Event-Triggered Safe Bayesian Optimization on Quadcopters

Antonia Holzapfel*   ANTONIA.HOLZAPFEL@DSME.RWTH-AACHEN.DE
Paul Brunzema*   PAUL.BRUNZEMA@DSME.RWTH-AACHEN.DE
Sebastian Trimpe   TRIMPE@DSME.RWTH-AACHEN.DE
Institute for Data Science in Mechanical Engineering, RWTH Aachen University, Germany

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

Bayesian optimization (BO) has proven to be a powerful tool for automatically tuning control parameters without requiring knowledge of the underlying system dynamics. Safe BO methods, in addition, guarantee safety during the optimization process, assuming that the underlying objective function does not change. However, in real-world scenarios, time-variations frequently occur, for example, due to wear in the system or changes in operation. Utilizing standard safe BO strategies that do not address time-variations can result in failure as previous safe decisions may become unsafe over time, which we demonstrate herein. To address this, we introduce a new algorithm, Event-Triggered SafeOpt (ETSO), which adapts to changes online solely relying on the observed costs. At its core, ETSO uses an event trigger to detect significant deviations between observations and the current surrogate of the objective function. When such change is detected, the algorithm reverts to a safe backup controller, and exploration is restarted. In this way, safety is recovered and maintained across changes. We evaluate ETSO on quadcopter controller tuning, both in simulation and hardware experiments. ETSO outperforms state-of-the-art safe BO, achieving superior control performance over time while maintaining safety.

Keywords: controller tuning, Bayesian optimization, safe learning, quadcopter, adaptive optimization, time-varying systems

1. Introduction

Bayesian optimization (BO) has emerged as a powerful tool to tune controllers without requiring an exact model of the physical system, e.g., in robotics (Calandra et al., 2016; Marco et al., 2016) or automotive applications (Neumann-Brosig et al., 2018). This is achieved by optimizing a black-box objective function that quantifies the closed-loop performance as a function of controller parameters. This (unknown) function is often modeled as a Gaussian process (GP), which is then used to formulate an acquisition function guiding the optimization process. Specifically, the acquisition function determines which control parameters to choose next trading off exploration and exploitation.

Unbounded exploration can lead to control parameters that are unsafe and can harm the hardware of the physical system or people operating it. Therefore, *safe* BO algorithms have been proposed (Sui et al., 2015; Berkenkamp et al., 2016) and their efficacy has been demonstrated on various control systems, including quadcopters (Berkenkamp et al., 2021). Starting from an initial safe controller, these algorithms build up a set of control parameters that are assumed to be safe. Safe BO considers no variations in the controlled system and thus, no variation in the performance function between experiments. However, the closed-loop performance may change over time, e.g., due to

---

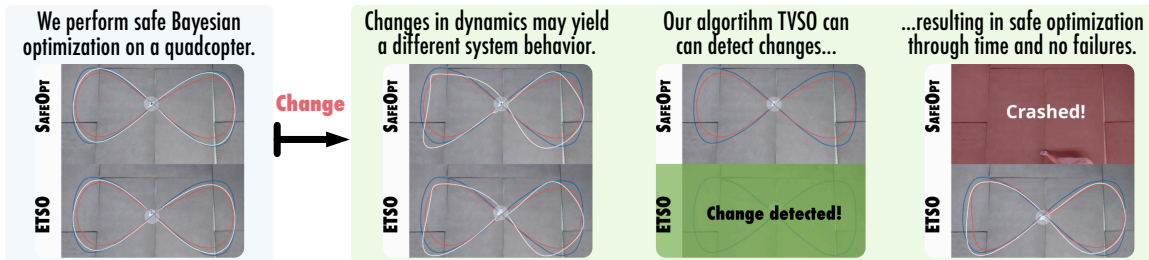* The authors contributed equally to this work.

Figure 1: We introduce ETSO, an algorithm for safe black-box optimization in time-varying environments. ETSO maintains safety across time variations in contrast to state-of-the-art algorithms such as SafeOpt. Images are from the video: https://youtu.be/nLmeO-fMIvg.

changes in the system dynamics or in the reference. As a consequence, the underlying model can become invalid, and once found optimal controllers may become sub-optimal. In addition, such changes may shift the safe set, and parameters that used to be safe may become unsafe. Hence, adaptation to these changes is vital to ensure both: good performance and safety over time.

Time-varying BO has been proposed to address change over time (Bogunovic et al., 2016; Brunzema et al., 2023). However, existing methods focus on performance, yet do not consider safe exploration. In general, aiming for a-priori safety guarantees under *arbitrary* changes is an ill-posed problem, and some assumptions are needed. In this work, we consider systems that remain in a fixed mode for a minimum of time. Such lower bounds on the time between changes are common in switched systems literature, where they are referred to as *dwell time* (Hespanha and Morse, 1999; Baumann et al., 2019). Scenarios that display this behavior are, e.g., those that involve abrupt, but infrequent changes. In this paper, we propose to combine existing mechanisms for safe exploration with the recently proposed event-triggered BO approach (Brunzema et al., 2023) to deal with changes. Event-triggered BO detects change by comparing the expected performance with obtained measurements, without requiring an explicit model of the time variations. We thus propose Event-Triggered SafeOpt (ETSO) as a new type of safe BO algorithm that can accommodate for time variations. Our algorithm learns for a fixed number of rounds until it obtains a suitable controller; a practical approach for various applications. During the optimization and afterwards, the event trigger monitors the learned function for changes. Once such changes are detected, ETSO reverts to a safe backup controller, resets the safe set, and restarts the exploration. Unlike previous contextual BO approaches (König et al., 2021), ETSO does not require any contextual measurements that quantify the temporal change, but only the noisy evaluations of the performance function.

We test ETSO for controller tuning in simulation and on hardware. Using a Bitcraze Crazyflie 2.1 quadcopter and experiments emulating typical variations, we show (cf. Figure 1): *(i)* standard safe BO can indeed fail due to time-variations, and *(ii)* ETSO remains safe across time variations in these situations. In summary, our main contributions are:

- **ETSO:** Combining safe BO and event triggering, we introduce the new algorithm ETSO that maintains safety across time variations by reverting to a backup controller and resetting its safe set when needed.
- **Empirical evaluation in simulation and hardware:** ETSO shows superior safety and performance compared to baselines like SafeOpt, which can fail on time variations.

2

2. Problem Setting

We aim to find optimal control parameters $\boldsymbol{\theta}_t^* \in \Theta \subseteq \mathbb{R}^d$ over a time horizon T in time-varying environments.[1] The performance of a controller is quantified by the performance function $J_{\xi_t} : \Theta \to \mathbb{R}$ where the functional dependency of J_{ξ_t} on $\boldsymbol{\theta}_t$ is unknown. Here, we denote ξ_t as the system mode at time t indicating that some variables of the closed loop system that influence the performance are time-varying. We consider the setting where it is unknown *which* variables can change, but the system has a minimum dwell time τ_{dwell} and the changes occur after some time steps since the last change $t' = t - \tau > \tau_{dwell}$, where τ is the time step of the change. To avoid system failure, we further require the optimization to be safe. We therefore aim to ensure a minimum safety threshold $J_{\xi_t}(\boldsymbol{\theta}_t) \geq J_{\min,t}$ in probability where $J_{\min,t} \in \mathbb{R}$ for all time steps $t \in \mathcal{T}_T := \{1, 2, \ldots, T\}$ as it is standard in SafeOpt-type algorithms (e.g., Sui et al. (2015)). Likewise, we assume that $J_{\xi_t}(\boldsymbol{\theta}_t)$ is L-Lipschitz continuous. Note that in our setting this bound is also time-varying as it may depend on ξ_t. For safety, we assume access to a backup controller. These are often supplied by the manufacturer or obtained by some robust controller design; they are usually robust, but sub-optimal.

Assumption 1 *We have access to a safe backup controller $\boldsymbol{\theta}_B$ that stabilizes the close loop system for all system modes ξ_t. A neighborhood of $\boldsymbol{\theta}_B$ is also safe such that $J_{\xi_t}(\boldsymbol{\theta}_{\mathrm{B}}) - \epsilon \geq J_{\mathrm{crit}}$.*

In Assumption 1, $J_{\mathrm{crit}} \leq J_{\min,t}$ is the critical performance before system failure and $\epsilon \in \mathbb{R}_{>0}$. Remaining safe implies that we may not be able to find the global optimum, but only the optimum within the reachable set $\mathcal{R}_t := \{\boldsymbol{\theta} \in \Theta \mid J_{\xi_t}(\boldsymbol{\theta}) \geq J_{\min,t}$ and $\boldsymbol{\theta}$ path-connected with $\boldsymbol{\theta}_B\}$. Here we use $\boldsymbol{\theta}_{\mathrm{B}}$ from Assumption 1 as an initial controller. The reachable set may change depending on ξ_t; hence $\mathcal{R}_t$ is time-varying (see Figure 2). With $\mathcal{R}_t$, we can formalize our optimization problem as

$$\boldsymbol{\theta}_t^* = \arg\max_{\boldsymbol{\theta} \in \mathcal{R}_t} J_{\xi_t}(\boldsymbol{\theta}). \tag{1}$$

At each time step t, an algorithm can query this performance function only once. In the controller tuning context, this means performing one experiment e.g., flying one round with a quadcopter (see Figure 1). The algorithm then receives an observation which is perturbed by independent and identically distributed (i.i.d.) zero mean Gaussian noise with σ_n^2 as the noise variance as

$$\hat{J}_t = J_{\xi_t}(\boldsymbol{\theta}_t) + w_t, \quad w_t \sim \mathcal{N}(0, \sigma_n^2). \tag{2}$$

Problem Statement. We aim to develop a practical algorithm to tune controllers in time-varying environments by optimizing (1). The optimization should be performed without safety violations. Contrary to related work in safe BO, we consider time variations in the underlying function, thus a changing reachable set $\mathcal{R}_t$, and no explicit measurements of the changing parameters.

3. Related Work

Our proposed algorithm ETSO builds on ideas from time-varying BO as well as safe BO to ensure safe learning over time. In the following, we discuss the related work to ETSO in more detail.

Safe Bayesian Optimization. Safe BO methods aim to learn optimal parameters while also ensuring safety. Most proposed algorithms follow the ideas of SafeOpt (Sui et al., 2015). Some popular examples of SafeOpt variants include modified SafeOpt (Berkenkamp et al., 2016), multiple-constraint SafeOpt (Berkenkamp et al., 2021), GoSafe (Baumann et al., 2021; Sukhija et al., 2023),

---

1. Throughout this paper, we highlight time dependency using a subscript $t \in \mathbb{N}$ to denote discrete time steps.

and StageOpt (Sui et al., 2018). These variants have been used in various control applications, such as quadcopters (Berkenkamp et al., 2021) or heat pumps (Khosravi et al., 2019). An alternative method, GoOSE (Turchetta et al., 2019), can transform any algorithms to be safe by incorporating similar safety measures as SafeOpt. However, all of these methods assume that the performance function is time-invariant. This means that safe decisions are assumed to remain safe over time. This does not hold in dynamic environments, where previously safe decisions may become unsafe. We build on the modified SafeOpt method and extend it to time-varying environments. It is worth noting that other approaches could be used; we predominantly choose modified SafeOpt due to computational efficiency, which is necessary due to the limited battery life of the quadcopters.

Adapting to Time-Variations. There are several approaches to time-varying BO (TVBO), which consider trade-offs between remembering and forgetting data. These approaches either use a Markov-chain model (Bogunovic et al., 2016; Brunzema et al., 2022, 2023) or a variation budget (Zhou and Shroff, 2021; Deng et al., 2022) to model time variations. Since we aim to be adaptive to changes, we build on the algorithm proposed in Brunzema et al. (2023) (cf. Section 4). It leverages ideas from event-triggered learning (Solowjow and Trimpe, 2020; Umlauft and Hirche, 2019), i.e., detecting changes online and re-learning only when necessary, and transfers them to TVBO. Unlike all proposed TVBO approaches, we additionally consider safety and obtain an algorithm that learns safely under time variations such that it can be applied to hardware systems.

Contextual Safe Bayesian Optimization. In contextual BO (Krause and Ong, 2011), the model of the underlying function includes a context parameter. This parameter represents environmental conditions that cannot be influenced during optimization and may vary over time. Safe contextual BO methods leverage this context parameter to account for time variations in the system and enable safe exploration using BO. Such methods have been successfully applied to various control problems (Su et al., 2018; Fiducioso et al., 2019; De Blasi et al., 2023). Other methods, such as GoOSE for adaptive control (König et al., 2021) and VACBO (Xu et al., 2023), are variants of this approach. Given our problem setting in Section 2, safe contextual BO methods cannot be applied as we assume to only have access to noisy observations and no additional environmental condition measurements. To circumvent this lack of additional information to quantify change, our approach utilizes an event-based trigger to detect changes based on the expected and observed performance.

4. Background

Our method combines safe Bayesian optimization with the adaptation to time variations using an event trigger. Next, we will introduce the necessary background and notation for both concepts.

Gaussian Processes. GPs (Williams and Rasmussen, 2006) are a probabilistic non-parametric method for regression that provide explicit uncertainty estimates for the learned function. This makes them a powerful tool for regression and also as the probabilistic surrogate model for Bayesian optimization. A GP is fully defined by its mean function $m : \Theta \to \mathbb{R}$ and kernel $k : \Theta \times \Theta \to \mathbb{R}$ and we denote it as $f(\boldsymbol{\theta}) \sim \mathcal{GP}(m, k)$. The prior belief defined through the mean function and kernel can be updated using a data set $\mathcal{D}_t := \{(\boldsymbol{\theta}_i, \hat{J}_i)\}_{i=1}^{t-1}$ to obtain a posterior prediction over a test point $\boldsymbol{\theta}$. Assuming i.i.d. additive Gaussian noise in the observations as in (2), the posterior mean and covariance are $\mu_{\mathcal{D}_t}(\boldsymbol{\theta}) = m(\boldsymbol{\theta}) + \boldsymbol{k}_t^T(\boldsymbol{\theta})(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}(\hat{\boldsymbol{J}} - m(\boldsymbol{\theta}))$ and $\sigma_{\mathcal{D}_t}^2(\boldsymbol{\theta}) = k(\boldsymbol{\theta}, \boldsymbol{\theta}) - \boldsymbol{k}_t^T(\boldsymbol{\theta})(\mathbf{K}_t + \sigma_n^2\mathbf{I}_t)^{-1}\boldsymbol{k}_t(\boldsymbol{\theta})$, respectively, where $\mathbf{K}_t = [k(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)]_{i,j=1}^{t-1}$ is the Gram matrix, $\boldsymbol{k}_t(\boldsymbol{\theta}) = [k(\boldsymbol{\theta}_i, \boldsymbol{\theta})]_{i=1}^{t-1}$, and noisy measurements are concatenated as $\hat{\boldsymbol{J}} = [\hat{J}_1, \ldots, \hat{J}_{t-1}]^\top$.

Safe Bayesian Optimization. BO (Garnett, 2023) is an optimization method for noisy black-box functions. The core components of BO are a surrogate model such as a GP and an acquisition function that balances exploration and exploitation. To *safely* optimize J, i.e., without violating some minimal performance $J_{\min}$ with high probability, SafeOpt-type algorithms impose regularity assumptions on J: first, that J is L-Lipschitz continuous, and second, that $J \in \mathcal{H}_k$, where $(\mathcal{H}_k, \| \cdot \|_k)$ is the unique reproducing kernel Hilbert space with the reproducing kernel k. Specifically, SafeOpt-type algorithms require that there exists a bound $R \in \mathbb{R}_{\geq 0}$ such that $\|J\|_k \leq R$. With this, one can leverage frequentist uncertainty bounds on the GP regression error which state that $|J(\boldsymbol{\theta}) - \mu_{\mathcal{D}_t}(\boldsymbol{\theta})| \leq \beta_t \sigma_{\mathcal{D}_t}(\boldsymbol{\theta})$ for all $\theta \in \Theta$ and all time steps holds true with probability at least $1 - \delta$ (see e.g., Srinivas et al. (2009, Theorem 6)). In our method, we use the modified SafeOpt algorithm (Berkenkamp et al., 2016) which directly computes the safe set based on the upper and lower confidence bounds based on such uncertainty bounds as $u_t(\boldsymbol{\theta}) := \mu_{\mathcal{D}_t}(\boldsymbol{\theta}) + \beta_t \sigma_{\mathcal{D}_t}(\boldsymbol{\theta})$ and $\ell_t(\boldsymbol{\theta}) := \mu_{\mathcal{D}_t}(\boldsymbol{\theta}) - \beta_t \sigma_{\mathcal{D}_t}(\boldsymbol{\theta})$, respectively. With this, the safe set, the maximizers, and the expanders, of the modified SafeOpt algorithm are $\mathcal{S}_t := \{\boldsymbol{\theta} \in \Theta \mid \ell_t(\boldsymbol{\theta}) \geq J_{\min}\}$, $\mathcal{M}_t := \{\boldsymbol{\theta} \in \mathcal{S}_t \mid u_t(\boldsymbol{\theta}) \geq \max_{\boldsymbol{\theta}' \in \mathcal{S}_t} \ell_t(\boldsymbol{\theta}')\}$ and $\mathcal{G}_t := \{\boldsymbol{\theta} \in \mathcal{S}_t \mid g_t(\boldsymbol{\theta}) > 0\}$, with $g_t(\boldsymbol{\theta}) := | \{\boldsymbol{\theta}' \in \Theta \setminus \mathcal{S}_t \mid \ell_{t,(\boldsymbol{\theta}, u_t(\boldsymbol{\theta}))}(\boldsymbol{\theta}') \geq J_{\min}\} |$, respectively. The acquisition function is then defined as $\boldsymbol{\theta}_t = \arg\max_{\boldsymbol{\theta} \in \mathcal{G}_t \cup \mathcal{M}_t} u_t(\boldsymbol{\theta}) - \ell_t(\boldsymbol{\theta})$. SafeOpt-type algorithms require an initial safe controller as in Assumption 1.

Event-Triggered Time-Varying Bayesian optimization. The concept of detecting and adapting to time-variations in BO using an event trigger was introduced by Brunzema et al. (2023). It allows their algorithm ET-GP-UCB to account for time variations without explicitly modelling and estimating a rate of change. The event trigger is defined as follows.

Definition 1 (Event-triggered TVBO framework) *Given a test function ψ_t and a threshold function κ_t, both of which can depend on the current dataset $\mathcal{D}_t$ and the latest query location and measurement pair $(\boldsymbol{\theta}_t, \hat{J}_t)$, the event trigger at time step t is defined as*

$$\gamma_{\text{reset}} = 1 \Leftrightarrow \psi_t \left(\mathcal{D}_t, (\boldsymbol{\theta}_t, \hat{J}_t) \right) > \kappa_t \left(\mathcal{D}_t, (\boldsymbol{\theta}_t, \hat{J}_t) \right) \tag{3}$$

where γ_{reset} is the binary indicator for whether to reset the dataset ($\gamma_{\text{reset}} = 1$) or not ($\gamma_{\text{reset}} = 0$).

Brunzema et al. (2023) define their test and threshold function such that the event trigger activates when observations deviate significantly from the prediction leveraging the following bound on deviations between measurements and the prediction where t' is the time step since the last reset.

Lemma 2 (Brunzema et al. (2023, Lemma 3)) *Let $J \sim \mathcal{GP}(0, k)$. Pick $\delta_B \in (0, 1)$ and set $\rho_{t'} = 2 \ln \frac{2\pi_{t'}}{\delta_B}$, where $\sum_{t' \geq 1} \pi_{t'}^{-1} = 1$, $\pi_{t'} > 0$. Also set $\bar{w}_{t'}^2 = 2\sigma_n^2 \ln \frac{2\pi_{t'}}{\delta_B}$. Then, observations $\hat{J}_t$ satisfy $|\hat{J}_t - \mu_{\mathcal{D}_t}(\boldsymbol{\theta}_t)| \leq \rho_{t'}^{1/2} \sigma_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \bar{w}_{t'}$ for all time steps $t' \geq 1$ with prob. at least $1 - \delta_B$.*

With this Lemma for non-changing functions J, they set the test function and threshold function to

$$\psi_t \left(\mathcal{D}_t, (\boldsymbol{\theta}_t, \hat{J}_t) \right) = \left| \hat{J}_t - \mu_{\mathcal{D}_t}(\boldsymbol{\theta}_t) \right| \quad (4) \quad \text{and} \quad \kappa_t \left(\mathcal{D}_t, (\boldsymbol{\theta}_t, \hat{J}_t) \right) = \sqrt{\rho_{t'}} \sigma_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + \bar{w}_{t'}. \quad (5)$$

5. ETSO – Combining Safety and Adaptation to Changing Environments

In this section, we present our algorithm Event-Triggered SafeOpt (ETSO) for safe BO in time-varying environments. Specifically, we aim to solve (1) for a given task by combining the concepts
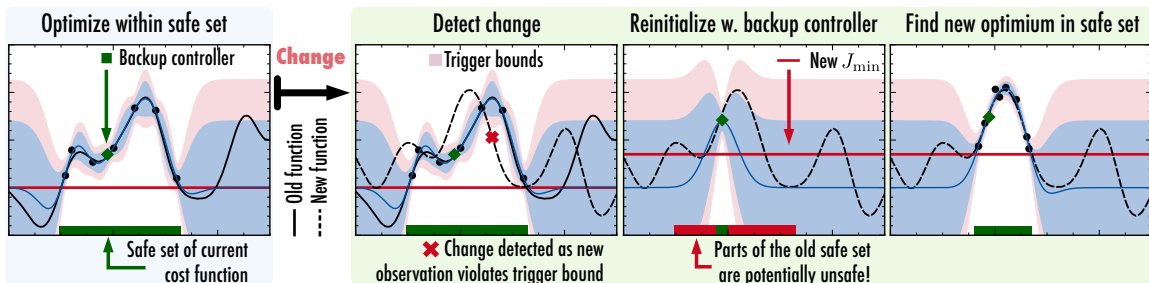
Figure 2: Overview of our proposed algorithm. ETSO starts by optimizing a the performance function with SafeOpt. Once our event trigger detects a significant change, we reset to the safe backup controller, calculate a new threshold $J_{\min,t}$, and restart the exploration.

of SafeOpt-type algorithms and adapting to changes by detecting them based on the expected performance with an event trigger that can reset the data and thus, the safe set. Figure 2 gives an overview of ETSO and it is summarized in Algorithm 1. Initially, ETSO performs safe BO for safe exploration for $T_L \leq T$ rounds while being monitored by the event trigger. This way, changes can already be detected during optimization. As discussed, we choose the modified SafeOpt algorithm (Berkenkamp et al., 2016) for this step, as well as an event trigger as in Definition 1 with (4) and (5). After T_L rounds, ETSO chooses the parameters that maximize the posterior mean within the safe set. If at any point the trigger is activated, ETSO resets the dataset, and explores the new system mode ξ_t with SafeOpt. For this, the first round is performed using $\boldsymbol{\theta}_{\mathrm{B}}$ and a new limit to the cost $J_{\min,t}$ with respect to it, as will be described in the implementation details section.

To prove safety for ETSO for different modes ξ_t, we require assumptions on the performance of our event trigger as well as on the performance of the controller for the iteration in the new modes.

Assumption 2 *When there is a change relevant for the performance, it is detected. Let the change happen at time τ. The parameters $\boldsymbol{\theta}_\tau$ are at least critically stable in the new system mode ξ_τ.*

The first part of the assumption is satisfied if the change is significant enough w.r.t. the threshold function of our event trigger. This threshold can be large during optimization given our acquisition function that selects control parameters with the highest variance in the safe set. It is substantially smaller after T_L rounds, when ETSO selects its best predicted control parameters. We will demonstrate in our experiments that not all changes can nor should be detected, as some are insignificant. The second part of the assumption is a natural extension of Assumption 1 and acts as a regularity assumption on the temporal change. With the two assumptions, the following is immediate:

Corollary 3 *Given Assumption 1 and Assumption 2, ETSO remains safe across time variations with probability at least $1 - \delta$.*

Proof We optimize safely with high probability using SafeOpt as shown in Sui et al. (2015, Theorem 1). With Assumption 2, once a change occurs, we detect it and have no system failure. ETSO resets the safe set and reverts to $\boldsymbol{\theta}_{\mathrm{B}}$, which is safe for the new system mode by Assumption 1. The following optimization is again safe by the above arguments proving the claim. ∎

Algorithm 1: Event-Triggered SafeOpt (ETSO).

Input: GP prior $(\mu_{\mathcal{D}_0}, k)$, backup controller $\boldsymbol{\theta}_{\mathrm{B}}$, $\delta_B \in (0,1)$, β, max. learn rounds $T_L > 1$

1   Initialize GP with $\mathcal{D}_1 = (\boldsymbol{\theta}_{\mathrm{B}}, \hat{J}_{\mathrm{B}})$; Set $J_{\min,t} \leftarrow J_{\min}(\hat{J}_{\mathrm{B}})$; Set $t' \leftarrow 1$

2   **for** $t \leftarrow 1$ **to** T **do**

3   Update GP with $\mathcal{D}_t$

4   **if** $t' \leq T_L$ **then**

5   Update $\mathcal{S}_t, \mathcal{M}_t, \mathcal{G}_t$ according to mod. SafeOpt (Berkenkamp et al., 2016)

6   **if** $t' < T_L$ **then**

7         $\boldsymbol{\theta}_t \leftarrow \arg\max_{\boldsymbol{\theta} \in \mathcal{G}_t \cup \mathcal{M}_t}(u_t(\boldsymbol{\theta}) - l_t(\boldsymbol{\theta}))$

8   **else**

9         $\boldsymbol{\theta}_t \leftarrow \arg\max_{\boldsymbol{\theta} \in \mathcal{S}_t}(\mu_t(\boldsymbol{\theta}))$

10      $\hat{J}_t \leftarrow \text{OBSERVATION}(\boldsymbol{\theta}_t)$

11      $\mathcal{D}_{t+1}, t' \leftarrow \textbf{Event\_Trigger}(\mathcal{D}_t, (\hat{J}_t, \boldsymbol{\theta}_t), \delta_B)$

12   **end**

13   **Event_Trigger**$(\mathcal{D}_t, (\hat{J}_t, \boldsymbol{\theta}_t), \delta_B)$:

14      $\gamma_{reset} \leftarrow$ evaluate (**??**) with (4) and (7)

15   **if** γ_{reset} **then**

        /* reset to backup controller */

16         $\hat{J}_{\mathrm{B}} \leftarrow \text{OBSERVATION}(\boldsymbol{\theta}_{\mathrm{B}})$

17         $\mathcal{D}_{t+1} = \{(\boldsymbol{\theta}_{\mathrm{B}}, \hat{J}_{\mathrm{B}}), (\boldsymbol{\theta}_t, \hat{J}_t)\}$

18         $J_{\min,t} \leftarrow J_{\min}(\hat{J}_{\mathrm{B}})$

19         $t' \leftarrow 2$

20   **else**

21         $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup (\boldsymbol{\theta}_t, \hat{J}_t)$

22         $t' \leftarrow t' + 1$

23   **return** $\mathcal{D}_{t+1}, t'$

This Corollary is an immediate extension of the safety proof from Sui et al. (2015) to the time-varying case with an event trigger under the stated assumptions. Notwithstanding, it formalizes the conditions where ETSO is guaranteed to be safe and underscores the significance of a reliable event trigger. Essentially, the proposed event trigger allows for decoupling the entire process into individual, stationary sections, for which the safety rationale of existing safe BO can be applied. We will demonstrate in our experiments, that ETSO will detect significant changes. We will next describe implementation details regarding the event trigger and the new initialization after a reset.

Implementation details. We next describe key algorithmic choices in our implementation of ETSO. First, the data cannot be standardized for the GP, because each data point is added to the SafeOpt algorithm individually, and all sets are updated at each time step. Instead, we normalize the data with the rounded up absolute value of $\hat{J}_{\mathrm{B}}$ and set the prior mean $\mu_{\mathcal{D}_0} \coloneqq m(\boldsymbol{\theta}) = -1$, obtaining the relation $-1 = \mu_{\mathcal{D}_0} \leq \mu_{\mathcal{D}_1}(\boldsymbol{\theta}_{\mathrm{B}}) \leq \hat{J}_{\mathrm{B}}/\lceil \text{abs}(\hat{J}_{\mathrm{B}})\rceil$. From SafeOpt we also derive some restrictions on the safety threshold and the prior standard deviation $\sigma_{\mathcal{D}_0}$. $J_{\min,t}$ must be below the performance of the backup controller and above the lower bound $\ell_t(\boldsymbol{\theta})$ such that points for which we have no relevant information cannot be classified as safe. This gives us the relation

$$\mu_{\mathcal{D}_0} - \beta_0 \cdot \sigma_{\mathcal{D}_0} < J_{\min,t} < \mu_{\mathcal{D}_1}(\boldsymbol{\theta}_{\mathrm{B}}) - \beta_1 \cdot \sigma_{\mathcal{D}_1}(\boldsymbol{\theta}_{\mathrm{B}}), \tag{6}$$

and we choose $J_{\min,t}$ and $\sigma_{\mathcal{D}_0}$ as $J_{\min,t} = \mu_{\mathcal{D}_1}(\boldsymbol{\theta}_{\mathrm{B}}) - \beta_1 \cdot \sigma_n - \epsilon$ and $\sigma_{\mathcal{D}_0} = \max(\frac{\mu_{\mathcal{D}_0} - J_{\min,t} + \epsilon}{\beta_0}, \frac{1}{3})$, respectively, as well as $\epsilon = 0.2$. Additionally, we scale the threshold function of the trigger as

$$\kappa_t\left(\mathcal{D}_t, (\boldsymbol{\theta}_t, \hat{J}_t)\right) = 3/4 \cdot \sqrt{\rho_{t'}}\sigma_{\mathcal{D}_t}(\boldsymbol{\theta}_t) + 1/4 \cdot \bar{w}_{t'}. \tag{7}$$

The reason for this is that the infinite series in Lemma 2 will result in large bound over time–especially in the noise term $\bar{w}_{t'}$. While this is necessary for theoretical reasons (Srinivas et al., 2009; Bogunovic et al., 2016; Brunzema et al., 2023), it has practical downsides regarding safety as some changes may not be detected in later rounds. By having the tighter bound in (7) that still increases in time, we increase safety in practice by reverting to $\boldsymbol{\theta}_{\mathrm{B}}$ more quickly.

6. Experiments on a Quadcopter

We demonstrate our algorithm in simulation and on hardware of a quadcopter. Specifically, we use a Bitcraze Crazyflie 2.1 with the Crazyswarm firmware (Preiss et al., 2017) and use a Vicon camera system to obtain position estimates. For the simulation, we use the gym-pybulletdrones (Panerati et al., 2021) environment.[2] The controller of the Crazyflie is a cascaded PID (Mellinger and Kumar, 2011) as in Figure 3. We only tune the PI gains of the position PID, i.e., $\boldsymbol{\theta} = [P_{xy}, P_z, I_{xy}, I_z] \in \mathbb{R}^4$. As the backup controller, we set $\boldsymbol{\theta}_{\mathrm{B}} = [0.4, 1.25, 0.05, 0.05]$

Figure 3: Control structure on the Crazyflie 2.1.

as specified in the firmware. For the GP, we use a Matérn kernel ($\nu = 5/2$), lenghtscales $[0.15, 0.75, 0.025, 0.050]$ as obtained from simulation, and set the noise standard deviation to $\sigma_n = 0.016$ based on measurements obtained from the real system. Additionally, we use the parameter bounds $[0, 10]$ for each parameter, thus including many unstable configurations in the search space. We use $\beta_{\mathrm{const}} = 2$ as the β hyperparameter for SafeOpt; a value widely used in the literature (e.g., Berkenkamp et al. (2016)). Alternatively, a logarithmic β can be used, which increases with time (Kandasamy et al., 2015). This more conservative approach can improve safety. We choose β_{const} because it leads to faster learning, which is necessary given the quadcopter's battery life. We choose for the optimization the following cost functions for simulation and hardware, respectively,

$$J_{\mathrm{sim}}(\boldsymbol{\theta}) = -\sum_{q=1}^{Q} |e_{xyz}(q)|^{\frac{1}{2}}, \;\; J_{\mathrm{hw}}(\boldsymbol{\theta}) = \sum_{q=1}^{Q} |e_{xyz}(q)|^{\frac{1}{4}}, \tag{8}$$

where $e_{xyz}(q) = \min_{k \in \mathcal{T}_K} \sqrt{(x_k - x_q)^2 + (y_k - y_q)^2 + (z_k - z_q)^2}$, for a trajectory with waypoint index $q \in \mathcal{T}_Q$ and position measurements at time steps $k \in \mathcal{T}_K$ within an experiment. Hence, we measure performance as the minimum deviation achieved by the quadcopter to pre-defined waypoints. We perform three different experiments. In each of them, a variation occurs at time $\tau > T_L$:

- **2D Trajectory variation (2DTV)**: the quadcopter flies a figure eight in the xy-plane, and after τ, an hourglass-like figure is specified as the reference.
- **3D Trajectory variation (3DTV)**: the quadcopter flies a figure eight in the xy-plane, and after τ, a figure eight with altitude changes is specified as the reference.
- **Change in attitude controller (AC$_C$)**: the quadcopter flies a figure eight in the xy-plane with the default attitude controller, and at τ, we change the gains of the attitude controller of the quadcopter by multiplying them with the factor C (cf. Figure 3).[3]

We execute the simulation experiments for $T_{\mathrm{sim}} = 60$ rounds, with $T_{L,\mathrm{sim}} = 15$ rounds of exploration for the finite learning algorithms. Due to the limited battery life of the quadcopters, we perform fewer rounds in the hardware experiments. For the 2DTV and AC experiments, we define $T_{\mathrm{hw}} = 20$ and $T_{L,\mathrm{hw}} = 9$, and for the 3DTV experiment, $T_{\mathrm{hw}} = 16$ and $T_{L,\mathrm{hw}} = 7$. We compare our algorithm **ETSO** to **SafeOpt** using the specified budgets above. Further, we compare to **SafeOpt$_\infty$** as a variant with unlimited budget as well as to the performance of the safe **backup controller**.
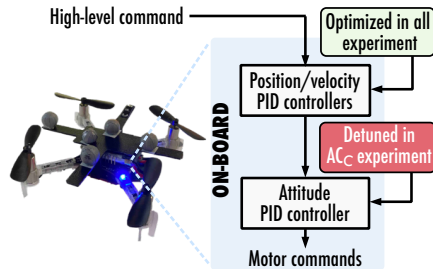
---

2. The repository for ETSO is freely available in https://github.com/antoHolz/ETSO.

3. In the output of an attitude controller, the computed torques usually have the form (from Luukkonen (2011)): $\tau_\phi = \left(K_{\phi,P} \cdot e_\phi + K_{\phi,I} \cdot \int e_\phi(t) + K_{\phi,D} \cdot \frac{\partial e_\phi(t)}{\partial t} \right) \cdot I_{xx}$ where I is the moments of inertia of the quadcopter. Here, changing all gains by some factor C is equivalent to changing the moment of inertia I_{xx}. In other words, changing the low level controller has the same effect as having miss-specified moment of inertia in the controller.
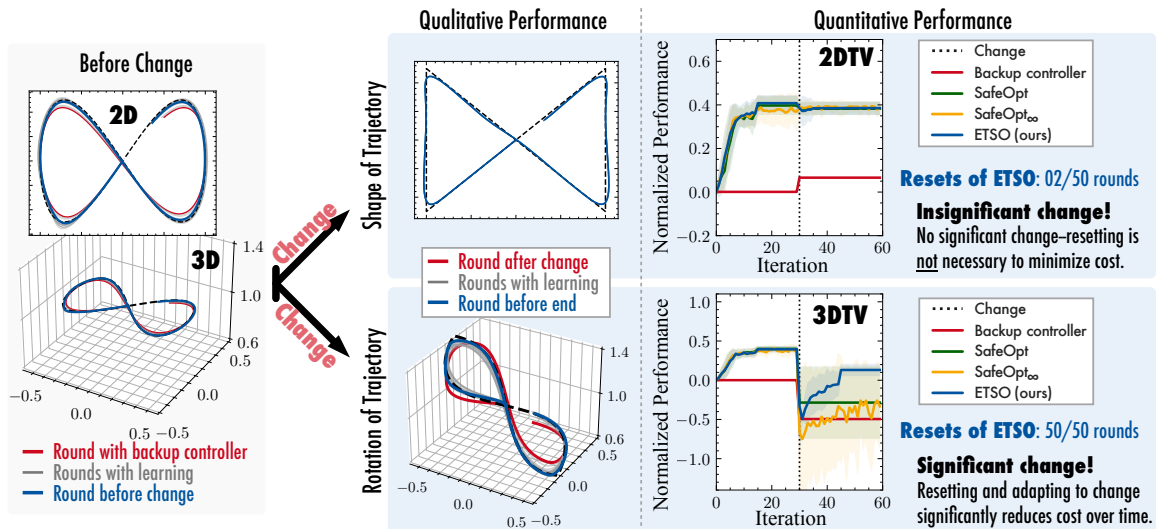
Figure 4: Trajectory change experiments in simulation. Left: Trajectories of the quadcopter before the variation time. Center: Trajectories of the quadcopter after the variation time, with ETSO. Right: Performance of the algorithms normalized by the initial performance of the backup controller as $\left(J_{\xi_0}(\boldsymbol{\theta}_B) - J_{\xi_t}(\boldsymbol{\theta}_t)\right)/J_{\xi_0}(\boldsymbol{\theta}_B)$ and the resets performed by ETSO.

6.1. Simulation Results

Figure 4 shows the results of the trajectory change experiments 2DTV and 3DTV. Both experiments have the same initial trajectory (a figure eight in the 2D plane) and undergo a change at $\tau = 30$. The right side of Figure 4 shows the performance of the algorithms for each experiment setup. For the experiment 2DTV (top), the change in reference results in an insignificant change in performance. With this insignificant change, the event trigger does not activate; hence, ETSO does not reset, saving resources compared to re-optimizing. Here, all baselines outperform the backup controller and yield similar performance. In the experiment 3DTV (bottom), ETSO outperforms SafeOpt$_\infty$ and SafeOpt. Changing the reference in altitude results in a significant change in the performance function. ETSO detects this change, resets (see bottom right in Figure 4), and then re-optimizes without stale data. We observe the same trend for ETSO and SafeOpt in Figure 5 (left) for the AC$_{.65}$ simulation experiments. However, using SafeOpt$_\infty$, the quadcopter crashed 50 out of 50 runs after some time steps after the system change. SafeOpt$_\infty$ assumes that areas that were previously safe remain safe, but a new system mode can shift the safe set. By resetting to $\boldsymbol{\theta}_B$, ETSO resets its safe set to ensure good performance as well as safe optimization over time with no crashes. SafeOpt also results in no crashes. Therefore, the system change in AC$_{.65}$ satisfied Assumption 2.

6.2. Hardware Results

The results of the hardware experiments displayed in Figure 5 confirm the insights from the simulations. Insignificant changes (2DTV) do not require re-learning. Here, ETSO and SafeOpt perform equivalently as ETSO does not reset. SafeOpt$_\infty$ exhibits marginally improved performance towards the end of the 2DTV experiment. The training time for the other algorithms is insufficient to find
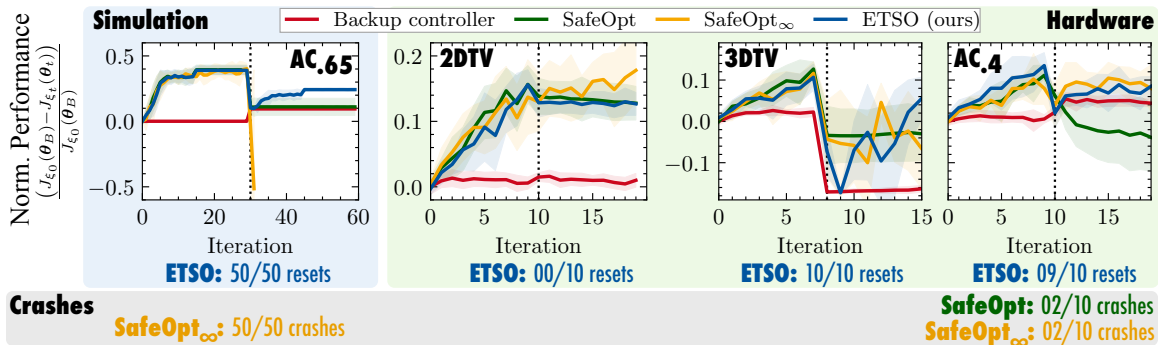
Figure 5: Performance in the remaining simulation and hardware experiments (without failures), normalized by the initial performance of the backup controller.

the optimal control parameters. This is unavoidable in the hardware setup due to battery constraints. For the experiments with the altitude change (3DTV), we observe that ETSO outperforms SafeOpt, picking the best performing controller at the last time step. In $AC_{.40}$, SafeOpt performs worse than in simulation. Furthermore, we recorded two crashes out of the 10 different rounds for SafeOpt. This highlights that in practice iterations with the quadcopter are not fully i.i.d. in the real world. As a remedy, we included a buffer time in between the rounds, but some influence of the previous control parameters in the next round was inevitable. ETSO and SafeOpt$_\infty$ yield similar performance, however, for SafeOpt$_\infty$ we recorded two out of 10 crashes. Using ETSO, we recorded no crashes, highlighting that our event trigger increases safety in practice by resetting the safe set in presence of significant changes. In Figure 1, we showed a comparative example of the performance of ETSO and SafeOpt$_\infty$. In the video, SafeOpt$_\infty$ fails two time steps after the change, while ETSO manages to reset to the backup controller and learn a better controller for the changed system.

7. Concluding Remarks

Event-Triggered SafeOpt (ETSO) is an algorithm that combines the concepts of SafeOpt with an event trigger as a first step toward safety in time-varying environments with only access to the cost signal. In our software and hardware experiments with a quadcopter, we find that:

- *(i)* ETSO only resets for significant changes. Without significant changes, ETSO performs equivalently to SafeOpt and SafeOpt$_\infty$ and better than the safe backup controller.
- *(ii)* After detecting significant changes, ETSO outperforms the other baselines.
- *(iii)* If changes cause the safe set to shift considerably, using SafeOpt or SafeOpt$_\infty$ can lead to failures during optimization, whereas ETSO avoids such failures by resetting the safe set.

ETSO is not restricted to the controller discussed here and fits any safety-critical optimization problem. It also allows for decoupling performance and safety in the optimization by using multiple constraints SafeOpt variants as in (Berkenkamp et al., 2021). Furthermore, ETSO does not depend on the existence of a dwell time, but it makes the change detection in Assumption 2 for systems with high dimension more likely. Nevertheless, Assumption 2 may not be viable for all systems. Therefore, extending ETSO to include crash prevention strategies as proposed in GoSafe-variants (Baumann et al., 2021; Sukhija et al., 2023) is a promising future direction.

Acknowledgments

References

Dominik Baumann, Fabian Mager, Romain Jacob, Lothar Thiele, Marco Zimmerling, and Sebastian Trimpe. Fast feedback control over multi-hop wireless networks with mode changes and stability guarantees. *ACM Transactions on Cyber-Physical Systems*, 4(2):1–32, 2019.

Dominik Baumann, Alonso Marco, Matteo Turchetta, and Sebastian Trimpe. Gosafe: Globally optimal safe robot learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4452–4458. IEEE, 2021.

Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with Gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 491–496. IEEE, 2016.

Felix Berkenkamp, Andreas Krause, and Angela P Schoellig. Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, pages 1–35, 2021.

Ilija Bogunovic, Jonathan Scarlett, and Volkan Cevher. Time-varying Gaussian process bandit optimization. In *Artificial Intelligence and Statistics*, pages 314–323. PMLR, 2016.

Paul Brunzema, Alexander Von Rohr, and Sebastian Trimpe. On controller tuning with time-varying Bayesian optimization. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 4046–4052. IEEE, 2022.

Paul Brunzema, Alexander von Rohr, Friedrich Solowjow, and Sebastian Trimpe. Event-triggered time-varying Bayesian optimization. *arXiv preprint arXiv:2208.10790*, 2023.

Roberto Calandra, André Seyfarth, Jan Peters, and Marc Peter Deisenroth. Bayesian optimization for learning gaits under uncertainty: An experimental comparison on a dynamic bipedal walker. *Annals of Mathematics and Artificial Intelligence*, 76:5–23, 2016.

Stefano De Blasi, Maryam Bahrami, Elmar Engels, and Alexander Gepperth. Safe contextual Bayesian optimization integrated in industrial control for self-learning machines. *Journal of Intelligent Manufacturing*, pages 1–19, 2023.

Yuntian Deng, Xingyu Zhou, Baekjin Kim, Ambuj Tewari, Abhishek Gupta, and Ness Shroff. Weighted Gaussian process bandits for non-stationary environments. In *International Conference on Artificial Intelligence and Statistics*, pages 6909–6932. PMLR, 2022.

Marcello Fiducioso, Sebastian Curi, Benedikt Schumacher, Markus Gwerder, and Andreas Krause. Safe contextual Bayesian optimization for sustainable room temperature pid control tuning. *arXiv preprint arXiv:1906.12086*, 2019.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

J.P. Hespanha and A.S. Morse. Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE Conference on Decision and Control*, volume 3, pages 2655–2660 vol.3, 1999.

Kirthevasan Kandasamy, Jeff Schneider, and Barnabás Póczos. High dimensional Bayesian optimisation and bandits via additive models. In *International conference on machine learning*, pages 295–304. PMLR, 2015.

Mohammad Khosravi, Annika Eichler, Nicolas Schmid, Roy S. Smith, and Philipp Heer. Controller tuning by Bayesian optimization an application to a heat pump. In *2019 18th European Control Conference (ECC)*, pages 1467–1472, 2019.

Andreas Krause and Cheng Ong. Contextual Gaussian process bandit optimization. *Advances in neural information processing systems*, 24, 2011.

Christopher König, Matteo Turchetta, John Lygeros, Alisa Rupenyan, and Andreas Krause. Safe and efficient model-free adaptive control via Bayesian optimization. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9782–9788, 2021.

Teppo Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22(22), 2011.

Alonso Marco, Philipp Hennig, Jeannette Bohg, Stefan Schaal, and Sebastian Trimpe. Automatic lqr tuning based on Gaussian process global optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 270–277. IEEE, 2016.

Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation*, pages 2520–2525. IEEE, 2011.

Matthias Neumann-Brosig, Alonso Marco, Dieter Schwarzmann, and Sebastian Trimpe. Data-efficient autotuning with Bayesian optimization: An industrial control study. *IEEE Transactions on Control Systems and Technology*, 2018.

Jacopo Panerati, Hehui Zheng, SiQi Zhou, James Xu, Amanda Prorok, and Angela P. Schoellig. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

James A. Preiss, Wolfgang Hönig, Gaurav S. Sukhatme, and Nora Ayanian. Crazyswarm: A large nano-quadcopter swarm. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3299–3304. IEEE, 2017.

Friedrich Solowjow and Sebastian Trimpe. Event-triggered learning. *Automatica*, 117:109009, 2020.

Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

Jie Su, Junfeng Wu, Peng Cheng, and Jiming Chen. Autonomous vehicle control through the dynamics and controller learning. *IEEE Transactions on Vehicular Technology*, 67(7):5650–5657, 2018.

Yanan Sui, Alkis Gotovos, Joel Burdick, and Andreas Krause. Safe exploration for optimization with Gaussian processes. In *International conference on machine learning*, pages 997–1005. PMLR, 2015.

Yanan Sui, Vincent Zhuang, Joel Burdick, and Yisong Yue. Stagewise safe Bayesian optimization with Gaussian processes. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4781–4789. PMLR, 10–15 Jul 2018.

Bhavya Sukhija, Matteo Turchetta, David Lindner, Andreas Krause, Sebastian Trimpe, and Dominik Baumann. Gosafeopt: Scalable safe exploration for global optimization of dynamical systems. *Artificial Intelligence*, 320:103922, 2023.

Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration for interactive machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Jonas Umlauft and Sandra Hirche. Feedback linearization based on Gaussian processes with event-triggered online learning. *IEEE Transactions on Automatic Control*, 65(10):4154–4169, 2019.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

Wenjie Xu, Colin N Jones, Bratislav Svetozarevic, Christopher R Laughman, and Ankush Chakrabarty. Violation-aware contextual Bayesian optimization for controller performance optimization with unmodeled constraints. *arXiv preprint arXiv:2301.12099*, 2023.

Xingyu Zhou and Ness Shroff. No-regret algorithms for time-varying Bayesian optimization. In *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2021.