Safe and Efficient Model-free Adaptive Control via Bayesian Optimization

Conference Paper**Author(s):**

König, Christopher; Turchetta, Matteo; Lygeros, John ; Rupenyan-Vasileva, Alisa Bohos ; Krause, Andreas

Publication date:

2021

Permanent link:

<https://doi.org/10.3929/ethz-b-000496544>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

<https://doi.org/10.1109/ICRA48506.2021.9561349>

Funding acknowledgement:

180545 - NFS Dependable Ubiquitous Automation (SNF)

815943 - Reliable Data-Driven Decision Making in Cyber-Physical Systems (EC)

Safe and Efficient Model-free Adaptive Control via Bayesian Optimization

Christopher König^{1,*}, Matteo Turchetta^{2,*}, John Lygeros³, Alisa Rupenyan^{1,3}, Andreas Krause²

Abstract—Adaptive control approaches yield high-performance controllers when a precise system model or suitable parametrizations of the controller are available. Existing data-driven approaches for adaptive control mostly augment standard model-based methods with additional information about uncertainties in the dynamics or about disturbances. In this work, we propose a purely data-driven, model-free approach for adaptive control. Tuning low-level controllers based solely on system data raises concerns on the underlying algorithm safety and computational performance. Thus, our approach builds on GOOSE, an algorithm for safe and sample-efficient Bayesian optimization. We introduce several computational and algorithmic modifications in GOOSE that enable its practical use on a rotational motion system. We numerically demonstrate for several types of disturbances that our approach is sample efficient, outperforms constrained Bayesian optimization in terms of safety, and achieves the performance optima computed by grid evaluation. We further demonstrate the proposed adaptive control approach experimentally on a rotational motion system.

I. INTRODUCTION

Adaptive control approaches are a desirable alternative to robust controllers in high-performance applications that deal with disturbances and uncertainties in the plant dynamics. Learning uncertainties in the dynamics and adapting have been explored with classical control mechanisms such as Model Reference Adaptive Control (MRAC) [1], [2]. Gaussian processes (GP) have been also used to model the output of a nonlinear system in a dual controller [3], while coupling the states and inputs of the system in the covariance function of the GP model. Learning dynamics in an $\mathcal{L}_1$ -adaptive control approach has been demonstrated in [4], [5].

Instead of modeling or learning the dynamics, the system can be represented by its performance, directly measured from data. Then, the low-level controller parameters can be optimized to fulfill the desired performance criteria. This has been demonstrated for motion systems in [6]–[10]. Such model-free approaches, however, have not been brought to continuous adaptive control, largely because of difficulties in continuously maintaining stability and safety in the presence of disturbances and system uncertainties, and because of the associated computational complexity. Recently, a sample-efficient

extension for safe exploration in Bayesian optimization has been proposed [11]. In this paper, we further optimize this algorithm to develop a model-free adaptive control method for motion systems.

Contribution. In this work, we make the following contributions: (1) we extend the GOOSE (Goal Oriented Safe Exploration) algorithm for policy search to adaptive control problems; that is, to problems where constant tuning is required due to changes in environmental conditions. (2) We reduce GOOSE’s complexity so that it can be effectively used for policy optimization beyond simulations. (3) We show the effectiveness of our approach in extensive evaluations on a real and simulated rotational axis drive, a crucial component in many industrial machines.

II. RELATED WORK

Bayesian optimization. Bayesian optimization (BO) [12] denotes a class of sample-efficient, black-box optimization algorithms that have been used to address a wide range of problems, see [13] for a review. In particular, BO has been successful in learning high-performance controllers for a variety of systems. For instance, [14] learns the parameters of a discrete event controller for a bipedal robot with BO, while [15], trades off real-world and simulated control experiments via BO. In [16], variational autoencoders are combined with BO to learn to control an hexapod, while [17] uses multi-objective BO to learn robust controllers for a pendulum.

Safety-aware BO. Optimization under unknown constraints naturally models the problem of learning in safety-critical conditions, where *a priori* unknown safety constraints must not be violated. In [18] and [19] safety-aware variants of standard BO algorithms are presented. In [20]–[22], BO is used as a subroutine to solve the unconstrained optimization of the augmented Lagrangian of the original problem. While these methods return feasible solutions, they may perform unsafe evaluations. In contrast, the SAFEOPT algorithm [23] guarantees safety at all times. It has been used to safely tune a quadrotor controller for position tracking [6], [24]. In [7], it has been integrated with particle swarm optimization (PSO) to learn high-dimensional controllers. Unfortunately, SAFEOPT may not be sample-efficient due to its exploration strategy. To address this, many solutions have been proposed. For example, [25] does not actively expand the safe set, which may compromise the optimality of the algorithm but works well for the application considered. Alternatively, STAGEOPT [26] first expands the safe set and, subsequently, optimizes over

This project was funded by the Swiss Innovation Agency, grant Nr. 46716, by the Swiss National Science Foundation under NCCR Automation, by the ETH Max Planck Center for Learning Systems, and by the European Unions Horizon 2020 research and innovation program grant agreement Nr. 815943.

¹ Inspire AG, Zurich, Switzerland

² Learning & Adaptive Systems group, ETH Zurich, Switzerland

³ Automatic Control Laboratory, ETH Zurich, Switzerland

* The authors contributed equally.

it. Unfortunately, it cannot provide good solutions if stopped prematurely. GoOSE [11] addresses this problem by using a separate optimization oracle and expanding the safe set in a goal-oriented fashion only when necessary to evaluate the inputs suggested by the oracle.

III. SYSTEM AND PROBLEM STATEMENT

In this section, we present the system of interest, its control scheme, and the mathematical model we use for our numerical evaluations. Finally, we introduce the safety-critical adaptive control problem we aim to solve.

A. System and controller

The system of interest is a rotational axis drive, a positioning mechanism driven by a synchronous 3 phase permanent magnet AC motor equipped with encoders for position and speed tracking (see Fig. 1). Such systems are routinely used

Param.	Value	Unit
m	0.0191	kgm ²
b	30.08	kgm ² /s
c_1	$1.78e-3$	Nm
c_2	0.0295	Nm/rad
c_3	0.372	rad
c_4	$8.99e-3$	Nm
c_5	0.11	rad

Fig. 1: Rotational axis drive (left) and its parameters (right).

as components in the semiconductor industry, in biomedical engineering, and in photonics and solar technologies.

We model the system as a combination of linear and nonlinear blocks, where the linear block is modeled as a damped single mass system, following [9]:

$$G(s) := [G_p(s), G_v(s)]^T = \left[\frac{1}{ms^2 + bs}, \frac{1}{ms + b} \right]^T, \quad (1)$$

where $G_p(s)$ and $G_v(s)$ are the transfer functions respectively from torque to angular position and torque to angular velocity, m is the moment of inertia, and b is the rotational damping coefficient due to the friction. The values of m and b are obtained via least squares fitting and shown in Fig. 1.

Next, we introduce the model for the nonlinear part of the dynamics f_c , which we subtract from the total torque signal, see Fig. 2. The nonlinear cogging effects due to interactions between the permanent magnets of the rotor and the stator slots [27] are modelled using a Fourier truncated expansion as $f_c(p) = c_1 + c_2 p + \sum_{k=1}^n c_{2k+2} \sin\left(\frac{2k\pi}{c_3} p + c_{2k+3}\right)$ where p is the position, c_1 is the average thrust torque, c_2 is the gradient of the curve, c_3 is the largest dominant period described by the angular distance of a pair of magnets, n is the number of modelled frequencies, and, c_{2k+2} and c_{2k+3} are respectively the amplitudes and the phase shifts of the sinusoidal functions, for $k = 1, 2, \dots, n$. The parameters $c := [c_1, \dots, c_{2n+3}]$ are estimated using least squares error minimization between the modelled cogging torque signals and the measured torque signal at constant velocity to cancel the effects of linear dynamics. The estimates of the parameters are shown in Fig. 1.

To model the noise of the system, zero mean white noise with $6.09e-3$ Nm variance is added to the torque input signal of the plant.

The system is controlled by a three-level cascade controller shown in Figure 2. The outermost loop controls the position with a P-controller $C_p(s) = K_p$, and the middle loop controls the velocity with a PI-controller $C_v(s) = K_v(1 + \frac{1}{T_i s})$. The innermost loop controls the current of the rotational drive. It is well-tuned and treated as a part of the plant $G(s)$. Feedforward structures are used to accelerate the response of the system. Their gains are well-tuned and not modified during the retuning procedure. However, in our experiments, we perturb them to demonstrate that our method can adapt to new operating conditions by adjusting the tunable parameters of the controller.

Fig. 2: Scheme of the proposed BO-based adaptive control.

B. Adaptive control approach

The adaptive control problem that we aim to solve consists in tuning continuously the parameters of the cascade controller introduced in Section III-A to maximize the tracking accuracy of the system, following [9]. Let $\mathcal{X}$ denote the space of admissible controller parameters, $x = [K_p, K_v, T_i]$, and let $f : \mathcal{X} \rightarrow \mathbb{R}$ be the objective measuring the corresponding tracking accuracy. In particular, we define f as the position tracking error averaged over the trajectory induced by the controller $f(x) = \frac{1}{N} \sum_{i=1}^N |p_i^{\text{err}}(x)|$, where p_i^{err} is the deviation from the reference position at sampling time i . Crucially, f does not admit a closed-form expression even when the system dynamics are known. However, for a given controller $x \in \mathcal{X}$, the corresponding tracking accuracy $f(x)$ can be obtained experimentally. Notice that f can be extended to include many performance metrics, to minimise oscillations, or to reduce settling time, as in [8], [9].

In practice, we cannot experiment with arbitrary controllers while optimizing f due to safety and performance concerns. Thus, we introduce two constraints that must be satisfied at all times. The first one is a safety constraint $q_1(x)$ defined as the maximum of the fast Fourier transform (FFT) of the torque measurement in a fixed frequency window, prohibiting instabilities. The second one is a tracking performance constraint $q_2(x) = \|p^{\text{err}}(x)\|_\infty$ defined as the infinity norm of the position tracking error. Finally, in reality, the system may be subject to sudden or slow disturbances such as a change of load or a drift in the dynamics due to components wear. Our goal is to automatically tune the controller parameters of our system to maximize tracking accuracy for varying operating conditions that we cannot control, while never violating safety and quality constraints along the way.

IV. BACKGROUND

Gaussian processes. A Gaussian process (GP) [28] is a distribution over the space of functions commonly used in non-parametric Bayesian regression. It is fully described by a mean function $\mu : \mathcal{X} \rightarrow \mathbb{R}$, which, w.l.o.g, we set to zero for all inputs $\mu(x) = 0, \forall x \in \mathcal{X}$, and a kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. Given the data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^t$, where $y_i = f(x_i) + \varepsilon_i$ and $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ is zero-mean i.i.d. Gaussian noise, the posterior belief over the function f has the following mean, variance and covariance:

$$\mu_t(x) = \mathbf{k}_t^\top(x)(\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{y}_t, \quad (2)$$

$$k_t(x, x') = k(x, x') - \mathbf{k}_t^\top(x)(\mathbf{K}_t + \sigma^2 \mathbf{I})^{-1} \mathbf{k}_t(x'), \quad (3)$$

$$\sigma_t(x) = k_t(x, x), \quad (4)$$

where $\mathbf{k}_t(x) = (k(x_1, x), \dots, k(x_t, x))$, $\mathbf{K}_t$ is the positive definite kernel matrix $[k(x, x')]_{x, x' \in \mathcal{D}_t}$, and $\mathbf{I} \in \mathbb{R}^{t \times t}$ denotes the identity matrix. In the following, the superscripts f and q denote GPs on the objective and on the constraints.

Multi-task BO. In multi-task BO, the objective depends on the extended input $(x, \tau) \in \mathcal{X} \times \mathcal{T}$, where x is the variable we optimize over and τ is a task parameter set by the environment that influences the objective. To cope with this new dimension, multi-task BO uses GPs with kernels of the form $k_{\text{multi}}((x, \tau), (x', \tau')) = k_\tau(\tau, \tau') \otimes k(x, x')$, where $\otimes$ denotes the Kronecker product. This kernel decouples the correlations in objective values along the input dimensions, captured by k , from those across tasks, captured by k_τ [29].

GOOSE. GOOSE extends any standard BO algorithm to provide high-probability safety guarantees in presence of *a priori* unknown safety constraints. It builds a Bayesian model of the constraints from noisy evaluations based on GP regression. It uses this model to build estimates of two sets: the *pessimistic* safe set, which contains inputs that are safe, i.e., satisfy the constraints, with high probability and the *optimistic* safe set that contains inputs that could potentially be safe. At each round, GOOSE communicates the optimistic safe set to the BO algorithm, which returns the input it would evaluate within this set, denoted as x^* . If x^* is also in the pessimistic safe set, GOOSE evaluates the corresponding objective. Otherwise, it evaluates the constraints at a sequence of provably safe inputs, whose choice is based on a heuristic priority function, that allow us to conclude that x^* either satisfies or violates the constraints with high probability. In the first case, the corresponding objective value is observed. In the second case, x^* is removed from the optimistic safe set and the BO algorithm is queried for a new suggestion. Compared to [23], [26], GOOSE achieves a higher sample efficiency [11] while compared to [18]–[22], it guarantees safety at all times with high probability, under regularity assumptions.

GOOSE assumptions. To infer constraint and objective values of inputs before evaluating them, GOOSE assumes that these functions belong to a class of well-behaved functions, i.e., functions with a bounded norm in some reproducing kernel Hilbert space (RKHS) [30]. Based on this assumption, we can build well-calibrated confidence intervals over them. Here, we present these intervals for the safety constraint, q (the construction for f is analogous). Let $\mu_t^q(x)$ and $\sigma_t^q(x)$ denote the

posterior mean and standard deviation of our belief over $q(x)$ computed according to Eqs. (2) and (3). We recursively define these monotonically increasing/decreasing lower/upper bounds for $q(x)$: $l_t^q(x) = \max(l_{t-1}^q(x), \mu_{t-1}^q(x) - \beta_{t-1}^q \sigma_{t-1}^q(x))$ and $u_t^q(x) = \min(u_{t-1}^q(x), \mu_{t-1}^q(x) + \beta_{t-1}^q \sigma_{t-1}^q(x))$. The authors of [31], [32] show that, for functions with bounded RKHS norm, an appropriate choice of β_t^q implies that $l_t^q(x) \leq q(x) \leq u_t^q(x)$ for all $t \in \mathbb{R}^+$ and $x \in \mathcal{X}$. However, in practice, a constant value of 3 is sufficient to achieve safety [6], [11], [24].

To start collecting data safely, GOOSE requires knowledge of a set of inputs that are known *a priori* to be safe, denoted as S_0 . In our problem, it is easy to identify such set by designing conservative controllers for simplistic first principle models of the system under control.

V. GOOSE FOR ADAPTIVE CONTROL

In adaptive control, quickly finding a safe, locally optimal controller in response to modified external conditions is crucial. In its original formulation, GOOSE is not suitable to this problem for several reasons: (i) it assumes knowledge of the Lipschitz constant of the constraint, which is unknown in practice, (ii) it relies on a fine discretization of the domain, which is prohibitive in large domains, (iii) it explicitly computes the optimistic safe set, which is expensive, (iv) it does not account for external changes it cannot control. In this section, we address these problems step by step and we present the resulting algorithm.

Algorithm 1: GOOSE for adaptive control

- 1 **Input:** Safe seed S_0 , $f \sim \mathcal{GP}(\mu^f, k^f; \theta^f)$,
 $q \sim \mathcal{GP}(\mu^q, k^q; \theta^q)$, $\tau = \tau_0$;
 - 2 Grid resolution: Δx s.t. $k^q(x, x + \Delta x)(\sigma_\eta^q)^{-2} = 0.95$
while machine is running do
 - 3 $S_t \leftarrow \{x \in \mathcal{X} : u_t^q(x, \tau) \leq \kappa\}$;
 - 4 $L_t \leftarrow \{x \in S_t : \exists z \notin S_t, \text{ with } d(x, z) \leq \Delta x\}$;
 - 5 $W_t \leftarrow \{x \in L_t : u_t^q(x, \tau) - l_t^q(x, \tau) \geq \epsilon\}$;
 - 6 $x^* \leftarrow \text{PSO}(S_t, W_t)$;
 - 7 **if** $|f(x^\dagger(\tau)) - l_t^f(x^*, \tau)| \geq \epsilon_{\text{tol}}$ **then**
 - 8 **if** $u_t^q(x^*, \tau) \leq \kappa$ **then** evaluate $f(x^*)$, $q(x^*)$,
update τ ;
 - 9 **else**
 - 10 **while** $\exists x \in W_t$, s.t. $g_\epsilon^t(x, x^*) > 0$ **do**
 - 11 $x_w^* \leftarrow \arg \min_{x \in W_t} d(x^*, x)$ s.t. $g_\epsilon^t(x, x^*) \neq 0$;
 - 12 Evaluate $f(x_w^*)$, $q(x_w^*)$, update τ , S_t , L_t , W_t ;
 - 13 **else**
 - 14 set system to $x^\dagger(\tau)$, update τ
-

Task parameter. In general, the dynamics of a controlled system may vary due to external changes. For example, our rotational axis drive may be subject to different loads or the system components may wear due to extended use. As dynamics change, so do the optimal controllers. In this case, we must adapt to new regimes imposed by the environment. To this end, we extend GOOSE to the multi-task setting presented in Section IV by introducing a task parameter τ that captures the exogenous conditions that influence the system's dynamics, and by using the kernel introduced in Section IV.

Fig. 3: Comparison of GoOSE and CBO for 10 runs of the stationary control problem. On the left, we see the cost for each run and its mean and standard deviation. The center and right figures show the constraint values sampled by each method. GoOSE reaches the same solution as CBO (table), albeit more slowly (left). However, CBO heavily violates the constraints (center-right).

Algorithm 2: PSO

```

1 Input: Safe set  $S_t$ , Boundary  $W_t$ , # particles  $m$ ;
2  $p_i \sim \mathcal{U}(S_t)$ ,  $v_i \sim \pm \Delta x$ ,  $i = 1, \dots, m$ ;
3 for  $j \leftarrow 0$  to  $J$  do
4   for  $i \leftarrow 0$  to  $m$  do
5      $\text{cond} \leftarrow u_t^q(p_i^j, \tau) \leq \kappa \vee \exists x \in W_t : g_t^f(x, p_i^j) > 0$ ;
6      $\text{cond} \leftarrow \text{cond} \wedge l_t^f(p_i^j, \tau) < l_t^f(z_i^{j-1}, \tau)$ ;
7     if  $\text{cond}$  then  $z_i^j \leftarrow p_i^j$  else  $z_i^j \leftarrow z_i^{j-1}$ ;
8      $\bar{z}^j \leftarrow \arg \min_{i=1, \dots, m} l_t^f(z_i^j, \tau)$ ;
9      $r_{1,2} \sim \mathcal{U}([0, 2])$ ;
10     $v_i^{j+1} \leftarrow \alpha v_i^j + r_1(z_i^j - p_i^j) + r_2(\bar{z}^j - p_i^j)$ ,  $\forall i$ ;
11     $\tilde{p}_i^{j+1} \leftarrow p_i^j + v_i^j$ ,  $\forall i$ ;
12 return  $\bar{z}^J$ 

```

To guarantee safety, we assume that the initial safe seed S_0 contains at least one safe controller for each possible task.

Lipschitz constant. GOOSE uses the Lipschitz constant of the safety constraint L_q , which is not known in practice, to compute the pessimistic safe set and an optimistic upper bound on constraint values. For the pessimistic safe set, we adopt the solution suggested in [6] and use the upper bound of the confidence interval to compute it (see L. 3 of Algorithm 1). While pessimism is crucial for safety, optimism is necessary for exploration. To this end, GOOSE computes an optimistic upper bound for the constraint value at input z based on the lower bound of the confidence interval at another input x as $l_t^q(x, \tau) + L_q d(x, z)$, where $d(\cdot, \cdot)$ is the metric that defines the Lipschitz continuity of q . Here, we approximate this bound with $l_t^q(x, \tau) + \|\mu_t^\nabla(x, \tau)\|_\infty d(x, z)$, where $\mu_t^\nabla(x, \tau)$ is the mean of the posterior belief over the gradient of the constraint induced by our belief over the constraint which, due to properties of GPs, is also a GP. This is a local version of the approximation proposed in [33]. Based on this approximation, we want to determine whether, for the current task τ , an optimistic observation of the constraint at controller x , $l_t^q(x, \tau)$ would allow us to classify as safe a controller z despite an ϵ uncertainty due to noisy observations of the constraint. To this end, we introduce the optimistic noisy expansion operator

$$g_\epsilon^t(x, z) = \mathbb{I} [l_t^q(x, \tau) + \|\mu_t^\nabla(x, \tau)\|_\infty d(x, z) + \epsilon \leq \kappa],$$

where $\mathbb{I}$ is the indicator function and κ is the upper limit of the constraint. For a safe x , $g_\epsilon^t(x, z) > 0$ determines that: (i) z can plausibly be safe and (ii) evaluating the constraint at x could include z in the safe set.

Optimization and optimistic safe set. Normally, GOOSE explicitly computes the optimistic safe set and uses GP-

LCB [32] to determine the next input where to evaluate the objective, x^* . In other words, GOOSE solves $x_t^* = \arg \min_{x \in S_t^o} \mu_{t-1}^f(x) - \beta_{t-1}^f \sigma_{t-1}^f(x)$, where S_t^o is the optimistic safe set at iteration t . However, this requires a fine discretization of the domain $\mathcal{X}$ to represent S_t^o as finite set of points in $\mathcal{X}$, which does not scale to large domains. Moreover, the recursive computation of S_t^o is expensive and not well suited to the fast responses required by adaptive control. Similarly to [7], here we rely on particle swarm optimization (PSO) [34] to solve this optimization problem, which checks that the particles belong to the one-step optimistic safe set as the optimization progresses and avoids computing it explicitly. We initialize m particles positioned uniformly at random within the discretized pessimistic safe set with grid resolution Δx , velocity Δx with random sign (L. 2 of Algorithm 2) and fitness equal to the lower bound of the objective $l_t^f(\cdot)$. If a particle belongs to the optimistic safe set (L. 5) and its fitness improves (L. 6), we update its best position. This step lets the particle diffuse into the optimistic safe set without computing it explicitly. Subsequently, we update the particles' positions (L. 11) and velocities (L. 10) based on the particles' best position z_i^j and overall best position $\bar{z}^j$, which is updated in L. 8.

Algorithm. We present our variant of GOOSE for adaptive control. First, we compute the pessimistic safe set S_t (L. 3 of Algorithm 1) on a grid with lengthscale-dependent resolution Δx , its boundary L_t (L. 4) and the uncertain points on its boundary W_t (L. 5), which are used to determine whether controllers belong to the optimistic safe set. Based on these, a new suggestion x^* is computed (L. 6). If its lower bound is close to the best observation for the current task $x^\dagger(\tau) = \arg \min_{\{(x', \tau', f(x')) \in \mathcal{D} : \tau' = \tau\}} f(x')$, we stop (L. 7). Otherwise, if the suggestion is safe, we evaluate it and possibly update the task parameter (L. 8). Finally, if we are not sure it is safe, we evaluate all the expanders $x_{w'}^*$ in increasing order of distance from the suggestion x^* , until either x^* is in the pessimistic safe set and can be evaluated or there are no expanders for x^* and a new query to PSO is made (L. 11 and 12). During this, the task parameter is constantly updated.

Our adaptation of GOOSE invalidates the optimality and safety guarantees presented in [11]. However, these properties are retained in practice in our experimental evaluations.

VI. NUMERICAL RESULTS

We first apply Algorithm 1 to tune the controller in Fig. 2, simulating the system model in Section III in stationary conditions. Later, we use our method for adaptive control of instantaneous and slow-varying changes of the plant.

Fig. 4: Simulated adaptive control experiments with a sudden change of the moment of inertia m . The thin lines show the cost for each experiment. The thick one shows the best cost found for the current task. The tables show the mean values over 10 repetitions for these experiments. GOOSE quickly finds optimal solutions and is able to adapt to the disturbance.

Fig. 5: Figure and table equivalent to Fig. 4 for a simulated experiment with a slow change of the rotational damping b . GOOSE quickly finds optimal solutions and adapts to the drift.

The optimization ranges of the controller parameters are set to $K_p \in [5, 50]$, $K_v \in [0.1, 0.11]$ for the position and velocity gains, and $T_i \in [1, 10]$ for the time constant. For each task, GOOSE returns the controller corresponding to the best observation, $x^\dagger(\tau) = (K_p^\dagger, K_v^\dagger, T_i^\dagger)$. The cost $f(x)$ is provided in $[\text{deg } 10^{-3}]$ units. We denote the ‘true’ optimal controller computed via grid search as $x_{\text{grid}}^\dagger$. Even for this controller, the tracking error is larger than zero, as shown in table in Fig. 3. We use a zero mean prior and squared exponential kernel with automatic relevance determination, with length scales for each dimension $[l_{K_p}, l_{K_v}, l_{T_i}, l_m, l_{K_{ff}}, l_b] = [30, 0.03, 3, 0.5, 0.3, 5]$, identical for f , q_1 and q_2 for the numerical simulations and the experiments on the system. The likelihood variance is adjusted separately for each GP model.

Control for stationary conditions. In this section, we compare GOOSE to CBO (Constrained Bayesian Optimization) for controller tuning [18], [10] in terms of the

Fig. 6: Figures and table equivalent to those in Fig. 4 for the real-world adaptive control experiments with sudden change of the feed-forward gain K_{ff} . GOOSE quickly finds optimal controllers for all the regimes (plot) without violating the constraints (table) despite the location of the optimum keeps changing (table).

cost and the safety and performance constraints introduced in Section III, when tuning the plant under stationary conditions. We benchmark these methods against exhaustive grid-based evaluation using a grid with $5 \times 11 \times 10$ points.

For each algorithm, we run 10 independent experiments, which vary due to noise injected in the simulation. Each experiment starts from the safe controller $[K_p, K_v, T_i] = [15, 0.05, 3]$. The table in Fig. 3 shows the median number of iterations needed to minimize the cost. Fig. 3 (left) shows the convergence of both algorithms for each repetition. While both algorithms converge to the optimum, GOOSE prevents constraint violations for all iterations. In contrast, CBO violates the constraints in 27.8% of the iterations to convergence, reaching far beyond the safety limit of acceptable vibrations (Fig. 3, center), showing that additional safety-related measures are required. While the constraint violations incurred by CBO can be limited in stationary conditions by restricting the optimization range, this is not possible for adaptive control.

Instantaneous changes. We show how our method adapts to an instantaneous change in the load of the system. We modify the moment of inertia m and estimate this from the system data. We inform the algorithm of the operating conditions through a task parameter, $\tau_m = \log_{10}(\frac{1}{N} \sum_{t=1}^N |\text{FFT}(v - v_{ff})|)$, which is calculated from the velocity measurement v , using the feed-forward signal v_{ff} from position to velocity. This data-driven task parameter exploits the differences in the levels of noise in the velocity signal corresponding to different values of the moment of inertia m . As the velocity also depends on the controller parameters, configurations in a range of ± 0.15 of the current τ_m value are treated as the same task configuration. The algorithm is initialized with $[K_p, K_v, T_i] = [15, 0.05, 3]$ as safe seed for all tasks. The moment of inertia of the plant m is switched every 100 iterations. The table in Fig. 4 summarizes 10 repetitions of the experiment with a stopping criterion set to $\epsilon_{\text{tol}} = 0.002$.

Fig. 4 shows the sampled cost and the task parameters for one run, and the running minimum for each task. We

Task (τ_b)	2.72	4.53	2.73
It. to conv.	99	78	19
$f(x^\dagger(\tau_b))$	1.59	1.61	1.59
$(K_p^\dagger, K_v^\dagger, T_i^\dagger)$	(32, .09, 6)	(30, .09, 7)	(37, .1, 7)
Violations (q_1, q_2)	(0, 0)	(0, 0)	(0, 0)

Fig. 7: Figures and table equivalent to those in Fig. 4 for the real-world adaptive control experiments with sudden change of the rotational damping $\tilde{b}$. GOOSE quickly finds optimal controllers for all the regimes (plot) without violating the constraints (table) despite the location of the optimum keeps changing (table).

clearly see that GOOSE quickly finds a high performing controller for the initial operating conditions ($m = 0.0191\text{kgm}^2$). Subsequently, it adapts K_p and K_v in few iterations to the new regime ($m = 0.0382\text{kgm}^2$) due to the capability of the GP model to generalize across tasks (Fig. 4, table). Finally, when the system goes back to the initial regime, GOOSE immediately finds a high-performing controller and manages to marginally improve over it. In particular, the best cost $f(x^\dagger(\tau_m))$ found for each condition coincides with the optimum obtained by grid evaluation $f(x_{\text{grid}}^\dagger(\tau_m))$. Moreover the constraints are never violated (Fig. 4, table).

Gradual changes. We show how our method adapts to slow changes in the dynamics. In particular, we let the rotation damping coefficient increase linearly with time: $b(t) = b_0(1 + \frac{t}{1000})$, where $b_0 = 30.08\text{kgm}^2/\text{s}$. In this case, time is the task parameter and, therefore, we use the temporal kernel $k(t, t') = (1 - \epsilon_t)^{|t-t'|/2}$ [35] as task kernel with $\epsilon_t = 0.0001$. This kernel increases the uncertainty in already evaluated samples with time. To incorporate the drift of the dynamics, we evaluate the best sample $x^\dagger$ in Algorithm 1 with respect to the stopping criterion threshold $\epsilon_{\text{tol}} = 0.001$ in a moving window of the last 30 iterations. Learning is slower, due to the increased uncertainty of old data points. On average, the stopping criterion (L. 7 in Algorithm 1) is fulfilled in 74 out of 300 iterations. The optimum increases, (Fig. 5), corresponding to a change in the system dynamics. In the second half of the experiment the cost decreases, and reaches optimum more often, as the GPs successfully learn the trend of the parameter $b(t)$. There were no constraint violations in any of the repetitions, see table in Fig. 5.

VII. EXPERIMENTAL RESULTS

We now demonstrate the proposed adaptive control algorithm on a rotational drive system. The system has an encoder resolution of $\Delta p = 1\text{deg} \times 10^{-7}$ for the angular position, $\Delta v = 0.0004\text{RPM}$ for the angular velocity and $\Delta T = 0.0008\text{Nm}$ for the torque. The angular position of the

system has no hardware limit. The limits of the angular velocity and the torque are $v_{\text{lim}} = 50\text{RPM}$ and $T_{\text{lim}} = 3.48\text{Nm}$, respectively. First, we show how the controller parameters adapt when the algorithm is explicitly informed about a change in the feed-forward gain K_{ff} . We then demonstrate the performance when an external change occurs, corresponding to change in the rotational resistance, which can be estimated from the system's data. The optimization ranges and the kernels hyperparameters are the same as in Section VI. The likelihood variance is adjusted separately for each GP model.

Internal parameter change. We start an experiment with the nominal feed-forward gain, $K_{\text{ff}} = 1$, and switch it subsequently four times between 0.9 and 1.1 in intervals of 50 iterations. For each value of K_{ff} , the starting point is a safe sample, collected with $[K_p, K_v, T_i] = [15, 0.05, 3]$. The value of K_{ff} is used as task parameter $\tau_{K_{\text{ff}}}$. The stopping criterion is set to $\epsilon_{\text{tol}} = 0.001$. The convergence of the optimization accelerates with increasing data. Fig. 6 shows that convergence is not reached for the nominal K_{ff} during the first 50 iterations, whereas the last configuration with $\tau_{K_{\text{ff}}} = 1.1$ requires only 16 until convergence, showing that learning is efficient, even if the optimum shifts w.r.t. the configuration. Constraint violations are prevented for all task parameters (table in Fig. 6).

External parameter change. We validate experimentally the change introduced to the controller by modifying the friction $\tilde{b}$ in the system, which is related to the non-linearity in the dynamics of the plant. We estimate the change in $\tilde{b}$ by the average torque measurement and provide it as task parameter $\tau_{\tilde{b}}$, as shown in Fig. 7. We start with optimizing the nominal controller parameters for 100 iterations, followed by an increase of $\tilde{b}$ (and of $\tau_{\tilde{b}}$ accordingly), which is achieved by wrapping elastic bands around the rotational axis and fixing them at the frame of the system. In the last 100 iterations we switch back to the nominal condition. The stopping criterion is set to $\epsilon_{\text{tol}} = 0.001$. Since $\tau_{\tilde{b}}$ is not fixed and is influenced by noise, all configurations with $\tau_{\tilde{b}}$ in a range of ± 0.1 of the current $\tau_{\tilde{b}}$ value are treated as the same task configuration. The cost increases after the first intervention, then reaches close to nominal values in 10 iterations, and adapts quickly at the next $\tau_{\tilde{b}}$ switch (Fig. 7). The constraints are never violated.

VIII. CONCLUDING REMARKS

We present a model-free approach to safe adaptive control. To this end, we introduce several modifications to GOOSE, a safe Bayesian optimization method, to enable its practical use on a rotational motion system. We demonstrate numerically and experimentally that our approach is sample efficient, safe, and achieves the optimal performance for different types of disturbances encountered in practice. However, scaling to higher dimensional controllers is an open problem, which we hope to address in future research.

REFERENCES

- [1] G. Chowdhary, H. A. Kingravi, J. P. How, and P. A. Vela, "Bayesian nonparametric adaptive control using Gaussian processes," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 3, pp. 537–550, 2015.

- [2] R. C. Grande, G. Chowdhary, and J. P. How, "Experimental validation of Bayesian nonparametric adaptive control using Gaussian processes," *Journal of Aerospace Information Systems*, vol. 11, no. 9, pp. 565–578, 2014.
- [3] D. Sbarbaro and R. Murray-Smith, *Self-tuning Control of Non-linear Systems Using Gaussian Process Prior Models*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 140–157.
- [4] A. Gahlawat, P. Zhao, A. Patterson, N. Hovakimyan, and E. Theodorou, "L1-gp: L1 adaptive control with Bayesian learning," in *Learning for Dynamics and Control*, ser. Proceedings of Machine Learning Research, vol. 120. The Cloud: PMLR, 10–11 Jun 2020, pp. 826–837.
- [5] D. D. Fan, J. Nguyen, R. Thakker, N. Alatur, A.-a. Agha-mohammadi, and E. A. Theodorou, "Bayesian learning-based adaptive control for safety critical systems," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4093–4099.
- [6] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 491–496.
- [7] R. 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] M. Khosravi, V. Behrunani, R. S. Smith, A. Rupenyan, and J. Lygeros, "Cascade control: Data-driven tuning approach based on bayesian optimization," *IFAC World Congress*, 2020, <https://arxiv.org/abs/2005.03970>.
- [9] 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.
- [10] C. König, M. Khosravi, M. Maier, R. S. Smith, A. Rupenyan, and J. Lygeros, "Safety-Aware Cascade Controller Tuning Using Constrained Bayesian Optimization," *arXiv e-prints*, p. arXiv:2010.15211, Oct. 2020.
- [11] 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.
- [12] J. Mockus, *Bayesian Approach to Global Optimization: Theory and Applications*, ser. Mathematics and its Applications. Springer Netherlands, 2012.
- [13] 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.
- [14] R. Calandra, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian optimization for learning gaits under uncertainty," *Annals of Mathematics and Artificial Intelligence*, vol. 76, no. 1, pp. 5–23, 2016.
- [15] A. Marco, F. Berkenkamp, P. Hennig, A. P. Schoellig, A. Krause, S. Schaal, and S. Trimpe, "Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with Bayesian optimization," *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1557–1563, 2017.
- [16] R. Antonova, A. Rai, T. Li, and D. Kragic, "Bayesian optimization in variational latent spaces with dynamic compression," in *Conference on Robot Learning (CoRL)*. PMLR, 2020, pp. 456–465.
- [17] M. Turchetta, A. Krause, and S. Trimpe, "Robust model-free reinforcement learning with multi-objective Bayesian optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [18] 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.
- [19] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. Adams, and Z. Ghahramani, "Predictive entropy search for Bayesian optimization with unknown constraints," in *Proceedings of International Conference on Machine Learning (ICML)*, 2015.
- [20] V. Picheny, R. B. Gramacy, S. Wild, and S. Le Digabel, "Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 1435–1443.
- [21] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, and S. M. Wild, "Modeling an augmented Lagrangian for blackbox constrained optimization," *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016.
- [22] S. Ariafar, J. Coll-Font, D. H. Brooks, and J. G. Dy, "ADMMBO: Bayesian optimization with unknown constraints using ADMM," *Journal of Machine Learning Research (JMLR)*, vol. 20, no. 123, pp. 1–26, 2019.
- [23] 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.
- [24] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics," *arXiv preprint arXiv:1602.04450*, 2016.
- [25] 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.
- [26] Y. Sui, vincent Zhuang, J. Burdick, and Y. Yue, "Stagewise safe Bayesian optimization with Gaussian processes," in *Proceedings of International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 10–15 Jul 2018, pp. 4781–4789.
- [27] F. Villegas, R. Hecker, M. Peña, D. Vicente, and G. Flores, "Modeling of a linear motor feed drive including pre-rolling friction and aperiodic cogging and ripple," *The International Journal of Advanced Manufacturing Technology*, vol. 73, 07 2014.
- [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] K. Swersky, J. Snoek, and R. Adams, "Multi-task Bayesian optimization," *Advances in Neural Information Processing Systems (NeurIPS)*, 01 2013.
- [30] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation and Machine Learning series, 2018.
- [31] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *Proceedings of International Conference on Machine Learning (ICML)*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 06–11 Aug 2017, pp. 844–853.
- [32] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," in *Proceedings of International Conference on Machine Learning (ICML)*, vol. 58, no. 5. Institute of Electrical and Electronics Engineers (IEEE), May 2010, p. 3250–3265.
- [33] J. González, Z. Dai, P. Hennig, and N. Lawrence, "Batch Bayesian optimization via local penalization," in *Artificial intelligence and statistics*, 2016, pp. 648–657.
- [34] S. Kiranyaz, T. Ince, and M. Gabbouj, *Multidimensional particle swarm optimization for machine learning and pattern recognition*. Springer, 2014.
- [35] I. Bogunovic, J. Scarlett, and V. Cevher, "Time-varying Gaussian process bandit optimization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 314–323. [Online]. Available: <http://proceedings.mlr.press/v51/bogunovic16.html>