Safe Bayesian Optimization for Complex Control Systems via Additive Gaussian Processes

Hongxuan Wang[1], Xiaocong Li[2], Lihao Zheng[3], Adrish Bhaumik[1], and Prahlad Vadakkepat[1]

***Abstract*—Controller tuning and optimization have long been recognized as fundamental challenges in robotics and mechatronic systems. Traditional controller design techniques are usually model-based, and their closed-loop performance depends on the fidelity of the mathematical model. Subsequent tuning of the controller parameters is frequently carried out via empirical rules, which may still suffer from model inaccuracies. In control applications with complex dynamics, obtaining a precise model is often challenging, leading us towards a data-driven approach. While various researchers have explored the optimization of a single controller, it remains a challenge to obtain the optimal controller parameters safely and efficiently when multiple controllers are involved. In this letter, a method called SAFECTRLBO is proposed to optimize multiple controllers simultaneously while ensuring safety. The exploration process in existing safe Bayesian optimization is simplified to reduce computational effort without sacrificing expansion capability. Additionally, additive Gaussian kernels are employed to enhance the efficiency of Gaussian process updates for unknown functions. Hardware experiments on a permanent magnet synchronous motor (PMSM) demonstrate that, compared to baseline safe Bayesian optimization algorithms, SAFECTRLBO attains the best overall performance while ensuring safety.**

***Index Terms*—Optimization and optimal control, machine learning for robot control, safe Bayesian optimization (BO), Gaussian process (GP), permanent magnet synchronous motors (PMSM).**

I. INTRODUCTION

OPTIMIZING complex systems with multiple controllers is challenging, particularly in cascade feedback architectures common in motor drives, as well as advanced schemes such as feedforward control, disturbance observers (DOB) [1], and active disturbance rejection control (ADRC) [2]. For instance, field-oriented control (FOC) of permanent magnet synchronous motors (PMSM) [3], [4], [5] typically involves tuning three proportional-integral (PI) controllers with six gains, each with distinct ranges. Manual tuning is time-consuming and expertise-demanding, motivating efficient automatic methods.

Traditional auto-tuning often relies on simplified linear models and is suboptimal under nonlinear dynamics [6]. Yet suboptimal operations contain information that model-based methods may underutilize. Data-driven optimization exploits this information, with iterative, sample-efficient approaches proving effective. In this line of research, various algorithms have been designed. Gradient-based methods require reliable gradients, but finite-difference or noisy estimates make them sensitive and locally trapped [7], even with robust variants [8]. Genetic algorithms [9] need extensive evaluations, limiting their applicability in hardware experiments.

Bayesian optimization (BO) [10] mitigates these issues by using Gaussian processes (GP) [11] to model system performance. However, standard BO may query unsafe parameters due to high uncertainty. Safety-aware BO methods like SAFEOPT [12] and STAGEOPT [13] introduce safe sets to avoid evaluating unsafe regions. Follow-up studies [14], [15], [16] have further examined the efficiency and theoretical foundations of safe BO. Practical implementations, such as quadrotor tuning [6], typically address low-dimensional problems (e.g., two parameters per controller). Similarly, [17] automated tuning for just two parameters in a simulated temperature control system.

High-dimensional optimization problems can sometimes be decomposed into simpler subspaces [18]. Leveraging this, LINEBO was introduced [19], decomposing the search space into one-dimensional subproblems. Although high-dimensional BO variants like LINEBO have been successfully applied to particle-accelerator tuning [20], those studies still demand a large number of experimental evaluations, which may face practical problems such as accumulated hardware wear, especially in industrial scenarios with tight experiment budgets. Additionally, typical industrial control problems often have moderate dimensionality, such as PMSM (6 parameters) [3], quadrotors (6 parameters) [6], [21], and gantry robots (6–9 parameters) [22], [23], [24], unlike the higher-dimensional problems addressed by LINEBO. The LoS-GP-UCB method [16] can handle problems with roughly 10–20 tunable parameters, but it requires a known upper bound of the global Lipschitz constant.

In high-dimensional settings, the information gain of standard stationary kernels grows exponentially with the dimension, making them sample-inefficient. Additive Gaussian processes [25] reduce this complexity by modeling the function as a sum of low-dimensional components, thereby achieving much faster learning while retaining flexibility [26], [27], [28].

Manuscript received: April, 13, 2025; Revised July, 25, 2025; Accepted August, 28, 2025.

This paper was recommended for publication by Editor Lucia Pallottino upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by RIE2025 Manufacturing, Trade and Connectivity (MTC) Industry Alignment Fund – Pre-Positioning (IAF-PP) under Grant M22K4a0044 through WP3-Energy Efficient Motor Drive System with GaN-based Traction Inverters. (*Corresponding author: Xiaocong Li.*)

[1]Hongxuan Wang, Adrish Bhaumik, and Prahlad Vadakkepat are with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 hongxuanwang@u.nus.edu, {adrish07, prahlad}@nus.edu.sg

[2]Xiaocong Li is with the College of Information Science and Technology, Eastern Institute of Technology, Ningbo, Ningbo, China 315200 xiaocongli@eitech.edu.cn

[3]Lihao Zheng is with the School of Data Science, The Chinese University of Hong Kong, Shenzhen, China 518172 lihaozheng@link.cuhk.edu.cn

Digital Object Identifier (DOI): see top of this page.

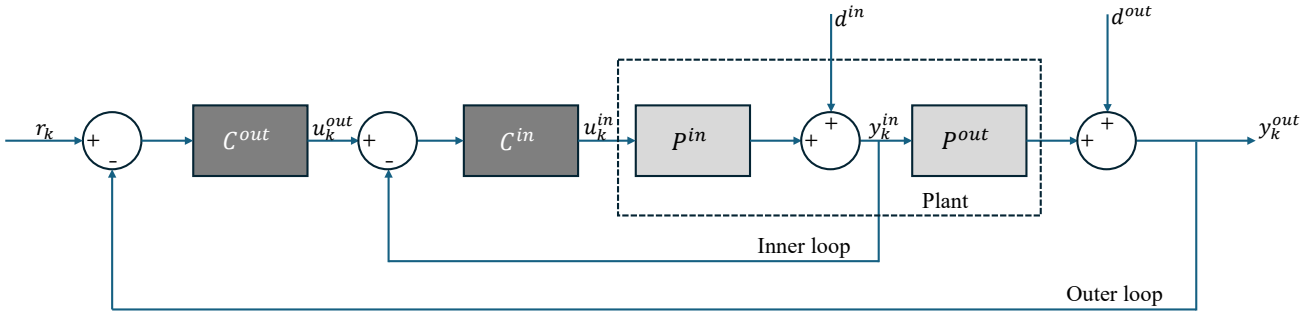


Fig. 1: A block diagram for a 2-layer cascade system. The dark grey blocks represent controllers, and the light grey blocks represent plants.

Very recently, additive Gaussian kernels are integrated with safe Bayesian optimization [29], and the algorithm is validated on a biaxial-gantry platform with three independent PID loops. However, the work is limited to parallel controllers, provides only empirical convergence evidence, and offers a limited systematic benchmark comparison.

Based on the above, we make three main contributions in this letter:

1) We extend safe Bayesian optimization to hierarchically coupled cascade controllers by introducing additive Gaussian kernels, thereby accommodating the widely different gain scales of inner and outer loops while preserving asymptotic safety guarantees.
2) We develop a simplified, boundary-set acquisition rule that cuts per-iteration computation without sacrificing safe exploration.
3) We validate the proposed method, SAFECTRLBO, on synthetic benchmarks and on a PMSM FOC drive with nested current–speed loops, showing performance on par with state-of-the-art safe BO algorithms under realistic industrial conditions.

II. PROBLEM STATEMENT

Similar to the safe optimization framework in [30], we consider complex cascade systems consisting of multiple controllers arranged such that the output of an outer-loop controller serves as the input for the subsequent inner-loop controller. Specifically, we adopt the discrete-time proportional-integral (PI) control law:

$$u_k = k_p(r_k - y_k) + k_i \sum_{t=0}^{k} (r_t - y_t), \tag{1}$$

where u_k denotes the control input, y_k the plant output, r_k the reference signal, and (k_p, k_i) represent the proportional and integral gains. For a two-layer cascade system (cf. Fig. 1), the inner- and outer-loop control laws become:

$$u_k^{\text{in}} = k_p^{\text{in}}(u_k^{\text{out}} - y_k^{\text{in}}) + k_i^{\text{in}} \sum_{t=0}^{k} (u_t^{\text{out}} - y_t^{\text{in}}), \tag{2}$$

$$u_k^{\text{out}} = k_p^{\text{out}}(r_k - y_k^{\text{out}}) + k_i^{\text{out}} \sum_{t=0}^{k} (r_t - y_t^{\text{out}}). \tag{3}$$

In the general case, denoting the outermost loop as layer 0 and the q-th inner loop as layer q, the control action at layer q can be expressed as:

$$u_k^q = g\left((y_k^0, y_k^1, \ldots, y_k^q), r_k, \mathbf{a}\right), \tag{4}$$

where $\mathbf{a} \in \mathcal{A}$ are the controller parameters, and $\mathcal{A}$ is the parameter domain.

Instead of explicitly modeling the plant dynamics, we treat the performance function $f : \mathcal{A} \to \mathbb{R}$ and the safety constraint functions $g_i : \mathcal{A} \to \mathbb{R}$ as black boxes that can only be queried through experiments or simulations. Starting from an initial safe parameter vector $\mathbf{a}_0$, at each iteration n we apply a candidate $\mathbf{a}_n$ and observe a noisy performance measurement $\tilde{f}(\mathbf{a}_n) = f(\mathbf{a}_n) + d_n$, where d_n denotes measurement noise. Stability requirements and physical limits (e.g., current, voltage, power) are captured by the noisy safety measurement $\tilde{g}_i(\mathbf{a}_n) = g_i(\mathbf{a}_n) + d_{i_n}$.

Our goal is to sequentially select $\mathbf{a}_1, \mathbf{a}_2, \cdots \in \mathcal{A}$ so as to maximize f while satisfying, with high probability, the safety constraints

$$\Pr\Big(\forall n \le N,\ \forall i :\ g_i(a_n) \ge h_i\Big) \ \ge\ 1 - \delta, \tag{5}$$

where the probability is over the measurement noise and the algorithm's randomness, h_i are user-defined safe thresholds, and $\delta \in (0, 1)$ is a confidence parameter (e.g. $\delta = 0.05$ yields a 95 % confidence level). Because each safety constraint is modelled by a Gaussian process (which will be discussed in Section III), satisfaction is guaranteed only with high-probability confidence bounds in the frequentist sense ($\ge 1 - \delta$). This formalizes a *high-probability safety guarantee*: we accept at most a δ probability of entering an unsafe region. As $\delta \to 0$ the guarantee approaches deterministic safety at the cost of more conservative exploration; larger δ relaxes conservativeness but increases risk. In practice, we choose $\delta = 0.05$, striking a balance between sample efficiency and operational safety, and complement the statistical guarantee with conventional hardware interlocks that act as a final fail-safe. This high-probability (frequentist) safety formulation follows [12]. The resulting parameters improve transient response, overshoot, and steady-state error without violating the safety constraints.

III. SAFE BAYESIAN OPTIMIZATION

In prior studies, safe BO methods have been designed to address sequential decision-making problems involving safety constraints. The unknown performance function f and each safety constraint function g_i are treated as independent realizations of GP priors, $f, g_i \sim \mathcal{GP}\big(\mu(\mathbf{a}),\ k(\mathbf{a},\mathbf{a}')\big)$, where $k : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ is a positive definite *kernel* function that encodes the smoothness of f and g_i. We assume $\mu(\mathbf{a}) = 0$ w.l.o.g., then given past observations, a GP predicts the mean and variance of the unknown function at unobserved points as follows [11]:

$$\mu_n(\mathbf{a}) = \mathbf{k}_n(\mathbf{a})(\mathbf{K}_n + \sigma_\omega^2 \mathbf{I}_n)^{-1}\tilde{\mathbf{f}}_n, \tag{6}$$

$$\sigma_n^2(\mathbf{a}) = k(\mathbf{a},\mathbf{a}) - \mathbf{k}_n(\mathbf{a})(\mathbf{K}_n + \sigma_\omega^2 \mathbf{I}_n)^{-1}\mathbf{k}_n^T(\mathbf{a}), \tag{7}$$

where $\tilde{\mathbf{f}}_n = [\tilde{f}(\mathbf{a}_1), \ldots, \tilde{f}(\mathbf{a}_n)]^\top$ is the vector of noisy observations, $\mathbf{K}_n$ is the kernel matrix with entries $[\mathbf{K}_n]_{ij} = k(\mathbf{a}_i, \mathbf{a}_j)$, and $\mathbf{k}_n(\mathbf{a}) = [k(\mathbf{a},\mathbf{a}_1), \ldots, k(\mathbf{a},\mathbf{a}_n)]$.

Existing safe BO methods fall into two main theory families. (i) RKHS-based GP approaches (e.g. SAFEOPT [12]) assume that the unknown performance and safety constraint functions have bounded norm in the associated reproducing kernel Hilbert space (RKHS) [13], [31], which yields high-probability GP confidence intervals. (ii) Lipschitz-based approaches (e.g. LoS-GP-UCB [16] and the recent work in [32]) dispense with the RKHS norm for safety guarantees and rely instead on a known global Lipschitz constant, while they may still invoke GP-based confidence bounds (hence an implicit RKHS membership) for regret analysis. Our work focuses on the first family and therefore adopts the bounded-RKHS-norm assumption, while also assuming global Lipschitz continuity to tighten exploration bounds.

Under the bounded-RKHS-norm assumption, the GP prediction is confined within confidence intervals defined by:

$$u_n(\mathbf{a}) = \mu_{n-1}(\mathbf{a}) + \beta_n \sigma_{n-1}(\mathbf{a}), \tag{8}$$

$$l_n(\mathbf{a}) = \mu_{n-1}(\mathbf{a}) - \beta_n \sigma_{n-1}(\mathbf{a}). \tag{9}$$

The bounds in (8)–(9) formally hold pointwise for a given n. To obtain rigorous safety guarantees, they must hold simultaneously for all $\mathbf{a} \in \mathcal{A}$ and all $n \le N$, which requires a suitably growing sequence β_n. For practical comparability with prior baselines, we set $\beta = 2$ as a heuristic choice in this work.

Acquisition functions then select new evaluation points based on these confidence intervals. Examples include the GP-UCB algorithm [33], which maximizes the upper confidence bound, and SAFEOPT [6], which maximizes variance within safe regions.

Most SAFEOPT-type methods organize the search around three working sets:

- Safe set ($\mathcal{S}_n$): the set of parameters where all safety constraints are satisfied with high probability,

$$\mathcal{S}_n = \{\mathbf{a} \in \mathcal{A} \mid l_n^f(\mathbf{a}) \ge h_f \ \wedge\ \forall i :\ l_n^{(i)}(\mathbf{a}) \ge h_i\}. \tag{10}$$

  Here $l_n^f(\mathbf{a})$ denotes the lower confidence bound (LCB) of the performance function f, and $l_n^{(i)}(\mathbf{a})$ denotes the LCB of the safety constraint functions g_i, at iteration n.

- Set of potential expanders ($\mathcal{E}_n$): the set of parameters within the safe set that could expand the safe region upon evaluation,

$$\mathcal{E}_n = \{\mathbf{a} \in \mathcal{S}_n \mid \forall i,\ \exists \mathbf{a}' \in \mathcal{A}\backslash\mathcal{S}_n,\ l^{(i)}_{n,(\mathbf{a},u_n(\mathbf{a}))}(\mathbf{a}') \ge h_i\}. \tag{11}$$

  Here $l^{(i)}_{n,(\mathbf{a},u_n(\mathbf{a}))}(\mathbf{a}')$ denotes the LCB that would result after adding a hypothetical observation $g_i(\mathbf{a}) = u_n(\mathbf{a})$ to the data set at iteration n.

- Set of potential maximizers ($\mathcal{M}_n$): the set of safe parameters likely to yield optimal performance measure,

$$\mathcal{M}_n = \{\mathbf{a} \in \mathcal{S}_n \mid u_n^f(\mathbf{a}) \ge \max_{\mathbf{a}' \in \mathcal{S}_n} l_n^f(\mathbf{a}')\}. \tag{12}$$

These sets are updated by comparing GP confidence bounds with pre-defined safe thresholds. By restricting every query to the safe set $\mathcal{S}_n$, SAFEOPT–type methods obtain a *δ-high-probability safety guarantee* in the frequentist sense (Equation (5)). Other strands of safe BO, e.g., LoS-GP-UCB [16], guarantee safety through a global Lipschitz constant.

IV. SAFECTRLBO

A. Additive Gaussian Kernels

Despite recent advances in high-dimensional safe BO (e.g., SWARMSAFEOPT [6] and LINEBO [19]), both methods still rely on stationary kernels such as the squared-exponential (SE) or Matérn family, whose maximum information gain remains unfavourable in large parameter spaces. For a Gaussian-process prior $f \sim \mathcal{GP}(0, k)$ and an evaluation budget of T, we obtain a set of noisy observations $\mathbf{y}_A = \big\{f(\mathbf{a}) + \varepsilon_{\mathbf{a}} \mid \mathbf{a} \in A,\ |A| = T\big\}$. The maximum information gain is then defined as $\gamma_T = \max_{A \subset \mathcal{A},\, |A| = T} I\big(\mathbf{y}_A;\, f\big)$, where $I(\cdot;\cdot)$ denotes mutual information. This quantity controls the regret and safety bounds of GP-based BO algorithms; smaller γ_T implies fewer samples are needed for a given accuracy [33], [34]. According to [35], for a d-dimensional SE kernel, the worst-case gain scales as $\gamma_T = \mathcal{O}\big((\log T)^{d+1}\big)$, whereas for a d-dimensional additive or Bayesian linear kernel, one has the much milder $\gamma_T = \mathcal{O}(d \log T)$. Motivated by the fact that the transient-response cost of a cascade controller is *approximately additive* (the sum of contributions from the inner-loop gains, the outer-loop gains, and a few low-order cross terms under the usual time-scale separation assumption), we adopt an additive Gaussian-process prior [25] and equip the original stationary radial kernel with this additive structure. This additive prior explicitly encodes the known functional structure and *as a by-product*, reduces the maximum information gain γ_T, thereby improving sample efficiency.

The additive kernels for each order are formed by summing combinations of base kernels, where the base kernels we use are 1-D Gaussian kernels, $k(\mathbf{a}_i, \mathbf{a}_j) = \exp\left(-\dfrac{\|\mathbf{a}_i - \mathbf{a}_j\|^2}{2\sigma^2}\right)$. Let $k_i(\mathbf{a},\mathbf{a}')$ be the base kernel acting on the i-th coordinate. For any index set $S \subseteq [D] = \{1, \ldots, D\}$, define the $|S|$-way product kernel $k_S(\mathbf{a},\mathbf{a}') = \prod_{j \in S} k_j(\mathbf{a},\mathbf{a}')$. Then the additive kernels of order $m = 1, \ldots, D$ are

$$k_{\mathrm{add},m}(\mathbf{a},\mathbf{a}') = \sum_{\substack{S \subseteq [D] \\ |S| = m}} k_S(\mathbf{a},\mathbf{a}'), \quad m = 1, \ldots, D. \tag{13}$$

To include all interaction orders up to $n \leq D$, we define

$$k_{\text{add},\leq n}(\mathbf{a}, \mathbf{a}') \;=\; \sum_{m=1}^{n} \sum_{\substack{S \subseteq [D] \\ |S|=m}} k_S(\mathbf{a}, \mathbf{a}'). \tag{14}$$

Although we implement the additive structure with 1-D Gaussian kernels, the same construction applies to any stationary 1-D base kernel, including Matérn. Replacing Gaussian by Matérn simply changes the constant factors in the information-gain bound.

B. Acquisition Functions

Following the *stagewise* paradigm introduced by StageOpt [13], SafeCtrlBO alternates between a *safe–expansion* stage and an *optimization* stage. During the first stage, the algorithm tries to enlarge the current safe set $\mathcal{S}_n$; once no further expansion is possible (or a budget is reached), it switches to maximizing the performance function inside $\mathcal{S}_n$.

The original set of potential expanders $\mathcal{E}_n$ is expensive to enumerate in high dimensions. In practice, we retain only points whose GP posterior variance is large and whose Euclidean distance to the nearest *evaluated* safe point exceeds a user–defined margin. These high–variance points *often*, though not always, lie on the outer boundary of $\mathcal{S}_n$. We formalize this heuristic with the following definition.

Definition 1: We define the set of safe boundary points $\mathcal{B}_n$ by

$$\mathcal{B}_n = \{\mathbf{a} \in \mathcal{S}_n \mid \exists\, i : |l_n^{(i)}(\mathbf{a}) - h_i| \leq \tau\},$$

where the tolerance $\tau > 0$ guarantees $B_n \neq \varnothing$ even in a discrete domain.

In the *safe-expansion* stage, we pick the most uncertain boundary point

$$\mathbf{a}_n \;=\; \arg\max_{\mathbf{a} \in \mathcal{B}_n} \sigma_{n-1}^{f}(\mathbf{a}), \tag{15}$$

which is cheap to compute and focuses evaluations on the "edge" of knowledge.

After the safe set stops growing, we invoke the standard GP-UCB rule restricted to $\mathcal{S}_n$ (*optimization* stage). The complete SafeCtrlBO procedure is provided in Algorithm 1. The additive kernel of all interaction orders ($k_{\text{add},D} = \sum_{m=1}^{D} k_{\text{add},m}$) is used throughout both stages.

C. Asymptotic Analysis

The asymptotic efficiency of Bayesian optimization methods is closely related to the maximum information gain γ_T, which quantifies how quickly uncertainty is reduced over T noisy evaluations. According to [35], for the SE kernel, and more generally, any stationary kernel with super-exponential eigenvalue decay, the maximum information gain scales as $\gamma_T = \mathcal{O}\big((\log T)^{d+1}\big)$. For finite-smoothness Matérn kernels, the bound is polynomial in T and grows faster with d [36], where d is the dimensionality of the parameter space.

In contrast, when applying additive kernels, the maximum information gain behaves more favorably. Specifically,

Algorithm 1 SafeCtrlBO

Inputs: Controller parameter domain $\mathcal{A}$
Safe GP prior for performance function f and safety constraint functions g_i, $i \in \{1, \ldots, m\}$
Safe thresholds h_f, h_i, $i \in \{1, \ldots, m\}$
Additive kernel $k_{\text{add},D}$
Initial, safe controller parameter and its noisy performance and safety measure $(\mathbf{a}_0, \tilde{f}(\mathbf{a}_0), \tilde{g}_i(\mathbf{a}_0))$
Stage switching time T_0

Initialize GP with $(\mathbf{a}_0, \tilde{f}(\mathbf{a}_0), \tilde{g}_i(\mathbf{a}_0))$
for $n = 1, 2, \ldots, T_0$ (safe expansion stage) **do**
  $\mathcal{S}_n \leftarrow \{\mathbf{a} \in \mathcal{A} \mid l_n^f(\mathbf{a}) \geq h_f \;\wedge\; \forall i :\; l_n^{(i)}(\mathbf{a}) \geq h_i\}$
  $\mathcal{B}_n \leftarrow \{\mathbf{a} \in \mathcal{S}_n \mid \exists\, i \in \{1, \ldots, m\} : |l_n^{(i)}(\mathbf{a}) - h_i| \leq \tau\}$
  $\mathbf{a}_n \leftarrow \arg\max_{\mathbf{a} \in \mathcal{B}_n} \sigma_{n-1}^{f}(\mathbf{a})$
  Obtain noisy measurement $\tilde{f}(\mathbf{a}_n), \tilde{g}_i(\mathbf{a}_n)$
  Update GP with $(\mathbf{a}_n, \tilde{f}(\mathbf{a}_n), \tilde{g}_i(\mathbf{a}_n))$
end for
for $n = T_0 + 1, \ldots$ (optimization stage) **do**
  $\mathcal{S}_n \leftarrow \{\mathbf{a} \in \mathcal{A} \mid l_n^f(\mathbf{a}) \geq h_f \;\wedge\; \forall i :\; l_n^{(i)}(\mathbf{a}) \geq h_i\}$
  $\mathbf{a}_n \leftarrow \arg\max_{\mathbf{a} \in \mathcal{S}_n} u_n^f(\mathbf{a})$
  Obtain noisy measurement $\tilde{f}(\mathbf{a}_n), \tilde{g}_i(\mathbf{a}_n)$
  Update GP with $(\mathbf{a}_n, \tilde{f}(\mathbf{a}_n), \tilde{g}_i(\mathbf{a}_n))$
end for

if the kernel structure is fully additive (e.g., summing one-dimensional kernels), the information gain scales linearly with dimensionality $\gamma_T = \mathcal{O}(d \log T)$. More generally, using additive kernels of limited order (such as second-order interactions) yields an information gain of $\gamma_T = \mathcal{O}\left(d^n (\log T)^{n+1}\right)$, where n is the order of the additive kernel interactions considered ($n \ll d$ typically).

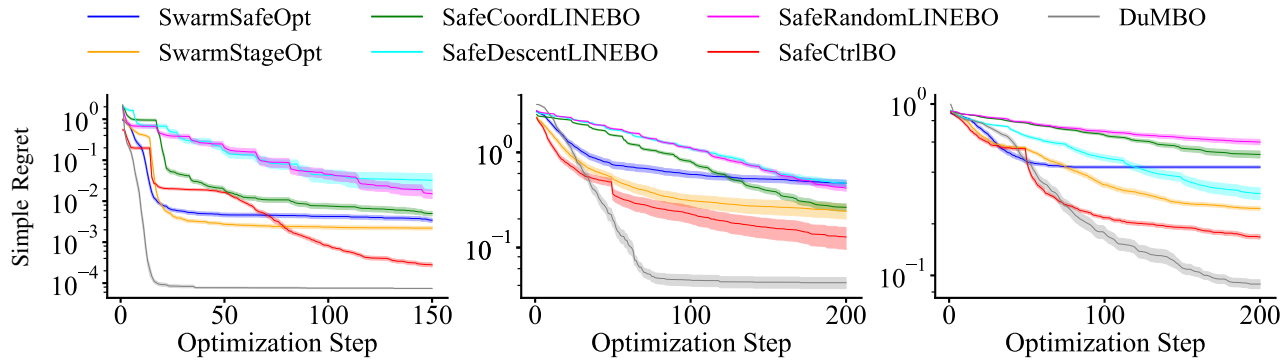
Thus, the use of additive kernels in SafeCtrlBO drastically reduces the asymptotic information complexity compared to standard safe BO approaches. However, the improved rate $\gamma_T = \mathcal{O}(d \log T)$ holds only when the *true* performance and safety constraint functions admit an (approximately) additive decomposition in the same coordinate blocks used by the kernel [25]. In our cascade-control setting, the cost is empirically well approximated by low-order interactions among inner- and outer-loop gains. For problems without such structure, one should use either higher-order additive terms or alternative kernels such as sparse-product [37] or spectral mixtures [38].

D. Computational Complexity Analysis

In standard safe BO methods, regardless of whether the covariance kernel is Gaussian, Matérn, or another stationary kernel, the dominant per-iteration cost is GP inference: inverting the $n \times n$ covariance matrix of the n observations, an $\mathcal{O}(n^3)$ operation.

SafeCtrlBO replaces the single high-dimensional kernel by an *additive* kernel that sums several low-dimensional components. For a fully additive kernel, the number of components is $\sum_{i=1}^{d} \binom{d}{i} = 2^d - 1$. Each component still needs one $\mathcal{O}(n^3)$ matrix inversion, leading to a worst-case

$$\mathcal{O}(M\, n^3) \tag{16}$$



(a) Camelback2D (b) Hartmann6D (c) Gaussian10D

Fig. 2: Optimization for synthetic benchmark functions.

time per iteration, where M denotes the number of additive kernel components used.

Standard SAFEOPT variants enumerate a potentially large expander set $\mathcal{E}_n$ in the full parameter space, an operation that can grow exponentially in d. SAFECTRLBO adopts a stage-wise scheme and reduces $\mathcal{E}_n$ to the *boundary set* $\mathcal{B}_n$. The safe-expansion acquisition simply selects the boundary point with maximal posterior variance, $\mathbf{a}_n = \arg\max_{\mathbf{a}\in\mathcal{B}_n} \sigma^{\hat{f}}_{n-1}(\mathbf{a})$, so the cost scales linearly with $|B_n|$ instead of with the size of the full grid.

a) Summary:

$$\text{Standard GP safe BO:} \quad \mathcal{O}\big(n^3 + |\mathcal{E}_n|\, n\big) \text{ per iteration,}$$
$$\text{SAFECTRLBO:} \quad \mathcal{O}\big(Mn^3 + |\mathcal{B}_n|\, n\big) \text{ per iteration.}$$

Although additivity multiplies the cubic term by M, the simplified safe expansion acquisition greatly reduces the dimensionality-dependent overhead.

b) Discussion: The additive kernel improves the information-gain rate from $\mathcal{O}((\log T)^{d+1})$ (SE kernel) to $\mathcal{O}(d \log T)$ (fully additive), which leads to orders-of-magnitude fewer iterations. Because each hardware experiment consumes time and induces mechanical wear, the net runtime and lifetime cost are dominated by *how many* iterations must be run, not by a modest increase in per-iteration computation, hence the raw n^3 scaling is usually harmless. The cost increase in SAFECTRLBO stems from the multiplicative factor $M = \mathcal{O}(d^2)$ attached to the matrix inversion when additive kernels are used. Conversely, the algorithm *reduces* the dimensionality-driven overhead by replacing the exponentially large expander set $\mathcal{E}_n$ with the much smaller boundary set $\mathcal{B}_n$. Empirical results on the PMSM cascade drive (Section V-B) confirm that SAFECTRLBO achieves the best overall time-to-performance among the tested safe-BO baselines.

V. EMPIRICAL STUDY

A. Simulations on Synthetic Benchmark Functions

In this section, we compare the safe-optimization performance of SAFECTRLBO with several established baselines on standard benchmark functions: Camelback (2D), Hartmann (6D), and a synthetic Gaussian function (10D).

Baselines. (i) SWARMSAFEOPT and SWARMSTAGEOPT, the high-dimensional implementations of SAFEOPT and STAGEOPT, respectively; (ii) three safe variants of LINEBO [19] that differ in how the one-dimensional search line is chosen: SAFECOORDINATELINEBO (axis-aligned line through the incumbent), SAFEDESCENTLINEBO (line along the local GP descent direction), and SAFERANDOMLINEBO (random line through the incumbent); and (iii) DUMBO [39], a state-of-the-art high-dimensional *unconstrained* additive BO algorithm that we include as a reference.

Since all three benchmark functions are designed for minimization, their outputs are inverted, and safe thresholds are introduced. For the inverted Camelback function, the maximum value is 1.0316, and a safe threshold is set at 0. Similarly, for the inverted Hartmann function, the maximum value is approximately 3.32237, and a safe threshold of 0.3 is imposed. For the Gaussian 10D function, $y = -\exp(-4\|x\|_2^2)$, the inverted maximum value is 1, and the safe threshold is set at 0.1.

To ensure a fair comparison, we adopt the noise model, GP hyper-parameters, and confidence-scale factor β used in the public implementation of LINEBO [19]: Gaussian noise $\varepsilon \sim \mathcal{N}(0,\, 0.04)$; squared-exponential base kernels with variance 1 and length-scale 0.2; and an empirical fixed $\beta = 2$. For Camelback and Hartmann functions, we use the additive kernel of all interaction orders. For Gaussian function, we include only first-, second-, and the highest-order components to avoid the $2^d - 1$ explosion in kernel count. 100 runs were conducted for each method on each benchmark function, plotting the mean and standard error of the simple regret (Fig. 2). Each run begins with a randomly generated safe initial point where the function value exceeds the safe threshold. All optimization methods were executed for 150 iterations on the Camelback function and 200 iterations on the Hartmann and Gaussian functions. The safe-expansion phase is capped at $T_0 = 15$ iterations for Camelback and $T_0 = 50$ for Hartmann and Gaussian.

a) Results: Fig. 2 presents the simulation results. DUMBO attained the lowest average simple regret across all benchmark functions but violated the safe thresholds 1009, 3820, and 12,557 times (over 100 runs) on the Camelback,

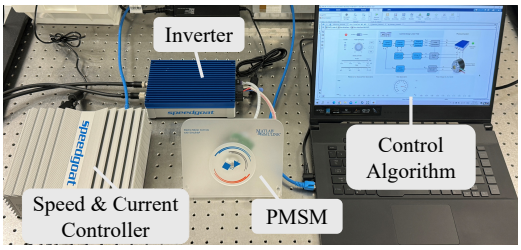


Fig. 3: Hardware experimental setup.

Hartmann, and Gaussian functions, respectively. In contrast, none of the safe BO algorithms violated any constraints under the specified settings, and SAFECTRLBO produced more competitive results.

B. Hardware Experiments on a SpeedGoat Real-Time Target Machine

In this section, hardware experiments are conducted using a SpeedGoat real-time target machine[1], as shown in Fig. 3. The configuration includes a SpeedGoat controller with integrated speed and current loops, a SpeedGoat inverter, and a PMSM.

Field-oriented control (FOC) algorithm is used to control the PMSM. This algorithm comprises a cascade control loop. The external speed controller is responsible for regulating the motor's rotational speed, and the two internal current controllers (d-axis and q-axis) manage the PWM output from the inverter. This nested cascade loops match our theoretical setting and offer a hardware platform whose dynamics are fast enough to run 100 safe BO iterations within laboratory time constraints.

The PMSM drive is a *core actuator* in a wide range of robotic motion systems, e.g., industrial manipulators, mobile bases, collaborative arms, and humanoid joints, all rely on FOC of PMSMs. Our objective is to optimize the controller parameters, maximizing transient-response speed while simultaneously minimizing overshoot and steady-state error. Such an objective is crucial in industrial applications ranging from precise robot joint control to electric vehicle control. Transient response is evaluated using the 2% settling time. The performance function is then designed as:

$$f = w_s \cdot (t_0 - t_s) - w_o \cdot O_s - w_e \cdot e_{ss}, \tag{17}$$

where w_s, w_o, and w_e are weight factors, t_0 is a time constant depending on the task, t_s is the value of settling time, O_s is the value of overshoot, and e_{ss} is the value of steady-state error. In the experiments, $w_s = 20$, $w_o = 1.5$, $w_e = 4$, $t_0 = 2.5$.

To guarantee safety, the motor system must remain stable with a sufficiently small steady-state error, and the control signal must be moderated to prevent excessive current that could damage the motor hardware. Two safety constraint functions are defined to address these concerns:

$$g_1 = C_{e0} - w_e' \cdot e_{ss}, \quad g_2 = C_{u0} - w_u \cdot \sum_{t=0}^{1} u(t)^2, \tag{18}$$

[1]https://www.speedgoat.com/products-services/real-time-target-machines/baseline-real-time-target-machine

where C_{e0} and C_{u0} are constants defined according to the system characteristics, and w_e' and w_u are weight factors. In the experiments, $C_{e0} = C_{u0} = 100$, $w_e' = 40$, and $w_u = 0.001$. The safe thresholds of both safety constraint functions are set at 0. The model's predefined parameters serve as the initial settings, and evaluations of these initial settings confirm that they meet the minimum threshold requirements.

In the PMSM FOC control loop, the six controller parameters have different physical meanings and parameter ranges. Specifically, (k_p, k_i) of the speed controller are set within the range $[0.01, 0.5]$, while the k_p of the d-axis and q-axis current controllers are set within $[0.1, 1]$, and their k_i are set within $[1, 200]$.

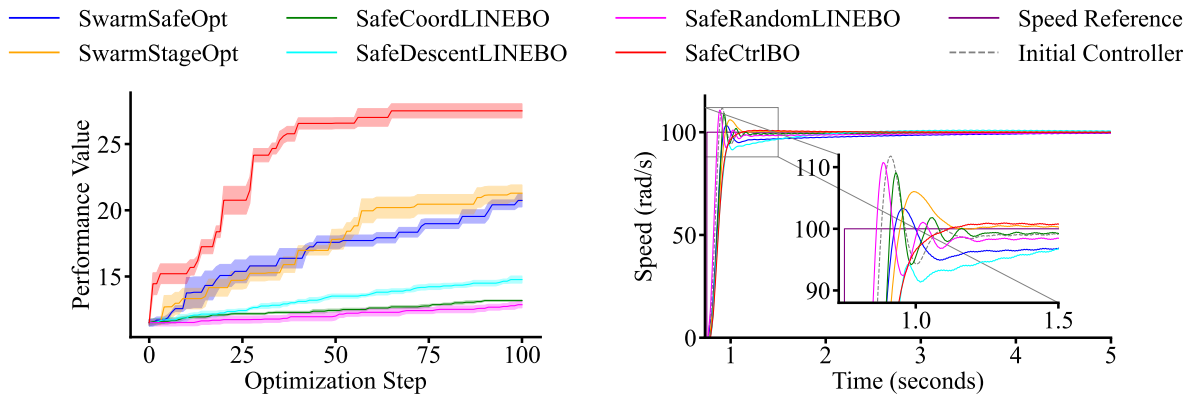
Due to hardware safety considerations, only SWARMSAFEOPT, SWARMSTAGEOPT, three variants of LINEBO, and SAFECTRLBO are compared. Because LINEBO can only deal with performance function, g_1 and g_2 are not applied in LINEBO. Five runs of each method were conducted, with each run comprising 100 iterations. The mean and standard error of the performance curves for all methods are depicted in Fig. 4a. Additionally, the highest performance measures among the five runs for each method are shown in Fig. 4b, and detailed metrics are provided in Table I.

a) Results: Figure 4a shows that all six methods continuously improve PMSM speed-tracking, with SAFECTRLBO achieving the most pronounced gains. From the curves in Fig. 4b and the metrics in Table I, SAFECTRLBO attains the smallest overshoot, the shortest 2% settling time, and the second-smallest steady-state error.

Constraint violations. Because LINEBO cannot enforce the steady-state-error or control signals, we record only violations of the performance threshold h_f: SAFECOORDINATELINEBO violates h_f once, SAFEDESCENTLINEBO four times, and SAFERANDOMLINEBO five times. The other methods enforce all three thresholds— h_f for performance, h_1 and h_2 for safety: SWARMSAFEOPT incurs 39 violations over five runs, SWARMSTAGEOPT 61, and SAFECTRLBO also 39. For SAFECTRLBO, the violations were observed primarily during the safe expansion stage. Most violations involved h_f, which is a soft threshold on performance; falling below h_f indicates suboptimal outcomes (e.g., large overshoot or slower transient response) but does not imply system unsafe. Such kinds of constraint violations are also reported in other studies [30].

b) Computation time: Table II reports the average wall–clock time per iteration (mean over 100 BO iterations on an Intel Ultra 9, 24-core CPU). Using the *full* expander set $\mathcal{E}_n$, SAFECTRLBO takes 19.87 s per iteration. With our boundary-set heuristic, the time drops to 13.10 s, a $\sim$34% reduction. For reference, we also measured SWARMSAFEOPT with the standard SE kernel: replacing $\mathcal{E}_n$ by $\mathcal{B}_n$ cuts its runtime from 0.280 s to 0.215 s. Although the additive kernel incurs a larger constant factor ($\mathcal{O}(Mn^3)$ vs. $\mathcal{O}(n^3)$), the net wall-clock time to a given performance level remains competitive because SAFECTRLBO needs markedly fewer hardware iterations (cf. Fig. 4).

c) Cost regimes: Whether the host-side overhead of SAFECTRLBO ($\mathcal{O}(Mn^3)$ per iteration) is important depends on the relative cost of running the plant:



(a) Comparison of performance values.

(b) Comparison of speed tracking curves.

Fig. 4: Hardware experiment results.

TABLE I: Performance comparison of the best PMSM speed tracking curves. The best results are written in **bold** text, and the second-best results are underlined.

Method	$J(a^*)$	$O_s(rad/s)$	$e_{ss}(rad/s)$	$2\%\ t_s(s)$
Initial Setting	11.6788	11.789	0.067	0.301
SAFEOPT	21.4277	3.323	0.417	1.721
STAGEOPT	23.7091	6.063	**0.030**	0.327
COORLINEBO	13.2615	9.743	0.081	0.431
DESCLINEBO	14.734	1.028	1.681	0.988
RANDLINEBO	12.2435	10.747	0.409	0.477
SAFECTRLBO	**28.7893**	**0.956**	0.037	**0.284**

TABLE II: Average computation time per iteration.

Method	Kernel	Expander set	Time (s)
Standard SWARMSAFEOPT	SE	full $\mathcal{E}_n$	0.280
Modified SWARMSAFEOPT	SE	boundary $\mathcal{B}_n$	0.215
SAFECTRLBO (ours)	Additive SE	boundary $\mathcal{B}_n$	13.10
Modified SAFECTRLBO	Additive SE	full $\mathcal{E}_n$	19.87

(i) Fast plant, cheap compute (e.g., pure simulation): wall-clock time is dominated by the number of iterations.
(ii) Slow plant, cheap compute (our PMSM drive): plant runtime dwarfs CPU time (< 20 s vs. > 4 min).
(iii) Slow plant, *costly* compute (e.g., high-fidelity CFD, large-d embedded MPC): both terms matter; additive kernels become advantageous only when combined with kernel-selection [40] or sparse linear algebra to curb the Mn^3 factor.

d) Discussion of hyper-parameter choices: **Budget** T_0: Excessive exploration is impractical in hardware tuning. On our PMSM platform, one 100-step optimization run lasts several hours (motor response dominates the wall-clock time), so we cap the initial safe-expansion stage at $T_0 = 15$, and stable, high-performance control was achieved after approximately 45 iterations.

β **and GP kernel hyper-parameters**: To ensure a fair comparison with the six baselines (including the three variants of LINEBO from [19]), we keep all common GP hyper-parameters identical across methods whenever possible; in particular, we set $\beta = 2$. We note, however, that this heuristic choice of β is a well-known limitation of SAFEOPT-type algorithms: the theoretically correct scaling should depend on the (unknown) RKHS-norm bound in order to guarantee δ-safety. In practice, adopting a fixed β (as commonly done in the SAFEOPT literature) cannot ensure strict theoretical safety and may occasionally lead to safety violations, as also observed in our experiments. Developing adaptive or theoretically justified schemes for β-selection therefore remains an important direction for future work.

Kernel variances and length-scales: FOC uses three nested PI loops. Denote the proportional and integral gains of the speed loop by $(k_{p,s}, k_{i,s})$, those of the q-axis current loop by $(k_{p,q}, k_{i,q})$, and those of the d-axis current loop by $(k_{p,d}, k_{i,d})$. Their influence on the closed-loop response decreases in the order $(k_{p,s}, k_{i,s}) > (k_{p,q}, k_{i,q}) > (k_{p,d}, k_{i,d})$. Accordingly, we assign larger kernel variances to the first pair, moderate variances to the second pair, and smaller variances to the third.

VI. CONCLUSIONS

This letter introduces SAFECTRLBO, a safe Bayesian optimization framework that combines additive Gaussian kernels with a simplified boundary-set expansion rule. The additive prior reduces the worst-case information gain, while the boundary-set heuristic avoids the exponential search cost of classical SAFEOPT. Benchmarks on synthetic functions and a field-oriented-controlled PMSM drive show that SAFECTRLBO achieves state-of-the-art sample efficiency without violating current, voltage, or stability limits.

Limitations and future work. The number of additive components grows rapidly with dimension. Beyond $d \approx 20$, fully or second-order additive kernels become computationally expensive and difficult to hand-design. Recent kernel-selection technique [40] offers a remedy by automatically identifying the most informative components; integrating such selection into SAFECTRLBO and evaluating it in very high-dimensional settings ($d > 100$) against methods like HDSAFEBO [41] is an ongoing research direction. In addition, we plan to validate the algorithm on larger robotic platforms, such as high-DoF

manipulators and quadrotors, where the PMSM controller studied here acts as a fundamental actuator module.

REFERENCES

[1] H. Jung and S. Oh, "Data-driven optimization of integrated control framework for flexible motion control system," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4762–4772, 2022.

[2] H. Cao, Y. Deng, Y. Zuo, H. Li, J. Wang, X. Liu, and C. H. T. Lee, "Improved adrc with a cascade extended state observer based on quasi-generalized integrator for pmsm current disturbances attenuation," *IEEE Transactions on Transportation Electrification*, vol. 10, no. 1, pp. 2145–2157, 2024.

[3] R. Gabriel, W. Leonhard, and C. J. Nordby, "Field-oriented control of a standard ac motor using microprocessors," *IEEE Transactions on Industry Applications*, vol. IA-16, no. 2, pp. 186–192, 1980.

[4] J. Lara, J. Xu, and A. Chandra, "Effects of rotor position error in the performance of field-oriented-controlled pmsm drives for electric vehicle traction applications," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pp. 4738–4751, 2016.

[5] Z. Wang, J. Chen, M. Cheng, and K. T. Chau, "Field-oriented control and direct torque control for paralleled vsis fed pmsm drives with variable switching frequencies," *IEEE Transactions on Power Electronics*, vol. 31, no. 3, pp. 2417–2428, 2016.

[6] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with gaussian processes," in *Proc. of 2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 491–496.

[7] X. Li, H. Zhu, J. Ma, W. Wang, T. J. Teo, C. S. Teo, and T. H. Lee, "Learning-based high-precision tracking control: Development, synthesis, and verification on spiral scanning with a flexure-based nanopositioner," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 5, pp. 3867–3876, 2024.

[8] S. Müller, A. von Rohr, and S. Trimpe, "Local policy search with bayesian optimization," *Advances in Neural Information Processing Systems*, vol. 34, pp. 20 708–20 720, 2021.

[9] Y. Davidor, *Genetic algorithms and robotics: a heuristic strategy for optimization*. WORLD SCIENTIFIC, Jan. 1991.

[10] J. Mockus, *Bayesian approach to global optimization: theory and applications*. Springer Science & Business Media, 2012.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[12] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *Proc. of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 997–1005.

[13] Y. Sui, V. Zhuang, J. Burdick, and Y. Yue, "Stagewise safe bayesian optimization with gaussian processes," in *Proc. of the 35th International Conference on Machine Learning (ICML)*, Stockholm, Sweden, 2018, pp. 4781–4789.

[14] M. Turchetta, F. Berkenkamp, and A. Krause, "Safe exploration for interactive machine learning," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[15] A. Bottero, C. Luis, J. Vinogradska, F. Berkenkamp, and J. R. Peters, "Information-theoretic safe exploration with gaussian processes," *Advances in Neural Information Processing Systems*, vol. 35, pp. 30 707–30 719, 2022.

[16] C. Fiedler, J. Menn, L. Kreisköther, and S. Trimpe, "On safety in safe bayesian optimization," *Transactions on Machine Learning Research*, 2024.

[17] M. Fiducioso, S. Curi, B. Schumacher, M. Gwerder, and A. Krause, "Safe contextual bayesian optimization for sustainable room temperature pid control tuning," in *Proc. of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019, pp. 5850–5856.

[18] J. Djolonga, A. Krause, and V. Cevher, "High-dimensional gaussian process bandits," in *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[19] J. Kirschner, M. Mutny, N. Hiller, R. Ischebeck, and A. Krause, "Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces," in *Proc. of the 36th International Conference on Machine Learning (ICML)*, 2019, pp. 3429–3438.

[20] J. Kirschner, M. Mutnỳ, A. Krause, J. Coello de Portugal, N. Hiller, and J. Snuverink, "Tuning particle accelerators with safety constraints using bayesian optimization," *Physical Review Accelerators and Beams*, vol. 25, no. 6, p. 062802, 2022.

[21] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, "Safe-control-gym: A unified benchmark suite for safe learning-based control and reinforcement learning in robotics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 142–11 149, 2022.

[22] J. Rothfuss, C. Koenig, A. Rupenyan, and A. Krause, "Meta-learning priors for safe bayesian optimization," in *Proceedings of The 6th Conference on Robot Learning (CoRL)*, ser. Proceedings of Machine Learning Research, vol. 205, 14–18 Dec 2023, pp. 237–265.

[23] W. Wang, J. Ma, Z. Cheng, X. Li, C. W. d. Silva, and T. H. Lee, "Global iterative sliding mode control of an industrial biaxial gantry system for contouring motion tasks," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1617–1628, 2022.

[24] W. Wang, J. Ma, X. Li, H. Zhu, C. W. de Silva, and T. H. Lee, "Hybrid active–passive robust control framework of a flexure-joint dual-drive gantry robot for high-precision contouring tasks," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 2, pp. 1676–1686, 2023.

[25] D. K. Duvenaud, H. Nickisch, and C. Rasmussen, "Additive gaussian processes," in *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[26] P. Rolland, J. Scarlett, I. Bogunovic, and V. Cevher, "High-dimensional bayesian optimization via additive models with overlapping groups," in *Proc. of the Twenty-First International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018, pp. 298–307.

[27] K. Kandasamy, J. Schneider, and B. Poczos, "High dimensional bayesian optimisation and bandits via additive models," in *Proc. of the 32nd International Conference on Machine Learning (ICML)*, 2015, pp. 295–304.

[28] M. Mutny and A. Krause, "Efficient high dimensional bayesian optimization with additivity and quadrature fourier features," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[29] H. Wang, X. Li, W. Wang, A. Bhaumik, R. Xie, L. Zheng, and P. Vadakkepat, "Safety-aware controller optimization for a flexure-joint biaxial gantry robot," *IEEE/ASME Transactions on Mechatronics*, vol. 30, no. 4, pp. 3117–3124, 2025.

[30] M. Khosravi, C. König, M. Maier, R. S. Smith, J. Lygeros, and A. Rupenyan, "Safety-aware cascade controller tuning using constrained bayesian optimization," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 2, pp. 2128–2138, 2023.

[31] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[32] D. Baumann, K. Kowalczyk, C. R. Rojas, K. Tiels, and P. Wachel, "Safety and optimality in learning-based control at low computational cost," *IEEE Transactions on Automatic Control*, pp. 1–13, 2025.

[33] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: no regret and experimental design," in *Proc. of the International Conference on Machine Learning (ICML)*, 2010, pp. 1015–1022.

[34] S. R. Chowdhury and A. Gopalan, "On kernelized multi-armed bandits," in *International Conference on Machine Learning*. PMLR, 2017, pp. 844–853.

[35] M. W. Seeger, S. M. Kakade, and D. P. Foster, "Information consistency of nonparametric gaussian process methods," *IEEE Transactions on Information Theory*, vol. 54, no. 5, pp. 2376–2382, 2008.

[36] S. Vakili, K. Khezeli, and V. Picheny, "On information gain and regret bounds in gaussian process bandits," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 82–90.

[37] J. Gardner, G. Pleiss, R. Wu, K. Weinberger, and A. Wilson, "Product kernel interpolation for scalable gaussian processes," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2018, pp. 1407–1416.

[38] A. Wilson and R. Adams, "Gaussian process kernels for pattern discovery and extrapolation," in *International conference on machine learning*. PMLR, 2013, pp. 1067–1075.

[39] A. Bardou, P. Thiran, and T. Begin, "Relaxing the additivity constraints in decentralized no-regret high-dimensional bayesian optimization," in *ICLR '24: International Conference on Learning Representations (ICLR)*, 2024.

[40] L. Zheng, H. Wang, X. Li, J. Ma, and P. Vadakkepat, "Robotic control optimization through kernel selection in safe bayesian optimization," in *2024 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2024, pp. 2208–2214.

[41] Y. Wei, Z. Yi, H. Li, S. Soedarmadji, and Y. Sui, "Safe bayesian optimization for the control of high-dimensional embodied systems," in *Conference on Robot Learning*. PMLR, 2025, pp. 4771–4792.