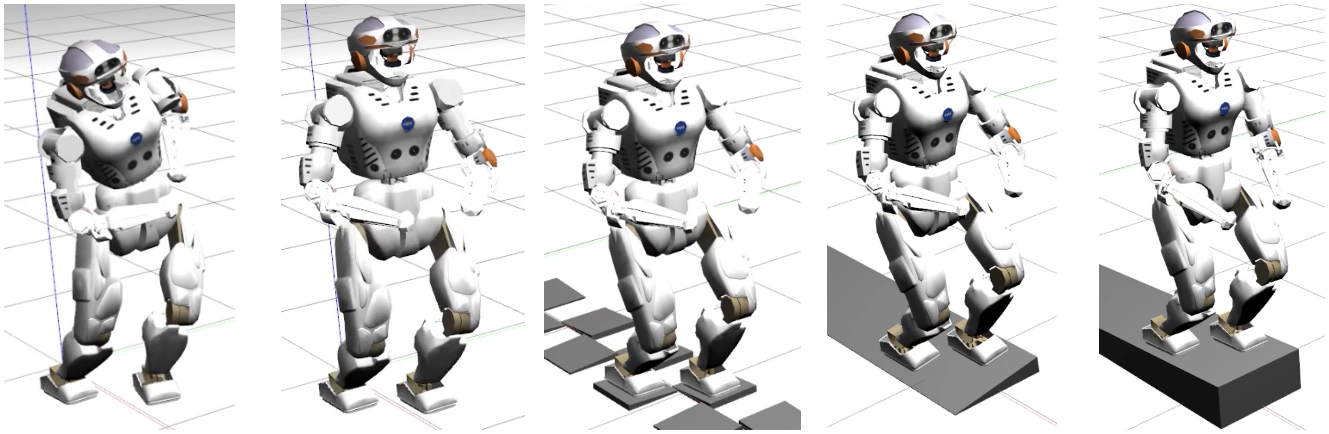
This section presents the results obtained from learning optimal parameters by interacting with the environment. All simulations were conducted in Gazebo using an accurate model provided by NASA for the humanoid Valkyrie [33] - a $1.80m$ tall, $139kg$ heavy humanoid robot with 44 actuated Degrees of Freedom (DOF).

An episode consists of a start and an end phase of $1s$ each and 5 straight steps of $0.25m$ with a step duration of $3s$. The walking motion is generated by using Model-Predictive Control as in [34] to track these pre-planned footsteps for both Gait Planning and Feedback Control. At a frequency of $50Hz$, the sum of 850 rewards (4) are gathered per episode as the scalar output of the reward function. To the best of our skills, the hand-tuned parameter set yielded a value of 726. At a theoretical maximum of 850 for absolutely perfect tracking, a value of over 700 is able to robustly perform all locomotion tasks (Fig. 3b). As a baseline, walking all 5 steps using bad tracking is possible for values over 550, and standing is possible for values over 250 (Fig. 3a). The objective weight and PD gain parameters for the alternating approaches are initialised (alg. 2, line 1) with zeros $X_W = X_{PD} = \mathbf{0}$, and the miscellaneous parameters are initialised using their mean values $X_M = [0.09, 0.04, 0.03, 0.5]$.

On average, optimal parameters were found within 6 iterations via ABO (alg. 2). These parameters were generalised to multiple locomotion and balancing tasks and exhibit good impedance behaviour of walking over unmodelled, uneven

TABLE I: Range for the optimization variables. For pose tracking, separate weights for position and orientation are used.

Objective weight	Objective	Range x in $[10^x]$	PD gain	Controlled link	Range x in $[10^x]$	Misc. param.	Range
O0	Track COM	$[0]$	C1-C2	PD COM position	$[-1,3]$	x_{front}	$[0, 0.18]$
O1-O2	Track torso pose	$[-3,3]$	C3-C4	PD COM orientation	$[-1,3]$	x_{back}	$[0, 0.08]$
O3-O4	Track pelvis pose	$[-3,3]$	C5-C6	PD foot position	$[-1,3]$	y_{side}	$[0, 0.06]$
O5-O6	Track Hand pose	$[-3,3]$	C7-C8	PD foot orientation	$[-1,3]$	μ	$[0.2, 0.8]$
O7-O8	Track Foot pose	$[-3,3]$	C9-10	PD torso/pelvis position	$[-1,3]$		
O9-O10	Track COP & weight dist.	$[-3,3]$	C11-C12	PD torso/pelvis orientation	$[-1,3]$		
O11	Joint angle $\boldsymbol{q}$	$[-3,3]$	C13-C14	PD hand position	$[-1,3]$		
O12-O16	Reg. $\boldsymbol{\tau}, \ddot{\boldsymbol{q}}, \dot{\boldsymbol{\tau}}, \dot{\boldsymbol{\lambda}}$	$[-3,3]$	C15-C16	PD hand orientation	$[-1,3]$		



(a) Objective value of 250 (poorly tuned, 1st iteration). (b) Objective value of 730 (well tuned, 6th iteration). (c) Gait on slabs with 5° inclination (well tuned). (d) Gait on roll inclination of 10° (well tuned). (e) Gait on pitch inclination of 5° (well tuned).

Fig. 3: Snapshots of different walking trails using automatically tuned parameters. The detailed motions and scenarios can be found in the accompanying video.

terrain (cf. supporting video, Fig. 1, Fig. 3c-e). The tracking performance of task space references is shown in Fig. 4.

The learning curves of the automatic parameter tuning are in Figure 5. The best values indicate the maximum value at the respective time step and therefore only shows improving values. The mean of this curve shows that the robot starts to walk after roughly 2 ABO iterations (140 evaluations), achieves a manual-tuned level after 4 ABO iterations (280 evaluations), and performs the final mean value of 732 after 8 ABO iterations (560 evaluations).

Table II shows the number of iterations and the final value at convergence from four different methods. Over 10 trials, ABO always found a parameter set for task completion. The alternating random approach has one trial finding such a parameter set, but the holistic approach has no trial of finding a working parameter set. BO using dropout-copy was able to find a task-completing parameter set for all 10 trials with both $d = 8, d = 16$, while BO using dropout-random was not able to find a working parameter set. The results in Table II suggest that the nature of alternating search of parameters for high-dimensional problems greatly improves the success of finding good parameters even for non-sophisticated search algorithms such as random search.

Both ABO and dropout-copy for $d = 8$ and $d = 16$ can achieve an objective value over 550 indicating that the robot is able to walk stably (Figure 5b). However, ABO converges faster and achieves a higher value than the dropout-copy, and thus tracks the reference trajectories better. Dropout-random achieves similar performance as the alternating random search, which is not able to succeed a stable 5-step walking task.

C. Validating Versatility of ABO by COCO Benchmarks

To further understand the versatility of solving other problems, ABO was validated on the COCO benchmarking suite that has 24 functions (f1-f24) in an explicit form for benchmarking global optimizers in a black-box setting. Notably, in addition to standard objective functions (f1, f5, f13), it also contains ill-conditioned (f2, f6, f10, f11, f12, f18), local minimum trapping (f3, f4, f7, f8, f9, f14, f15, f20, f23), irregular (f16, f21, f22), and multi-modal (f17, f19, f24) objective functions to thoroughly test the optimizer.

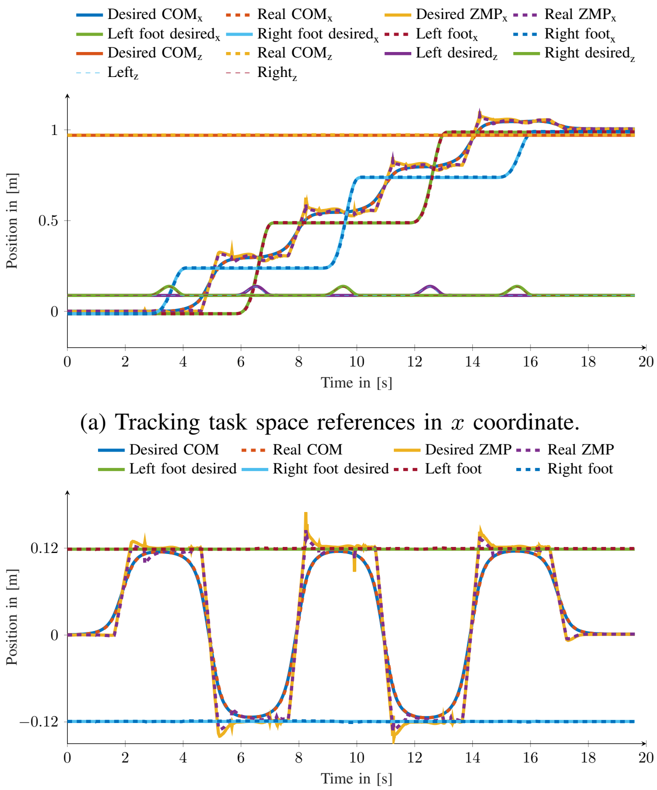
For normalization, every objective function is off-setted such that the smallest value is larger than zero. The smallest possible value varies for every objective function $f1 - f24$ and increases with the dimension of the problem. In addition to a maximal number of iterations, tolerances $t_1 = 1\%$ and $t_2 = 10\%$, representing the tolerance of being 1% and 10% away from the maximal possible value, are used as termination criterion calculated individually for each function $f1 - f24$. The average maxima for the well and poorly conditioned objective functions, and the success rate r_{t_1}, r_{t_2} of using tolerance rate t_1 and t_2 are described respectively in Table III. All parameters are uniformly randomly initialised, and the optimization variables are partitioned randomly.

TABLE II: Average number of iterations and final values over 10 trials for Whole-Body Control of NASA's Valkyrie. The values in parenthesis indicate the number of evaluation.

Prob. dim.	Opt. dim.	Mean iter.	Mean max.	Median iter.	Median max.	Best iter.	Best max.	Worst iter.	Worst max.	Method
36	36	8 (560)	304	8 (560)	299	8 (560)	360	8 (560)	289	Holistic BO
36	16	**5 (350)**	**734**	**6 (418)**	**737**	**2 (84)**	**779**	**8 (560)**	**665**	Alternating BO
36	16	8 (560)	452	8 (560)	448	8 (560)	662	8 (560)	259	Alternating Rand.
36	16	8 (560)	560	8 (560)	605	8 (560)	676	8 (560)	330	Dropout-copy, $d = 16$
36	16	8 (560)	403	8 (560)	384	8 (560)	470	8 (560)	330	Dropout-random, $d = 16$
36	8	8 (560)	611	8 (560)	584	8 (560)	742	8 (560)	543	Dropout-copy, $d = 8$
36	8	8 (560)	386	8 (560)	300	8 (560)	440	8 (560)	294	Dropout-random, $d = 8$

TABLE III: Average number of iterations and final values over 10 trials for well and poorly (f2, f7, f10, f11, f12, f18, f22) conditioned optimization problems. The random search is averaged over 1000 trials.

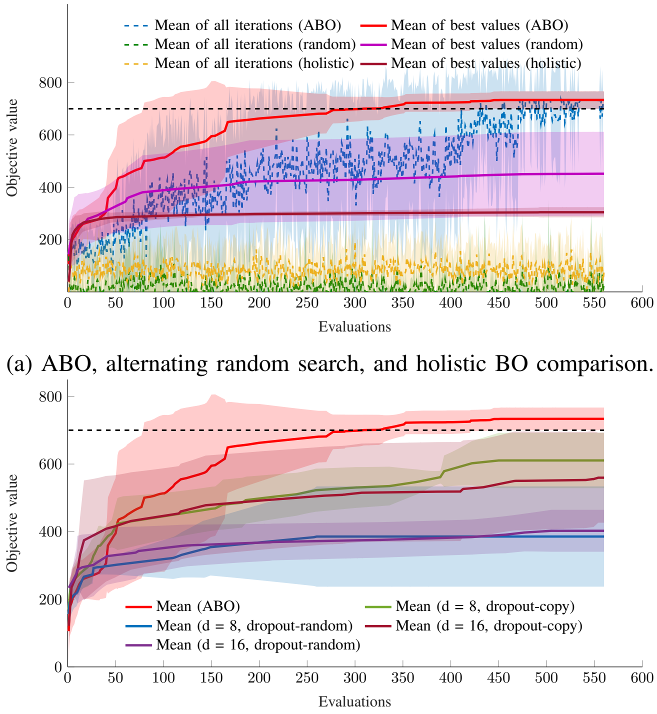
Prob. dim.	Opt. dim.	Max. value (good cond.)	Max. value (bad cond.)	Success rate r_{t_1} (opt.)	Success rate r_{t_2} (opt.)	Success rate r_{t_1} (rand.)	Success rate r_{t_2} (rand.)	Method
40	40	$6 \cdot 10^5$	$7 \cdot 10^{10}$	0.12	0.13	0.00	0.12	Holistic
20	20	$3 \cdot 10^5$	$8 \cdot 10^9$	0.13	0.24	0.09	0.16	Holistic
10	10	$3 \cdot 10^5$	$7 \cdot 10^9$	0.54	0.80	0.15	0.40	Holistic
5	5	$2 \cdot 10^5$	$3 \cdot 10^8$	0.64	0.88	0.26	0.46	Holistic
40	16	$6 \cdot 10^5$	$7 \cdot 10^{10}$	0.13	0.70	0.13	0.32	Alternating
40	10	$6 \cdot 10^5$	$7 \cdot 10^{10}$	**0.21**	**0.75**	0.13	0.38	Alternating
20	10	$3 \cdot 10^5$	$8 \cdot 10^9$	**0.27**	**0.74**	0.12	0.39	Alternating
10	5	$3 \cdot 10^5$	$7 \cdot 10^9$	**0.59**	**0.88**	0.31	0.48	Alternating



(a) Tracking task space references in x coordinate.

(b) Tracking task space references in y coordinate.

Fig. 4: Tracking of COM, COP and feet trajectories, where the yellow solid line is the online re-generated desired ZMP consists of the nominal ZMP and the feedback corrections.



(a) ABO, alternating random search, and holistic BO comparison.

(b) Comparison between ABO, BO using dropout-random ($d = 8, 16$), and BO using dropout-copy ($d = 8, 16$).

Fig. 5: Comparison between ABO, alternating random search, holistic BO, and BO using dropout for Whole-Body Control (shaded area: standard deviations).

The lower success rates from the alternating random search suggest that the higher success rates of ABO is attributed to our proposed method, rather than randomly finding the optimum.Furthermore, similar to the whole-body control ABO case, using an alternating approach increases the success rate over a holistic one, especially for high-dimensional problems. In contrast to [28], an evaluation budget is used. This is due to the fact that a limitless evaluation budget would lead to lengthy computations, as well as an exhaustive search that eventually leads to the optimum. The COCO benchmark, in which budget is considered as secondary, is designed for being challenging to "defeat" the optimization problem. We aim to preserve the sample efficient nature of BO, by trading off an absolute success rate of the ABO approach that can arguably be improved. In fact, the success rate of ABO being 2-3 times higher than that of holistic BO has already shows an advantage. In summary, a holistic BO approach yields good results for dimensions up to 10D but low success rates for the dimensions higher than 10D, whereas the ABO approach yields good results of success rates over 75%.

V. Discussion

This section discusses the potential use of our proposed ABO algorithm and the consideration of deploying ABO on real systems regarding the simulation-to-reality gap.

A. Reality Gap between Simulation and Real World

Model uncertainties hinder a direct transfer of the tuned parameters from simulation to reality, and are mainly caused by the inaccuracy of the masses, inertias, link dimensions, communication latency, and noises. The following study shows the robustness of our automatically tuned parameters against model uncertainties, and suggests the need of a small number of fine-tuning iterations on the real system given an initial set of parameters auto-tuned from a perfect model.

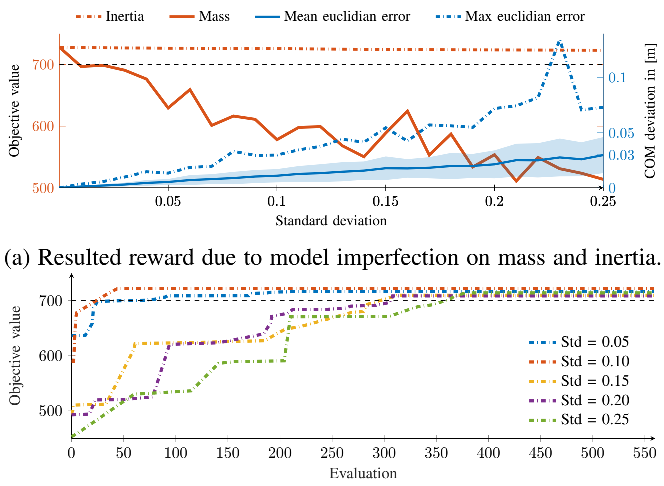
1) Model Imperfections: The imperfection was modelled by the differences between model and real robot in mass and inertia. We first ran ABO on a perfect model in simulation, and then for the test, we altered the simulation model by adding uncertainties of masses and inertia from a normal distribution with varying standard deviations.

In Figure 6a, the results averaged over 5 trials suggest that it is important to accurately identify mass as it contributes greatly to the reward and causes COM deviations (blue line, Fig 6a) of up to 3cm on average. In contrast, wrong inertia identification contributes much less to the reality gap. Inaccuracies of up to 5% can be tolerated without losing tracking quality, and even inaccuracies of up to 25% still yield stable gait despite bad tracking.

2) Fine-tuning of Parameters on Real Hardware: After obtaining a parameter set that works well in simulation, but exhibits bad tracking performance on the real model, ABO has the potential to be further used to fine-tune parameters on the real robot, which is studied in simulated cases here (not on the real hardware) by using a different model with intentional changes mimicking discrepancies between simulation and hardware. From the results in Fig. 6b, it can be seen that for small model errors, ABO converges after one iteration and 50 function evaluations; for a 25% discrepancy as the reality gap, the algorithm requires 6 ABO iterations (360 evaluations) to reach optimal behaviour.

B. Potential and Possible Applications of ABO

In addition to tuning whole-body control, ABO could also identify and tune additional parameters by considering model asymmetries between the left and right, using independent gains in x, y, z coordinates, and identifying dynamics and off-diagonal elements in the weight matrices (10). In the following we depict potential applications of ABO.



(a) Resulted reward due to model imperfection on mass and inertia.

(b) Learning curve for fine-tuning on realistic model.

Fig. 6: Reward from different model discrepancies and the resulted learning curves for fine-tuning the parameters.

1) Consideration of Asymmetries: In Table I symmetry between left and right side of the robot is assumed, which may not be true in reality. In Section V-A, we showed that the optimized parameters were robust to uncertainties including model asymmetry, and additional fine-tuning can further improve the performance. Alternatively, the model asymmetry could be directly included in the optimization setting by adding more optimization variables. As a proof of concept, we were able to achieve the same performance as before by adding 4 additional tuning parameters for the left and right foot separately.

2) Separate Parameters for Postural Control: In addition to physical asymmetries, the body pose can also be decomposed into 6 components with separate weights requiring longer training time due to higher dimensionality. We tested separate horizontal (x, y) and vertical (z) gains for the feet and COM, and obtained similar results as in the original non-separating case by an average increase of training time of 100 evaluations.

3) Adaptation to Variation of Dynamics Properties: In Section IV-B, we exemplarily showed the possibility of identifying friction parameters. Identification of more parameters could be conducted on the dynamics parameters. To this end, ABO would optimize over masses and inertia matrices. We used ABO to successfully identify the 8 heaviest links and the inertia matrix of torso with a deviation of up to 10% from the nominal values specified in the URDF.

4) Identification of Correlated Off-diagonal Elements: Perceiving correlations of the off-diagonals in manual tuning is difficult for humans, and thus ABO could potentially be a leverage to identify those elements in the cost function (10). Future work is needed to study ABO's ability to find suitable parameters given a much larger number of additional optimization variables.

VI. Conclusion

In this work, we proposed an Alternating Bayesian Optimization (ABO) algorithm capable of tuning optimal parameters for high-dimensional optimization problems. This was achieved by evaluating data from interactive trials of between the robot and the environment (simulated scenarios) and the search of appropriate hyper-parameters, which produced optimal performance of robust locomotion under model uncertainties. We tackle the arising problems in high-dimensionality, such as sample-inefficiency and exhaustive searches, by partitioning the whole parameter space into low-dimensional sub-spaces, and iteratively optimizing over each sub-space while fixing the rest sub-spaces.

We first applied dimensionality reduction to reduce 100 more parameters to 36, and then used the proposed ABO algorithm to automatically tune this 36-dimensional parameter set for whole-body control of NASA's Valkyrie robot to locomote over uneven terrains. The robot stood stably after 1 iteration, performed dynamic walking after 3 iterations, and converged to the best performance within 6 iterations.

The proposed method found better parameters within fewer iterations than the manual tuning from the experienced researchers. This is mainly due to human limitation in spotting correlations between parameters in high dimensions, whereas ABO utilises Gaussian Processes which are precisely designed to capture these correlations. Hence, our proposed algorithm can be applied to other systems that require automatic tuning of high-dimensional, correlated hyper-parameters. In the COCO benchmarking suite, the proposed ABO algorithm was further validated by finding global optima of the challenging objective functions. Lastly, we discussed potential applications and limitations of the algorithm that may arise particularly during the transfer from simulation to reality.

Our future work will focus on implementing ABO on real hardware for automatic gait tuning by using the parameters obtained from simulation as an initialisation. Furthermore, constrained Bayesian Optimization methods, such as SafeOpt [35], will be considered to protect the robot from damage.

References

[1] C. Zhou *et al.*, "Stabilization of bipedal walking based on compliance control," *Autonomous Robots*, 2016.

[2] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, 2005.

[3] S. Feng *et al.*, "3D Walking Based on Online Optimization," *International Conference on Humanoid Robots*, 2013.

[4] C. Zhou *et al.*, "Overview of gait synthesis for the humanoid coman," *Journal of Bionic Engineering*, vol. 14, no. 1, pp. 15–25, 2017.

[5] M. Kalakrishnan *et al.*, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, 2011.

[6] F. Hutter *et al.*, "Sequential model-based optimization for general algorithm configuration," in *International Conference on Learning and Intelligent Optimization*, 2011.

[7] Z. Tang *et al.*, "Humanoid walking gait optimization using ga-based neural network," in *Int. Conference on Natural Computation*, 2005.

[8] C. Niehaus *et al.*, "Gait optimization on a humanoid robot using particle swarm optimization," in *Second Workshop on Humanoid Soccer Robots*, 2007.

[9] H. Dallali *et al.*, "On global optimization of walking gaits for the compliant humanoid robot, coman using reinforcement learning," *Cybernetics and Information Technologies*, 2012.

[10] C. Yang *et al.*, "Learning whole-body motor skills for humanoids," in *International Conference on Humanoid Robots (Humanoids)*, 2018.

[11] C. Zhou *et al.*, "A generic optimization-based framework for reactive collision avoidance in bipedal locomotion," in *International Conference on Automation Science and Engineering (CASE)*, 2016.

[12] B. Shahriari *et al.*, "Taking the human out of the loop: A review of bayesian optimization," *Proceedings of the IEEE*, 2016.

[13] D. Jones *et al.*, "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 1998.

[14] J. Snoek *et al.*, "Practical bayesian optimization of machine learning algorithms," in *Adv. in neural information processing systems*, 2012.

[15] A. Rai *et al.*, "Bayesian optimization using domain knowledge on the ATRIAS biped," *Int. Conference on Robotics and Automation*, 2018.

[16] M. Tesch *et al.*, "Using response surfaces and expected improvement to optimize snake robot gait parameters," in *Intelligent Robots and Systems*, 2011.

[17] R. Calandra *et al.*, "Bayesian gait optimization for bipedal locomotion," in *Int. Conf. on Learning and Intelligent Optimization*, 2014.

[18] D. J. Lizotte *et al.*, "Automatic gait optimization with gaussian process regression." in *Int. Joint Conference on Artificial Intelligence*, 2007.

[19] R. Antonova *et al.*, "Deep kernels for optimizing locomotion controllers," *arXiv:1707.09062*, 2017.

[20] G. H. Dunteman, *Principal components analysis*, 1989.

[21] H. Wold, "Partial least squares," *Encyclopedia of statistical sciences*, vol. 9, 2004.

[22] A. Marco *et al.*, "Automatic lqr tuning based on gaussian process global optimization," in *Int. Conf. on Robotics and Automation*, 2016.

[23] J. Spitz *et al.*, "Trial-and-error learning of repulsors for humanoid qp-based whole-body control," in *International Conference on Humanoid Robotics (Humanoids)*, 2017.

[24] C. Li *et al.*, "High dimensional bayesian optimization using dropout," in *International Joint Conf. on Artificial Intel.*, 2017.

[25] R. Akrour *et al.*, "Local bayesian optimization of motor skills," in *International Conf. on Machine Learning*, 2017.

[26] J. C. Bezdek and R. J. Hathaway, "Convergence of alternating optimization," *Neural, Parallel & Scientific Computations*, 2003.

[27] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of optimization theory and applications*, 2001.

[28] N. Hansen *et al.*, "Coco: A platform for comparing continuous optimizers in a black-box setting," *arXiv:1603.08785*.

[29] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes in Machine Learning*. MIT Press, 2006.

[30] P. Auer, "Using confidence bounds for exploitation-exploration trade-offs," *Journal of Machine Learning Research*, 2002.

[31] N. Srinivas *et al.*, "Gaussian process optimization in the bandit setting: No regret and experimental design," *arXiv:0912.3995*, 2009.

[32] S. Caron *et al.*, "Stability of surface contacts for humanoid robots: Closed-form formulae of the contact wrench cone for rectangular support areas," in *Int. Conference on Robotics and Automation*, 2015.

[33] N. A. Radford *et al.*, "Valkyrie: NASA' s First Bipedal Humanoid Robot," *Journal of Field Robotics*, 2014.

[34] K. Yuan and Z. Li, "An improved formulation for model predictive control of legged robots for gait planning and feedback control," *International Conference on Intelligent Robots and Systems*, 2018.

[35] Y. Sui *et al.*, "Safe exploration for optimization with gaussian processes," in *International Conference on Machine Learning*, 2015.