

Safe Risk-averse Bayesian Optimization for Controller Tuning

Christopher König¹, Miks Ozols², Anastasia Makarova³,
Efe C. Balta¹, Andreas Krause³, Alisa Rupenyan⁴

Abstract—Controller tuning and parameter optimization are crucial in system design to improve both the controller and underlying system performance. Bayesian optimization has been established as an efficient model-free method for controller tuning and adaptation. Standard methods, however, are not enough for high-precision systems to be robust with respect to *unknown* input-dependent noise and stable under safety constraints. In this work, we present a novel data-driven approach, RAGoOSe, for safe controller tuning in the presence of heteroscedastic noise, combining safe learning with risk-averse Bayesian optimization. We demonstrate the method for synthetic benchmark and compare its performance to established BO-based tuning methods. We further evaluate RaGoose performance on a real precision-motion system utilized in semiconductor industry applications and compare it to the built-in auto-tuning routine.

I. INTRODUCTION

High-precision, high-stakes systems operation often requires risk-averse parameter selection, even if this results in a trade-off with the best expected performance of the systems. The standard approach in tuning the control parameters of such systems assumes constant observation noise, and tuning is often based on a model-based or rule-based procedure, thus limiting the system's performance. The goal of this work is to demonstrate a novel *flexible, risk-averse* approach to find optimal control parameters under input-dependent noise, based only on observation of the system performance.

Optimizing performance indicators derived from system data for tuning of control parameters has been explored through various approaches, e.g., iterative feedforward tuning [1], variable gain selection [2], data-driven feedforward learning [3]. The performance criteria can be represented by features in the data measured during system operation at different values of the controller parameters [4]. Then, the low-level controller parameters can be optimized to fulfill the desired performance criteria. Bayesian optimization (BO) [5] denotes a class of sample-efficient, black-box optimization algorithms that have been used to address a wide range of problems, see [6] for a review. BO has been successfully demonstrated in controller tuning with custom unknown objectives and constraints in the existing literature [7], [8].

An important aspect for practical applicability of BO is safety. In this work, we follow the definition for *safety*

This project was funded by the Swiss Innovation Agency, grant Nr. 46716, and by the Swiss National Science Foundation under NCCR Automation, grant Nr. 180545.

¹ Control and Automation Group, Inspire AG, Zurich, Switzerland

² Automatic Control Laboratory, ETH Zurich, Switzerland

³ Learning & Adaptive Systems group, ETH Zurich, Switzerland

⁴ ZHAW Centre for Artificial Intelligence, ZHAW Zurich University of Applied Sciences, Switzerland

from [9] as optimization under constraints, separate from the objective. BO has been combined with heteroscedastic noise for controller optimization [10] to find optimal hyperparameters of a stochastic model predictive controller, however, the safety aspect of the approach was not studied. Our approach considers safety of the underlying system and combines aspects of safe BO and risk-averse BO. This ensures that there are no constraint violations during the optimization, and enables the application of the approach to continuous optimization, e.g., [11], [12] concerning controller parameter adaptation for systems operating under changing conditions.

Contribution. In this work, we make the following contributions: (1) we propose a Bayesian optimization algorithm for safe policy search in the presence of heteroscedastic noise. The noise affects the surrogate models used in the data-driven constrained optimization procedure, which are built using features from the underlying noisy data, and the exploration-exploitation strategy. (2) We demonstrate the approach for a benchmark problem with safety constraints and compare the achieved performance to that achieved without accounting for the noise based on [11], and that achieved with another constrained BO approach [13]. (3) We demonstrate the approach on a numerical simulation of a high-precision motion system, and apply it in practice to tune the controller on the real system (Fig. 3), used in semiconductor manufacturing, in comparison with the industry standard tuning approach, via the built-in auto-tuner in the system.

II. RELATED WORK

BO for controller tuning. BO has been applied in the tuning of various types of controllers for fast motion systems. BO-based tuning for enhanced performance has been demonstrated for cascade controllers of linear axis drives, where data-driven performance metrics have been used to intentionally increase the traversal time and the tracking accuracy while reducing vibrations [14]. In model predictive control (MPC), instead of adapting the controller for the worst-case scenarios, the prediction model can be selected to provide the best closed-loop performance by tuning the parameters in the MPC optimization objective for maximum performance [15]–[17].

Safe BO. To ensure optimal control performance for time-varying systems, controllers should adapt their parameters over time, while avoiding unsafe, or unstable parameters. Especially in high-precision motion systems, even one iteration with excessive vibrations is not allowed during operation. Thus, learning the parameters of cascaded controllers with safety and stability constraints is needed to ensure that only safe parameters are explored [11], [12]. The SafeOpt

algorithm [18] achieves safety, but is inefficient due to its exploration strategy [9]. This has been addressed in [19], where the safe set is not actively expanded, which may compromise optimality but works well for the considered application. Another approach applied to controller tuning, is to add a safety-related log-barrier in the cost, as demonstrated in [20], to reduce constraint violations as compared to an implementation based on BO with inequality constraints [13]. An efficient way enabling safe exploration is GoOSE (Goal Oriented Safe Exploration) [21], ensuring that every input to the system satisfies an unknown, but observable constraint. For controller tuning, GoOSE unifies time-varying Gaussian process bandit optimization [22] with multi-task Gaussian processes [23], and with efficient safe set search based on particle swarm optimization, enabling its application in the adaptive tuning of motion systems [11].

Tuning in the presence of heteroscedastic noise.

Input-dependent noise and lack of repeatability are common in robotic and motion systems. The main challenge is that both the objective and noise are unknown. Taking it into account in controller design leads to performance improvement, as demonstrated in the MPC-based controller design of robotic systems. For example, [10] and [24] account for heteroscedastic noise via an additional flexible parametric noise model. These approaches, however, utilize a restrictive model of the noise variance.

Alternatively, [25] focus on the risk associated with querying noisy points and proposing RAHBO (Risk-Averse Heteroscedastic Bayesian Optimization), a flexible approach enjoying theoretical convergence guarantees. In contrast to the standard BO paradigm that optimizes the expected value only and might fail in the risk-averse setting, RAHBO trades the mean and input-dependent variance of the objective, while learning both the objective and the noise distribution on the fly. To this end, RAHBO extends the concentration inequalities in the case of heteroscedastic noise [26] to the setting of unknown noise following the optimism-under-uncertainty principle [27].

III. PRELIMINARIES AND PROBLEM STATEMENT

This section introduces the optimization problem of interest and related preliminaries.

Classical optimization objective. Consider a problem of sequentially optimizing a fixed and unknown objective $f : \mathcal{X} \rightarrow \mathbb{R}$ over the compact set of inputs $\mathcal{X} \subset \mathbb{R}^d$:

$$\min_{x \in \mathcal{X}} f(x). \quad (1)$$

At every iteration t , for a chosen input $x_t \in \mathcal{X}$, one observes a noise-perturbed evaluation y_t resulting into the dataset $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^t$, with:

$$y_t = f(x_t) + \varepsilon(x_t), \quad (2)$$

where $\varepsilon(x_t)$ is zero-mean noise, independent across different time steps t . Note that if the standard noise independence assumption does not hold, additional terms in the noise model covariance need to be included for Gaussian noise models. In the standard setup, the noise is assumed to be identically

distributed over the whole domain \mathcal{X} , i.e., $\varepsilon_t \sim \mathcal{N}(0, \rho^2)$. This, however, might be restrictive in practice, leading to sub-optimal solutions [25]. Here we present the generalized version of the optimization with $\varepsilon_t \sim \mathcal{N}(0, \rho^2(x_t))$, where $\rho^2(\cdot)$ denotes the heteroscedastic variance as a function of its argument throughout the text, and later describe why it is important.

Bayesian optimization employs two main ingredients: *probabilistic model* (for uncertainty quantification) and *acquisition function* (for sampling policy of the next inputs).

Probabilistic model. Gaussian processes (GPs) [28] provide a distribution over the space of functions commonly used in non-parametric Bayesian regression. Posterior GP mean and variance denoted by $\mu_t(\cdot)$ and $\sigma_t^2(\cdot)$, respectively, are computed based on the previous measurements $y_{1:t} = [y_1, \dots, y_t]^\top$ and a given kernel $\kappa(\cdot, \cdot)$:

$$\mu_t(x) = \kappa_t(x)^T (K_t + \Sigma_t)^{-1} y_{1:t}, \quad (3)$$

$$\sigma_t^2(x) = (\kappa(x, x) - \kappa_t(x)^\top (K_t + \Sigma_t)^{-1} \kappa_t(x)), \quad (4)$$

where $\Sigma_t = \text{diag}(\rho^2(x_1), \dots, \rho^2(x_t))$, $(K_t)_{i,j} = \kappa(x_i, x_j)$, $\kappa_t(\cdot)^\top = [\kappa(x_1, \cdot), \dots, \kappa(x_t, \cdot)]^\top$, $f \sim GP(\mu_0, \lambda^{-1}\kappa)$ is a prior.

As long as the GP-based model of the objective f is well-calibrated, we can use it to construct high-probability confidence bounds for f [27]. In particular, as long as f has a bounded norm in the reproducing kernel Hilbert space (RKHS) associated with the covariance function κ used in the GP, f is bounded by lower and upper confidence bounds $\text{lcb}_t^f(x)$ and $\text{ucb}_t^f(x)$, respectively,

$$\begin{aligned} \text{lcb}_t^f(x) &= \mu_t(x) - \beta_t^f \sigma_t^f(x), \\ \text{ucb}_t^f(x) &= \mu_t(x) + \beta_t^f \sigma_t^f(x), \end{aligned} \quad (5)$$

where $\beta_t^f > 0$ is a parameter that ensures validity of the confidence bounds [26].

Acquisition function $\text{acq}_t : \mathcal{X} \rightarrow \mathbb{R}$ expresses the informativeness of an input x about the location of objective optimum, given the probabilistic model of f . The goal of the acquisition function is to trade off *exploration*, i.e., learning about uncertain inputs, and *exploitation*, i.e., choosing inputs that lead to low cost function values. Thus, BO reduces the original black-box optimization problem to cheaper problems $x_t = \arg\min_{x \in \mathcal{X}} \text{acq}_t(x)$. One of the most popular acquisition functions, GP-LCB (Lower Confidence Bound) [27], uses the so-called principle of *optimism in the face of uncertainty*. The idea is to rely on an optimistic guess of $f(x)$ via the lower confidence bound in Eq. (5) and choose x_t with the lowest guess. Algorithm 1 provides the standard BO loop.

Algorithm 1 Bayesian Optimization loop

- 1: Prior $f \sim GP(0, \kappa)$
 - 2: **for** $t = 1, \dots, T$ **do**
 - 3: $x_t \leftarrow \arg\min_{x \in \mathcal{X}} \text{acq}_t(x)$
 - 4: Observe $y_t \leftarrow f(x_t) + \varepsilon(x_t)$
 - 5: Update GP posterior with y_t
-

A. Safe optimization under constraints

Consider a safety metric $q : \mathcal{X} \rightarrow \mathbb{R}$, which should stay bounded within a predefined region with high probability. Formally, the optimization problem is then

$$\min_{x \in \mathcal{X}} \{f(x) \mid q(x) \leq c\}, \quad (6)$$

under the feasibility assumption $\{x \mid q(x) \leq c\} \neq \emptyset$, for a given $c \in \mathbb{R}$. The safety metric $q(x)$ here is an unknown, expensive-to-evaluate function that should be learned on the fly. *GoOSE* extends any standard BO algorithm providing high-probability safety guarantees in such a setup [11].

GoOSE keeps track of two subsets of the optimization domain: *the safe set* \mathcal{X}_t and *the optimistic safe set* \mathcal{X}_t^{opt} . \mathcal{X}_t contains the set of points classified as safe while \mathcal{X}_t^{opt} contains the points that potentially could be safe, i.e.,

$$\mathcal{X}_t = \{x \in \mathcal{X} \mid \text{lcb}_t^q(x) \leq c\}, \quad (7)$$

$$\mathcal{X}_t^{opt} = \{x \in \mathcal{X} \mid \exists \bar{x} \in W_t, \text{ s.t. } g_t^\epsilon(\bar{x}, x) > 0\}. \quad (8)$$

Here, the subscript t indicates the iteration of the Bayesian optimization, and the superscript q marks the constraint $q(x)$. $W_t \subset \mathcal{X}_t$ is the set of expanders - the periphery of the safe set such that $\forall x \in W_t$ it holds true that $|\text{lcb}_t^q(x) - \text{lcb}_t^q(\bar{x})| > \epsilon$, and $g_t^\epsilon(\bar{x}, x)$ is the noisy expansion operator taking values of 0 or 1, defined as:

$$g_t^\epsilon(\bar{x}, x) = \mathbb{I} [\text{lcb}_t^q(\bar{x}) + \|\mu_t^\nabla(\bar{x})\|_\infty d(\bar{x}, x) + \epsilon \leq c], \quad (9)$$

for some $\epsilon > 0$ uncertainty threshold relating to the observation uncertainty of $q(x)$ (see also [11]). $\|\mu_t^\nabla(\bar{x})\|$ is the mean of the posterior over the gradient of the constraint function, and $d(\cdot, \cdot)$ is the distance metric associated with Lipschitz continuity of $q(x)$. Prior to the initialization of BO, *GoOSE* assumes a known non-empty safe seed \mathcal{X}_0 , which does not need to be large in size and can be constructed from previous measurements.

The acquired points belong to the joint domain $\mathcal{X}_t \cup \mathcal{X}_t^{opt}$

$$x_t = \underset{x \in \{\mathcal{X}_t \cup \mathcal{X}_t^{opt}\}}{\operatorname{argmin}} \text{acq}_t(x). \quad (10)$$

If $x_t \in \mathcal{X}_t$, $f(x_t)$ and $q(x_t)$ are evaluated via available observations. Otherwise, if $x_t \in \mathcal{X}_t^{opt}$, then an expander of the safe set is evaluated. This strategy allows both to query the points satisfying the constraints and strategically expand the safe set in the direction of a promising input.

B. Risk-averse optimization

Consider two different solutions that have similar expected function values but one produces noisier realizations (see Section V as a toy example). This is important when it comes to the actual deployment of the solutions in a real system where one might prefer more stable solutions. While standard risk-neutral BO methods, such as GP-LCB, provide guarantees in the risk-neutral (homoscedastic/heteroscedastic) BO setting [27], they might fail in the risk-averse setting. To this end, [25] proposes a practical risk-averse algorithm with theoretical guarantees.

Mean-variance objective is a simple and frequently used way to incorporate risk:

$$\min_{x \in \mathcal{X}} \{MV(x) = f(x) + \alpha \rho^2(x)\} \quad (11)$$

with $\alpha \geq 0$ is the *coefficient of absolute risk tolerance*. Intuitively, the noise variance $\rho^2(x)$ acts as a penalizing term to the objective $f(x)$, encouraging to acquire points with less noise. The coefficient of absolute risk-tolerance α allows accounting for various degrees of risk tolerance, with $\alpha = 0$ corresponding to the risk-neutral case of using GP-LCB.

Acquisition function. Since the noise variance is unknown in practice and both $f(x)$ and $\rho(x)$ are learned during the optimization, *RAHBO* proposes the acquisition strategy trading off not only exploration-exploitation but also risk:

$$\text{acq}_t(x) = \text{lcb}_t^f(x) + \alpha \text{lcb}_t^{var}(x), \quad (12)$$

where lcb_t^{var} denotes the lower confidence bound Eq. (5) constructed for the variance $\rho^2(x)$. For the latter, one need to observe evaluations s_t^2 of the noise variance, which is done by querying k observations $\{y_t^i\}_{i=1}^k$ of $f(x)$ at x_t :

$$y_t = \frac{1}{k} \sum_{i=1}^k y_t^i, \quad s_t^2 = \frac{1}{k-1} \sum_{i=1}^k [y_t - y_t^i(x_t)]^2. \quad (13)$$

The empirical variance s_t^2 serves as an unbiased estimator for $\rho^2(x_t)$, and it is used for constructing GP^{var} , explained below in Section IV. For a complete introduction and treatment of *RAHBO*, we refer the reader to [25]. Having introduced *RAHBO*, constituting an important part of the framework presented in this work, we are ready to present *RAGoOSE* – our Safe Risk-averse Bayesian Optimization method.

IV. RAGoOSE: SAFE RISK-AVERSE BO

We aim to minimize the risk-averse mean-variance objective subject to unknown safety constraint function. Similar techniques could be used to capture heteroscedastic noise in the constraint measurements. Formally,

$$\min_{x \in \mathcal{X}} \{MV(x) \mid q(x) \leq c\}, \quad (14)$$

for a given $c \in \mathbb{R}$. As before, we get noise-perturbed evaluations as follows: $m(x) = q(x) + \varepsilon_q$ with $\varepsilon_q \sim \mathcal{N}(0, \sigma_q^2)$, $\sigma_q^2 > 0$, $y(x) = f(x) + \varepsilon(x)$ with $\varepsilon(x) \sim \mathcal{N}(0, \rho^2(x))$ and similarly $s^2(x) = \rho^2(x) + \varepsilon_{var}$ with $\varepsilon_{var} \sim \mathcal{N}(0, \sigma_{var}^2)$. The latter is observed through sample mean y_t and variance s_t^2 defined in Eq. (13).

A. Building blocks of RAGoOSE

The main elements enabling the safe and risk-aware optimization of *RAGoOSE* rely on *RAHBO* and *GoOSE*.

RAGoOSE probabilistic models. To reason under uncertainty about the unknown objective $f(x)$, noise variance $\rho^2(x)$ and safety metrics $q(x)$, we model each with a GP - GP^f , GP^{var} , and GP^q , respectively, characterised by a kernel function κ^f , κ^{var} , and κ^q , respectively.

The posteriors of GP_t^{var} and GP_t^q can be simplified since the variance $\rho^2(x)$ and safety metrics $q(x)$ are assumed to be perturbed by homoscedastic noise:

$$\begin{aligned}\mu_t^q(x) &= \kappa_t^q(x)^T(K_t^q + \sigma_q^2 I)^{-1}y_{1:t}, \\ \sigma_t^{q2}(x) &= \kappa_t^q(x,x) - \kappa_t^q(x)^T(K_t^q + \sigma_q^2 I)^{-1}\kappa_t^q(x),\end{aligned}$$

where $I \in \mathbb{R}^{t \times t}$ is the identity matrix, and the rest corresponds to the notation in Eqs. (3) and (4). The posterior mean and variance equations for GP_t^{var} are built analogously.

The posterior GP_t^f in Eqs. (3) and (4), however, relies on the unknown noise variance $\rho^2(x)$ in Σ_t . To this end, [25] proposes instead to use $\hat{\Sigma}_t$ with the upper confidence bound $ucb_t^{var}(\cdot)$:

$$\hat{\Sigma}_t := \frac{1}{k} \text{diag}(ucb_t^{var}(x_1), \dots, ucb_t^{var}(x_t)), \quad (15)$$

where $\hat{\Sigma}_t$ is corrected by k due to the sample mean. This correction guarantees correct, though conservative, confidence bounds for the objective $f(x)$, thus allowing for effective optimization (see [25] for more details).

We assume the functions $f(x)$, $q(x)$, and $\rho^2(x)$ to belong to RKHS (Reproducing kernel Hilbert space) associated by their respective kernels and have bounded respective RKHS-norms, other assumptions follow [11], [25]. Under these assumptions, the lower and upper confidence bounds for the functions are defined according to Eq. (5) and denoted as $lcb_t^q(x)$, $ucb_t^q(x)$ for $q(x)$ and $lcb_t^{var}(x)$, $ucb_t^{var}(x)$ for $\rho^2(x)$. The corresponding parameters β^f , β^q , and β^{var} are individually adjusted to ensure the validity of confidence bounds and balance exploration vs. exploitation (further referenced collectively as β parameters).

RAGoOSE acquisition strategy. With the confidence bounds defined, RAGoOSE acquires point x_t at each iteration t following the acquisition strategy in Eq. (12) conditioned on the safety of the inputs. Particularly, x_t is obtained by:

$$x_t = \underset{x \in \{\mathcal{X}_t \cup \mathcal{X}_t^{opt}\}}{\operatorname{argmin}} \{lcb_t^f(x) + \alpha lcb_t^{var}(x)\} \quad (16)$$

As described in [25], the framework allows for fine-tuning the trade-off between exploration, exploitation, and risk. This is exactly done via the parameter β^f , allowing to adjust exploration vs. exploitation trade-off of the objective, and α , adjusting the trade-off between optimum value of the expected objective and the associated observation variance at this point. While higher values of β^f result in a more exploratory behaviour of the algorithm, higher values of α result in a more risk-aware algorithm. However, the parameters β^{var} and β^q act as tuning knobs to control exploration. As $\text{acq}_t(x)$ is using lcb_t^{var} , an increasing β^{var} corresponds to a more optimistic belief on the observation noise variance of $f(x)$, whereas an increasing β^q corresponds to a smaller safe set \mathcal{X}_t , and hence, to a more conservative estimate of the true safe set \mathcal{X}_{safe} via \mathcal{X}_t , where \mathcal{X}_{safe} is defined as $\mathcal{X}_{safe} = \{x \in \mathcal{X} \mid q(x) \leq c\}$. RAGoOSE keeps track of its own estimate of the safe set \mathcal{X}_t and the optimistic safe set \mathcal{X}_t^{opt} , defined in the same way as described in section III.

B. The Algorithm

Initialization. We assume the compact optimization domain $\mathcal{X} \subset \mathbb{R}^d$ to be defined and known, and an initial set of n_0 observations $\mathcal{D}_f = \{y_i\}_{i=1}^{n_0}$, $\mathcal{D}_q = \{m_i\}_{i=1}^{n_0}$, and $\mathcal{D}_{var} = \{s_i\}_{i=1}^{n_0}$ for $f(x)$, $q(x)$, and $\rho^2(x)$ at $x \in \mathcal{X}_0 \subseteq \mathcal{X}_{safe} \subseteq \mathcal{X}$ must be available. We use \mathcal{X}_0 as a safe-seed set required for RAGoOSE initialization, and as a fallback input in case no safe inputs are found through RAGoOSE iterations. Given the initialization sets, we initialize the three Gaussian processes GP^f , GP^{var} and GP^q with pre-selected kernels for the GPs and tune their hyperparameters.

We then configure the RAGoOSE optimization with our selected values of α and β , defining the exploration-exploitation-risk aversion setting. Additionally, we select appropriate uncertainty threshold parameter ϵ governing the augmentation of the set of expanders, via the noisy expansion operator g_t^ϵ . We also set the number of evaluations per iteration k , and the total number of iteration loops of RAGoOSE, T .

Optimization loop. Having set the configuration and all parameters and input sets initialized, we present the main algorithm of our work, representing an end-to-end execution of RAGoOSE, in Algorithm 2. We note that an explicit modeling of the variance in the constraint evaluations is not included, though a possible extension of the algorithm. Furthermore, the expansion strategy (see Algorithm 2 line 12-15) is not risk-averse, since the expanders are only evaluated to potentially enable an optimizer for evaluation. As final result the pessimistic optimum of the GPs is chosen (see Algorithm 2 line 16), to ensure that the optimal cost has both low mean and low uncertainty predictions.

Algorithm 2 RAGoOSE optimization loop

```

1: Input: Parameters  $\alpha, \beta, k, \epsilon$ ,  

   Initial data  $\mathcal{X}_0, \mathcal{D}_f, \mathcal{D}_{var}, \mathcal{D}_q$ ,  

   Kernel functions  $\kappa^f, \kappa^q, \kappa^{var}$ ,  

   Prior means  $\mu_0^f, \mu_0^q, \mu_0^{var}$ 
2: for  $t = 1, 2, \dots$  do
3:   update  $GP^f, GP^{var}, GP^q$ 
4:    $\mathcal{X}_t \leftarrow \{x \in \mathcal{X} : u_t^q(x) \leq c\}$ 
5:    $L_t \leftarrow \{x \in \mathcal{X}_t : \exists z \notin \mathcal{X}_t \text{, with } d(x, z) \leq \Delta x\}$ 
6:    $W_t \leftarrow \{x \in L_t : u_t^q(x) - l_t^q(x) \geq \epsilon\}$ 
7:   if  $x_{t-1} = x_{opt}$  then
8:      $x_{opt} \leftarrow \underset{x \in \{\mathcal{X}_t \cup \mathcal{X}_t^{opt}\}}{\operatorname{argmin}} \{lcb_t^f(x) + \alpha lcb_t^{var}(x)\}$ 
9:   if  $ucb_t^q(x) \leq c$  then
10:     $x_t \leftarrow x_{opt}$ 
11:    evaluate  $y_t, m_t, s_t^2$ 
12:   else
13:     $x_{exp} \leftarrow \operatorname{argmin}_{x \in W_t} d(x, x_{opt})$ :  $g_t^\epsilon(x, x_{opt}) \neq 0$ 
14:     $x_t \leftarrow x_{exp}$ 
15:    evaluate  $y_t, m_t, s_t^2$ 
16: Output:  $x_{final}^* \leftarrow \underset{x \in \mathcal{X}_t}{\operatorname{argmin}} \{ucb_t^f(x) + \alpha ucb_t^{var}(x)\}$ 

```

C. Implementation

When optimizing the acquisition function, the current implementation of RAGoOSE uses particle swarm optimization (PSO) [29], that initializes a set of particles across \mathcal{X}_t , and through its update rules also allows the particles to expand into the optimistic safe set \mathcal{X}_t^{opt} , avoiding calculating the optimistic safe set explicitly [11].

Another important aspect affecting the performance of the algorithm is the chosen discretization size Δx , as finer discretization allows a better assumption of \mathcal{X}_t^{opt} , due to a finer grid of expanders, while slowing down the computation time of the algorithm. We follow the rule-of-thumb in selection of the discretization step from [7] based on the lengthscale parameter ℓ^q of the GP^q kernel κ^q , with Δx s.t. $\kappa^q(x, x + \Delta x) (\sigma_q)^{-2} = 0.95$.

The parameter $\beta = 3$ was fixed for every GP of the implementation, resulting in a 99% confidence interval. The exploration threshold was set to $\epsilon = 6\sigma_q$ for all experiments (see also Algorithm 2 line 6). The observation window $k = 10$ was chosen empirically, balancing the evaluation time and the representativeness of the sample mean y_t and variance s_t^2 of the true mean and variance at x_t .

V. CASE STUDY

We now apply the proposed approach, first on an illustrative example (sinusoidal function with varying noise levels), then on a numerical simulation of a precision motion system, and finally we run experiments on the real system. In the synthetic example the hyperparameters were tuned every iteration minimizing negative log marginal likelihood. The hyperparameters we used for the GPs in the other experiments can be read from:

Numeric experiment	f	q	ρ^2
$l = [P_{kp}, V_{ki}]^T$ $(\sigma_n, \mu_0, \lambda)$	$[50, 200]^T$ (2, 6, 0)	$[50, 200]^T$ (4, 6, 1e-2)	$[50, 200]^T$ (1e-2, 0.3, 1e-6)
Real experiment	f	q	ρ^2
$l = [P_{kp}, V_{ki}, A_{ff}]^T$ $(\sigma_n, \mu_0, \lambda)$	$[50, 100, 200, 0.5]^T$ (1, 4.61, 0)	$[50, 100, 200, 0.5]^T$ (1e-2, 0.192, 9e-4)	$[50, 100, 200, 0.5]^T$ (1e-3, 1e-2, 1e-6)

TABLE I: GP Hyperparameters for experiments in section V

All experiments use a RBF kernel $\kappa(x, x') = \sigma_n^2 \exp(-\frac{(x-x')^2}{2l^2})$ for all GPs. All priors are set to a constant μ_0 . For the cost function GP^f the noise is modeled by GP^{var} .

A. Synthetic problem

We study *RAGoOSE* on a synthetic optimization problem, and compare it with two baselines - *GoOSE* [11] and Constrained Bayesian Optimization (*CBO*) [13].

Problem Setup. We use a sinusoidal cost function $f(x)$ and observe its noisy evaluations $y(x)$, with the cost observation noise variance function $\rho^2(x)$ as shown in Fig. 1. Similarly, we observe the constraint function $q(x)$ via its noisy observations $m(x)$, with a homoscedastic noise variance of $\sigma_q = 0.1$. Performance was assessed by repeating the optimization 30 times, with $T = 200$ BO iterations each, and $k = 10$ evaluations of the cost function. We repeat the optimization routine with *RAGoOSE* for all $\alpha \in \{0, 10, 30\}$. As Section V illustrates, the set-up gives rise to 3 optimizers - x_{CV}^* , that violates the constraint, x_{opt}^* , located at the low observation noise variance region of the domain, and x_{HV}^* , located in the high-variance region. All GPs were initialized with five a priori known points within the safe set interval $x \in [0, 1]$. In Fig. 2 we report the mean cost function $f(x_t)$, observation noise variance $\rho^2(x_t)$,

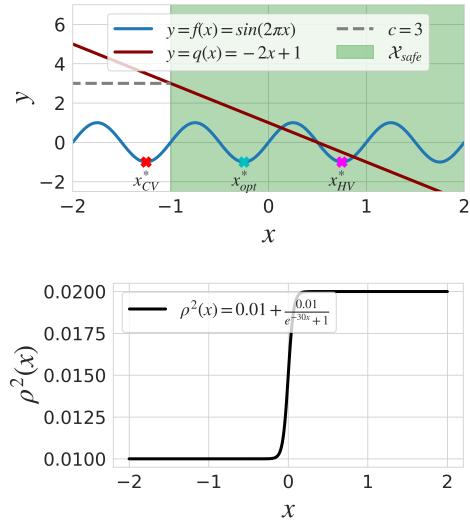


Fig. 1: Illustrative example in the case study. Top: Objective $f(x)$ with 3 global maxima marked as $(x_{CV}, x_{opt}, x_{HV})$ and constraint function $q(x)$ over the same domain showing that only two optima (x_{opt}, x_{HV}) are safe; Bottom: Heteroscedastic noise variance $\rho^2(x)$ over the same domain: the noise level at $(x_{CV}, x_{opt}, x_{HV})$ varies according to the sigmoid function. with the best performing *RAGoOSE* configuration and the benchmarks, and the intermediate regret r_t with varying α (with r_t defined in Eq. (17)) with ± 2 standard errors (SE) across the $n = 30$ performed repetitions of the optimization routine

$$r_t = [f(x_{opt}^*) + 50\rho^2(x_{opt}^*)] - [f(x_t) + 50\rho^2(x_t)]. \quad (17)$$

We report a constraint violation when the observed value m_t exceeds the constraint upper bound $c = 3$.

Results. All methods were able to converge to the optimum of the cost function successfully, as shown in Fig. 2. While *RAGoOSE* requires more iterations to converge, it shows superior performance in observation noise variance, converging to x_{opt}^* , yielding a 41% and 31% decrease in observed noise variance, compared with *GoOSE* and *CBO* respectively. It is shown, that *CBO* results in a significantly higher rate of constraint violations, because instead of only considering points that are with high probability safe as in *GoOSE*, *CBO* only weighted the acquisition function with the probability of feasibility for each candidate x . We observe a closer convergence to the low-variance optimum with increasing α , as Fig. 2c reports, with $\alpha = 50$ yielding the lowest regret at the terminal iteration. As expected, the risk-neutral case of $\alpha = 0$ fails to converge to the risk-averse optimum x_{opt}^* , yielding the highest intermediate regret r_t . *RAGoOSE* outperforms the other benchmarks in the number of constraint violations. The benchmark study results are summarized in Table II. The additional computation due to modeling the noise variance slows down the optimization time of *RAGoOSE*, but increases the performance and reduces constraint violations. Furthermore *RAGoOSE* shows slightly better prediction of the final optimum mean value, within one standard deviation of the noise.

B. Controller Tuning for Precision Motion

We now introduce the motion system, its controller, and the optimization setup, and then proceed with numerical results, followed by the experimental implementation.

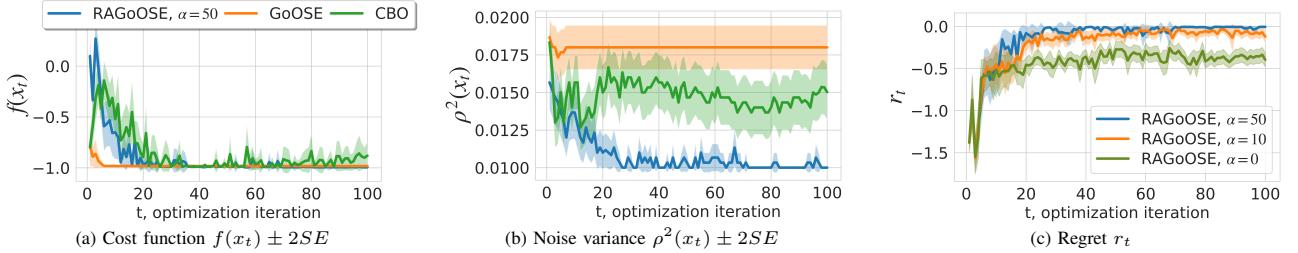


Fig. 2: Numerical study results for synthetic function (see Figure 1), number of iterations $t = 100$, and $n_{rep} = 30$ repetitions for each experiments, with reported average and two standard errors (SE). RAGOOSE successfully converges to the minimum of the cost function (a), while obtaining a lower observation noise at the solution (b) compared to the benchmarks. (c) Regret convergence as a function of the α parameter.

Metric/Parameter	RAGOOSE	GoOSE	CBO
$\overline{f(x^*)}$	-0.999	-0.987	-0.999
$\overline{\rho^2(x^*)}$	0.011	0.018	0.016
Constraint violations	0.03%	0.07%	7.90%
Mean optimization time [s]	68.6	28.4	21.4

TABLE II: Mean optimized performance metric (cost) $\overline{f(x^*)}$, mean noise parameter $\overline{\rho^2(x^*)}$, constraint violations and optimization time for different BO-based tuning methods applied on the synthetic problem across the 30 optimization experiments

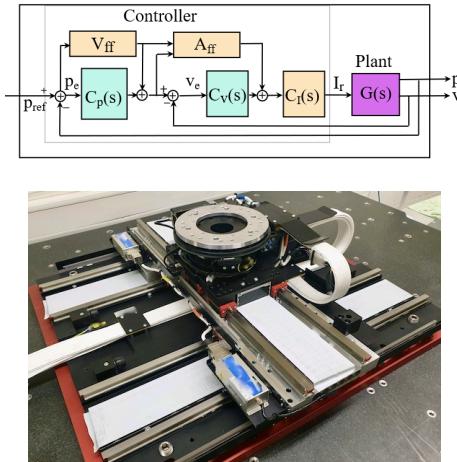


Fig. 3: Top: Simplified controller architecture in the experimental study. Bottom: Positioning system by Schneeberger Linear Technology AG.

System and controller. The system of interest is a linear axis of a high precision positioning system by Schneeberger Linear Technology, shown in Fig. 3. The axis is driven by a permanent magnet AC motor equipped with nanometer precision encoders for position and speed tracking. The axis is guided on two rails, reaching positioning accuracy of $\leq 10 \mu\text{m}$, repeatability of $\leq 0.7 \mu\text{m}$, and 3σ stability of $\leq 1 \text{ nm}$. The sampling time of the controller and the date acquisition of the system used is $f_s = 20 \text{ kHz}$. Such systems are routinely used for production and quality control in the semiconductor industry, in biomedical engineering, and in photonics and solar technologies. For example, in semiconductor manufacturing applications, considering risk or safety alone is not enough due to stringent production requirements. Therefore a tuning method that considers both process variance and safety, e.g., RAGOOSE, is beneficial.

The system is controlled by a three-level cascade controller shown in Section V-B. The outermost loop controls the position with a P-controller $C_p(s) = P_{kp}$, and

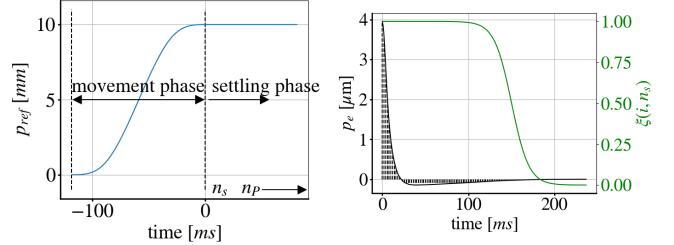


Fig. 4: Left panel: S-curve position reference function indicating move- and settle time. Right panel: position error $p_e(t_i)$ example (highlighted in black) and $\xi(i, n_s)$ filter.

the middle loop controls the velocity with a PI-controller $C_v(s) = V_{kp} + V_{ki}/s$. The inner loop, which controls the current of the drive, is well-tuned and not subject to tuning. Feedforward structures are used to accelerate the response of the system. The gain of the velocity feedforward V_{ff} is well-tuned and not modified during the retuning procedure, while the acceleration feedforward gain A_{ff} is retuned by the algorithm during the experiments on the real system and fixed to $A_{ff} = 0$ for the simulation experiments.

Optimization setup. We optimize the controller parameters providing the best tracking performance of the system, while avoiding instabilities of the controller in the presence of heteroscedastic noise. We estimate both the tracking performance and potential instabilities via the corresponding data-driven features $\phi(x)$ and $q(x)$ extracted from the system signals. The optimization problem can be written as follows:

$$\min_{x \in \mathcal{X}} \{\phi(x) \mid q(x) > c\}.$$

Specifically, the performance tracking metric is provided by $\phi(x) = \frac{1}{n_p - n_s} \sum_{i=n_s}^{n_p} |\xi(i, n_s)p_e(t_i)| + \delta(x)$. Here, $p_e(t_i)$ indicates the deviation from the reference position at each time instance t_i (see Figure 4), δ indicates if additional noise is added to the metric (only used in the numerical study). The stability (safety) constraint $q(x)$ is calculated from the Fourier transform of the filtered velocity error $\mathcal{F}[v_e(t_i)]$ and limited by a threshold c . The threshold is determined by a small number of prior experiments. Furthermore, in the experimental study, both the performance and the stability metrics are filtered by a right-sided sigmoid function $\xi(i, n_i)$ centered at 150ms after the start of the settling phase for robustness of the metric (see Figure 4). The settling phase extends from time instance n_s to time instance n_p . The optimization is over the controller gains P_{kp} , V_{kp} , V_{ki} and the feedforward gain A_{ff} .

We solve the optimization problem using RAGOOSE, following (14) and Algorithm 2. As our system exhibits

heteroscedastic noise which is included in $\phi(x)$, to provide the model for its variance i.e. $\rho^2(x)$, we need to acquire multiple repetitions at each candidate gain combination x . For observations of $f(x)$, we use the mean of the observations over the n repetitions, while we use the variance for $\rho^2(x)$. The optimum of the *RAGoOSE* experiments is found using the pessimistic prediction given by:

$$x^* = \operatorname{argmin}_{x \in \mathcal{X}_t} \{ \text{ucb}_t^f(x) + \alpha \text{ucb}_t^{var}(x) \}. \quad (18)$$

Numerical study. We perform the *RAGoOSE* optimization on a simulation of a single axis comparing the effect of $\alpha = 0$ and $\alpha = 100$ with $T = 200$ Bayesian optimization iterations and $n = 10$ evaluations of the cost function at each iteration. For simplicity we only optimize the gains $x = [P_{kp}, V_{ki}]$ in the simulation experiment. All *RAGoOSE* experiments are initialized using 3 initial samples, $x_0 = [[200, 1000], [300, 1000], [200, 1500]]^T$. To simulate the heteroscedasticity of the observations, non uniform noise is added on the cost function, dependent on one of the two parameters, e.g. for $x_1 = V_{ki}$,

$$\delta(x_1) = \begin{cases} \mathcal{N}(0, 1e-7), & x_1 \leq 1200 \\ \mathcal{N}(0, 1e-5), & x_1 > 1200. \end{cases} \quad (19)$$

Each experiment has ten repetitions for each candidate combination of parameters x , to estimate the noise variance $\rho(x)$. Furthermore, we repeat the *RAGoOSE* optimization ten times. Figure 5 shows the range of values for $f(x)$ and $\rho(x)$, obtained for each iteration during the ten repeated optimizations. The solid lines show the median over the repeated optimizations.

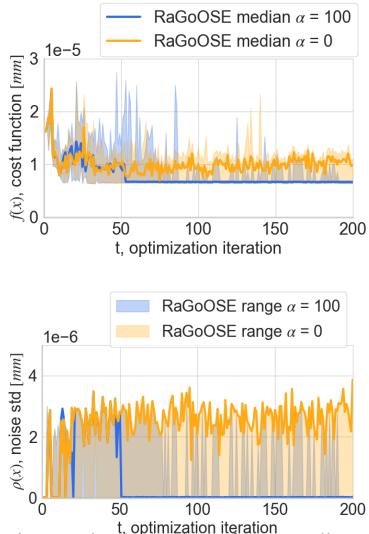


Fig. 5: Optimization results (numerical) for controller tuning comparing *RAGoOSE* using $\alpha = 0$ with $\alpha = 100$ for 10 different experiments. (a) Median and min-max interval of $f(x)$ over the 10 *RAGoOSE* optimizations for each iteration. (b) Median standard deviation of the cost $\rho(x)$ over the 10 repeated *RAGoOSE* optimizations for each iteration together with its min-max interval.

During the simulation experiment of the Argus linear axis, *RAGoOSE* using $\alpha = 0$ and $\alpha = 100$ show significant differences in performance and observation noise (see Figure 5). While both *RAGoOSE* configurations quickly find a parameter combination resulting in good performance, but high risk, *RAGoOSE* with $\alpha = 100$ continues the exploration

and finds a combination resulting in lower risk and thus also better performance, due to the effect of the noise on the cost observation. In contrast, *RAGoOSE* $\alpha = 0$ remains in a local optimum, where most of the explored configurations and their corresponding cost belong to the high noise level.

Experimental Study. We perform the optimization with *RAGoOSE* using $\alpha = 0$ and $\alpha = 100$ with $T = 100$ Bayesian optimization iterations and $n = 10$ evaluations of the cost function at each iteration. We show the distribution of the observations of the mean $f(x_t)$ and the observations of the variance $\rho^2(x_t)$ of $\phi(x)$ in Figure 6. We further

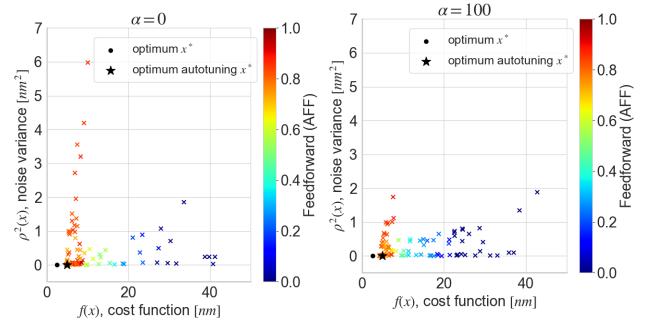


Fig. 6: Optimization results for controller tuning using *RAGoOSE* with, $\alpha = 0$ (left) and $\alpha = 100$ (right). Mean observation over the 10 repetitions $f(x)$ (x-axis) is plotted against the noise observation $\rho^2(x)$ (y-axis), for one *RAGoOSE* optimization. The resulting data points are color-coded corresponding to the values of one of the optimization variables, A_{ff} , to visualize the heteroscedasticity of the noise, which induces the particular shape of the obtained distribution with distinct "branches", corresponding to high or low noise.

compare the performance of *RAGoOSE* using $\alpha = 0$ and $\alpha = 100$ for controller tuning of the three controller gains (P_{kp}, V_{kp}, V_{ki}) and on the acceleration feedforward parameter (A_{ff}) with the performance of the auto-tuning routine of the built-in hardware controller balancing bandwidth with phase and gain margin method on the real Argus linear axis. Both *RAGoOSE* optimizations were stopped after 100 iterations. The first point of each of the two data-driven optimizations corresponds to the controller parameters found by manual tuning, with $x_0 = [200, 600, 1000, 0.0]^T$. Figure 6 shows the distribution of cost mean observations $f(x_t)$ with the corresponding variance observation $\rho^2(x_t)$. Clearly, $\alpha = 100$ avoids high risk parameter combinations compared to using $\alpha = 0$ and the preferred candidate controllers are along low variance regions, corresponding to lower values of the A_{ff} parameter which is mostly associated with heteroscedastic noise. The performance of the controllers corresponding

	RaGoOSE $\alpha = 0$	RaGoOSE $\alpha = 100$	Auto-tuning
$f(x^*)[nm]$	2.459	2.543	4.987
$\rho^2(x^*)[nm^2]$	9.187e-3	8.347e-3	5.832e-3
P_{kp}^*	330.4	319.8	350.0
V_{kp}^*	731.6	862.4	600.0
V_{ki}^*	1449.3	1174.6	2000.0
A_{ff}^*	0.83	0.815	0.0

TABLE III: Optimized performance metric (cost), noise parameter, and corresponding controller parameters for different values of α , compared to the controller's inbuilt auto-tuning in the experimental study.

to optima found by *RAGoOSE* using $\alpha = 0$ and $\alpha = 100$

(see Figure 6) after $t = 100$ iterations is similar in tracking performance (cost) and in variance. This is a result of the specific heteroscedasticity of the system's noise. Due to the distribution of the noise, the parameter regions of optimal performance correspond to the parameter regions of low risk, situated in the bottom left corner of the graphs in Figure 6. Thus, the final result of the RaGoOSE optimization is not strongly dependent of the choice of α . Both methods show significant improvement compared to the controllers tuned using the built-in auto-tuning routine (see Table III), due to the contribution of the noise model in the performance modelling. There were no constraint violations observed during the optimization of both RAGoOSE experiments. In terms of duration of the tuning, RAGoOSE needs ten times more evaluations, to estimate the noise variance, which slows it with one order of magnitude, compared to methods not considering heteroscedastic noise. In addition, as shown in Table II, it is slower due to the additional GP model of the noise. Thus, one iteration of RAGoOSE for tuning the precision motion system takes 32s, compared to 1.6s achieved with the method from [11].

VI. CONCLUDING REMARKS

We presented a risk-averse Bayesian optimization framework that performs constrained optimization with safe evaluations, accounting for the solution variance. To this end, we considered heteroscedastic noise in the process and extended the RAHBO method to consider the safety of evaluations in terms of constraint violations. We present results in both numerical benchmark studies and on a real high-precision motion system to show the improved performance over state-of-the-art methods. We demonstrate the proposed framework on a controller tuning application where safe evaluations are crucial to ensure physical safety. Future work will consider extending RAGoOSE to include the variance from the measurements of the constraints in the optimization. From the application perspective, the next challenge is applying the method in continuous optimization scenarios, without disrupting the operation of the physical system.

REFERENCES

- [1] X. Li, S.-L. Chen, J. Ma, C. S. Teo, and K. K. Tan, "Data-driven model-free iterative tuning approach for smooth and accurate tracking," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2018, pp. 593–598.
- [2] M. Li, Y. Zhu, K. Yang, and C. Hu, "A data-driven variable-gain control strategy for an ultra-precision wafer stage with accelerated iterative parameter tuning," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 5, pp. 1179–1189, 2015.
- [3] F. Song, Y. Liu, W. Jin, J. Tan, and W. He, "Data-driven feedforward learning with force ripple compensation for wafer stages: A variable-gain robust approach," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2020.
- [4] M. Khosravi, V. Behrunani, R. S. Smith, A. Rupenyan, and J. Lygeros, "Cascade control: Data-driven tuning approach based on bayesian optimization," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 382–387, 2020.
- [5] J. Mockus, *Bayesian Approach to Global Optimization: Theory and Applications*, ser. Mathematics and its Applications. Springer Netherlands, 2012.
- [6] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of Bayesian optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2015.
- [7] R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, "Constrained Bayesian optimization with particle swarms for safe adaptive controller tuning," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 11 800 – 11 807, 2017, 20th IFAC World Congress.
- [8] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic lqr tuning based on gaussian process global optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 270–277.
- [9] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *Machine Learning*, pp. 1–35, 2021.
- [10] R. Guzman, R. Oliveira, and F. Ramos, "Heteroscedastic bayesian optimisation for stochastic model predictive control," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 56–63, 2020.
- [11] C. König, M. Turchetta, J. Lygeros, A. Rupenyan, and A. Krause, "Safe and efficient model-free adaptive control via bayesian optimization," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9782–9788.
- [12] P. Brunzema, A. von Rohr, and S. Trimpe, "On controller tuning with time-varying bayesian optimization," *arXiv preprint arXiv:2207.11120*, 2022.
- [13] J. Gardner, M. Kusner, Zhixiang, K. Weinberger, and J. Cunningham, "Bayesian optimization with inequality constraints," in *Proceedings of International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 32, no. 2. Beijing, China: PMLR, 22–24 Jun 2014, pp. 937–945.
- [14] M. Khosravi, V. Behrunani, P. Myszkowski, R. S. Smith, A. Rupenyan, and J. Lygeros, "Performance-driven cascade controller tuning with bayesian optimization," *IEEE Transactions on Industrial Electronics*, p. 1–1, 2021.
- [15] D. Piga, S. Formentini, and A. Bemporad, "Direct data-driven control of constrained systems," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 4, pp. 1422—1429, Jul 2018.
- [16] F. Sorourifar, G. Makrygirgos, A. Mesbah, and J. A. Paulson, "A data-driven automatic tuning method for MPC under uncertainty using constrained Bayesian optimization," *arXiv:2011.11841*, 2020.
- [17] A. Rupenyan, M. Khosravi, and J. Lygeros, "Performance-based trajectory optimization for path following control using bayesian optimization," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 2116–2121.
- [18] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proceedings of International Conference on Machine Learning (ICML)*, 2015.
- [19] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, and A. Krause, "Safe contextual bayesian optimization for sustainable room temperature pid control tuning," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, December 2019, pp. 5850–5856.
- [20] M. Khosravi, C. Koenig, M. Maier, R. S. Smith, J. Lygeros, and A. Rupenyan, "Safety-aware cascade controller tuning using constrained bayesian optimization," *IEEE Transactions on Industrial Electronics*, 2022.
- [21] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration for interactive machine learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 2891–2901.
- [22] I. Bogunovic, J. Scarlett, and V. Cevher, "Time-varying gaussian process bandit optimization," in *Artificial Intelligence and Statistics*. PMLR, 2016, pp. 314–323.
- [23] K. Swersky, J. Snoek, and R. P. Adams, "Multi-task bayesian optimization," *Advances in neural information processing systems*, vol. 26, 2013.
- [24] R. Guzman, R. Oliveira, and F. Ramos, "Bayesian optimisation for robust model predictive control under model parameter uncertainty," *arXiv preprint arXiv:2203.00551*, 2022.
- [25] A. Makarova, I. Usmanova, I. Bogunovic, and A. Krause, "Risk-averse heteroscedastic bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 235–17 245, 2021.
- [26] J. Kirschner and A. Krause, "Information directed sampling and bandits with heteroscedastic noise," in *Conference On Learning Theory (COLT)*, 2018.
- [27] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, may 2012.
- [28] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, ser. Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, Jan. 2006.
- [29] J. Kennedy and R. Eberhart, "Particle swarm optimization," vol. 4, pp. 1942–1948 vol.4, 1995.