Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-20 (2025) 835–840

Data-Driven Task-Specific Optimization of H_∞ Controllers via Bayesian Optimization

Marta Manzoni * **Marco Lovera** *

* *Dipartimento di Scienze e Tecnologie Aerospaziali, Politecnico di Milano, Via La Masa 34, 20156, Milano, Italy (e-mail: {marta.manzoni,marco.lovera}@polimi.it).*

Abstract: Tuning controller parameters is crucial for optimizing performance in dynamical systems. In this paper, we propose a two-step approach that combines a model-based method with a data-driven strategy for controller tuning. First, H_∞ synthesis is employed to design a controller that ensures conservative and satisfactory performance across a wide range of operating conditions. Then, safe Bayesian Optimization (safeBO) is used to fine-tune controller parameters based on experimental data to enhance performance for a specific task that exceeds the limitations of the H_∞ design. The approach is validated through simulations and experiments on a quadrotor, achieving a reduction of up to 60% in position tracking error, demonstrating its effectiveness in safe, efficient, and automated performance optimization.

Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Bayesian Optimization (BO); Autotuning; Gaussian process (GP); PID tuning; H_∞ control; Quadrotor UAV

1. INTRODUCTION

Tuning controller parameters is a key aspect of controller design, as it directly affects the performance, stability, and efficiency of dynamical systems. This tuning process can be performed using either model-based or data-driven methods, each with its advantages and limitations (Hou and Wang, 2013).

Structured H_∞ control is a powerful model-based method that has proven effective in various engineering applications, particularly in aerospace (Riccardi and Lovera, 2014). It provides a systematic framework for tuning controller parameters while explicitly defining performance requirements in the frequency domain. This allows designers to impose constraints on sensitivity and control effort, ensuring a well-balanced trade-off between performance and control objectives, while maintaining reliable system behavior across a broad range of operating conditions. This approach relies on an accurate mathematical model of the system, which can be obtained through first-principle methods or system identification. Although well-established techniques for system modeling and identification exist (Pintelon and Schoukens, 2012), they often rely on simplifying assumptions that may not fully capture the complexity of real-world systems. Furthermore, H_∞ control typically requires linearizing the system dynamics around an equilibrium point. As a result, it performs well close to this equilibrium, but may lead to overly conservative control actions when the system operates outside the linearized region, resulting in performance degradation. Consequently, manual fine-tuning is often necessary to adjust controller parameters for specific tasks with requirements that exceed the limitations of the H_∞ design. However, manual tuning is time-consuming, costly, and may still result in suboptimal performance.

With the growing availability of data, there has been a shift toward leveraging data-driven techniques to automate and improve the tuning process. These methods tune controller parameters based on experimental data, without needing an explicit model of the system. This makes them particularly useful when prior knowledge of the plant is limited or when developing an accurate model is challenging. Several data-driven approaches have been proposed, including inversion-based control (Novara and Formentin, 2017), virtual reference feedback tuning (Campi et al., 2002), and correlation-based tuning (Van Heusden et al., 2011). Although effective, these techniques may struggle to manage input/output constraints or select reference models without prior knowledge. A promising data-driven alternative is Bayesian Optimization (BO), a probabilistic method for efficient black-box optimization (Shahriari et al., 2015). It builds a probabilistic model of the objective function and selects promising evaluation points by balancing exploration and exploitation. In controller tuning, BO typically starts with an initial set of parameters that ensure stability and iteratively selects new values to optimize the objective based on experimental data (Marco et al., 2016; Berkenkamp et al., 2016). A key challenge in safety-critical applications is that BO may propose parameter values that deviate significantly from the initial ones, potentially resulting in unsafe behavior. To address this, safeBO incorporates safety constraints to keep the system within safe limits during optimization (Sui et al., 2015). For this approach to work, an initial set of parameters that satisfy these constraints is required. This safe set can be derived from prior knowledge, empirical tuning, or model-based design techniques, such as H_∞ synthesis.

Building on this discussion, we propose a two-step controller tuning approach. First, a baseline controller is designed using H_∞ synthesis, leveraging its strong theoreti-

2405-8963 Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2025.11.256

cal framework to ensure stability and provide conservative but satisfactory performance across a wide range of operating conditions. Next, we automatically refine the controller parameters using safeBO to optimize performance for a specific task that goes beyond the limitations of the H_∞ design, ensuring safe, efficient, and automated tuning. We validate our approach through simulations and real-world experiments on a quadrotor, showing that BO effectively fine-tunes controller parameters to meet task-specific requirements while ensuring safety.

The structure of the paper is as follows: Section 2 provides an overview of Gaussian Processes (GP) and safe BO. Section 3 outlines the proposed two-step controller tuning approach. Section 4 applies this method to tune the position controller of a quadrotor, providing both simulation and experimental results. Finally, Section 5 concludes the paper.

2. PRELIMINARIES

In this section, we provide an overview GPs and safe BO.

2.1 Gaussian Processes

A GP is a probabilistic, non-parametric model used to approximate unknown functions. It defines a distribution over functions, characterized by a mean function $\mu(\theta)$ and a kernel function $k(\theta, \theta')$, which captures correlation between different inputs. The function being modeled is denoted as $J(\theta) : \Theta \to \mathbb{R}$, where $\Theta \subseteq \mathbb{R}^d$ represents the space of admissible input parameters, and d is the dimensionality of the input space. The GP framework makes predictions about an unknown function from a finite set of noisy observations $\hat{J}(\theta_i) = J(\theta_i) + \nu$, $i = 1, \ldots, n$, where $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$ is a Gaussian noise. Given a dataset of n observations, $\mathcal{D}_n = \{\theta_i, \hat{J}(\theta_i)\}_{i=1}^n$, the posterior mean and variance at a new input $\theta^\star$ are given by:

$$\mu_n(\theta^\star) = k_n(\theta^\star)^\top (K_n + \sigma_\nu^2 I_n)^{-1} \hat{J}_n \tag{1}$$

$$\sigma_n^2(\theta^\star) = k(\theta^\star, \theta^{\star\prime}) - k_n(\theta^\star)^\top (K_n + \sigma_\nu^2 I_n)^{-1} k_n(\theta^\star), \tag{2}$$

where $k_n(\theta^\star) = [k(\theta^\star, \theta_1), \ldots, k(\theta^\star, \theta_n)]^\top$ is a vector of covariances between the new input $\theta^\star$ and all previously observed data points. $\hat{J}_n = [\hat{J}(\theta_1), \ldots, \hat{J}(\theta_n)]^\top$ is the vector of observed function values, $K_n \in \mathbb{R}^{n\times n}$ is the kernel matrix with entries $[K_n]_{(i,j)} = k(\theta_i, \theta_j)$, $i, j \in \{1, \ldots, n\}$, and $I_n \in \mathbb{R}^{n\times n}$ is an identity matrix.

This probabilistic framework provides both predictions and explicit uncertainty estimates via the posterior variance $\sigma_n^2(\theta)$, which captures epistemic uncertainty. This is crucial in safety-critical settings, enabling cautious decisions that account for uncertainties.

2.2 Safe Bayesian Optimization

BO is a powerful framework for optimizing an unknown function $J(\theta)$ given noisy observations, particularly when function evaluations are expensive, such as those requiring real-world experiments or high-fidelity simulations (Shahriari et al., 2015). Safe BO (Sui et al., 2015) is a variant of BO that aims to maximize $J(\theta)$ over the parameter space Θ, while ensuring that safety constraints $g_i(\theta) \geq 0$, where $g_i : \Theta \to \mathbb{R}$, for $i = 1, \ldots, m$, are satisfied. The constrained optimization problem is expressed as:

$$\max_{\theta\in\Theta} J(\theta) \quad \text{subject to} \quad g_i(\theta) \geq 0 \quad \forall i = 1, \ldots, m. \tag{3}$$

Assuming an initial safe set of parameters $\mathcal{S}_0 \subseteq \Theta$ that satisfies the constraints is available, safe BO explores the parameter space while maintaining safety. To address this problem, we employ the `SAFEOPT-MC` algorithm from Berkenkamp et al. (2023), an extension of `SAFEOPT` by Sui et al. (2015), specifically designed to handle multiple safety constraints. The full procedure is outlined in Algorithm 1.

Algorithm 1 `SAFEOPT-MC` algorithm (Berkenkamp et al., 2023).

Inputs: Input space Θ, GP prior $k((\theta, i), (\theta', j))$, initial safe set $\mathcal{S}_0$.
for $n = 1, \ldots, N_{max}$ **do**
  $\mathcal{S}_n \leftarrow \{\mathcal{S}_0 \cup \{\theta \in \Theta \mid \forall i \in \mathcal{I}_g : l_n^i(\theta) \geq 0\}\}$
  $\mathcal{M}_n \leftarrow \{\theta \in \mathcal{S}_n \mid u_n^J(\theta) \geq \max_{\theta'} l_n^J(\theta')\}$
  $\mathcal{G}_n \leftarrow \{\theta \in \mathcal{S}_n \mid e_n(\theta) > 0\}$
  $\theta_n \leftarrow \arg\max_{\theta\in\mathcal{G}_n\cup\mathcal{M}_n} \max_{i\in\mathcal{I}} \left(u_n^i(\theta) - l_n^i(\theta)\right)$
  Measurements: $\hat{J}(\theta), \hat{g}_i(\theta) \quad \forall i = 0, \ldots, m$
  Update GP with new data
end for=0

To model both the performance function J and the safety constraints g_i as a GP, `SAFEOPT-MC` accounts for multiple correlated functions using a surrogate model:

$$h(\theta, i) = \begin{cases} J(\theta) & \text{if } i = 0 \\ g_i(\theta) & \text{if } i \in \mathcal{I}_g, \end{cases} \tag{4}$$

which returns either the performance function or an individual safety constraint depending on the auxiliary input $i \in \mathcal{I} = \{0, \ldots, m\}$, where $\mathcal{I}_g = \{1, \ldots, m\} \subset \mathcal{I}$ are the indices for the constraints. This surrogate function is modeled as a GP over the extended parameter space $\Theta \times \mathcal{I}$, where $\mu_n(\theta, i)$ and $\sigma_n(\theta, i)$ are used to denote the predictions of the GP over the extended space. For instance, a joint kernel that captures correlations between the performance function and a single safety constraint is defined as:

$$k((\theta, i), (\theta', j)) = \begin{cases} \delta_{ij} k_J(\theta, \theta') + k_{Jg}(\theta, \theta') & \text{if } i = 0 \\ \delta_{ij} k_g(\theta, \theta') + k_{Jg}(\theta, \theta') & \text{if } i = 1, \end{cases} \tag{5}$$

where δ_{ij} is the Kronecker delta, k_J and k_g are independent kernels for $J(\theta)$ and $g(\theta)$, and k_{Jg} is a covariance function that captures shared components.

Instead of global optimization, `SAFEOPT-MC` restricts the search to the safe set $\mathcal{S} = \{\theta \in \Theta \mid g_i(\theta) \geq 0, \quad \forall i = 0, \ldots, m\}$, which is not known initially, but is estimated through function evaluations. To ensure safe exploration, the algorithm constructs high-probability confidence intervals for J and g_i:

$$Q_n(\theta, i) \colon = [\mu_{n-1}(\theta, i) \pm \beta_n^{1/2} \sigma_{n-1}(\theta, i)], \tag{6}$$

where β_n determines the desired confidence interval. The lower and upper bounds of this set are defined as $l_n^i(\theta) =$

$\min Q_n(\theta, i)$ and $u_n^i(\theta, i) = \max Q_n(\theta, i)$, respectively. Thus, the safe set is defined as:

$$\mathcal{S}_n = \mathcal{S}_0 \cup \left\{\theta \in \Theta \mid \forall i \in \mathcal{I}_g : l_n^i(\theta) \geq 0\right\}. \tag{7}$$

At each iteration, the algorithm selects the next sample θ_n by maximizing an acquisition function, which balances exploration (safe set expansion) and exploitation (performance maximization). The acquisition function identifies the most informative points to evaluate next by leveraging the mean and uncertainty predictions provided by the GP model in (1) and (2). To achieve this, the algorithm identifies two subsets: the potential maximizers $\mathcal{M}_n$, which may contain the optimal parameters, defined as $\mathcal{M}_n = \{\theta \in \mathcal{S}_n \mid u_n^J(\theta) \geq \max_{\theta' \in \mathcal{S}_n} l_n^J(\theta')\}$, and the potential expanders $\mathcal{G}_n$, which can safely expand the safe set, defined as $\mathcal{G}_n = \{\theta \in \mathcal{S}_n \mid e_n(\theta) > 0\}$, where $e_n(\theta)$ is an optimistic indicator function for expansion. Finally, the next evaluation point is selected from $\mathcal{G}_n \cup \mathcal{M}_n$ based on the highest uncertainty:

$$\theta_n = \underset{\theta \in \mathcal{G}_n \cup \mathcal{M}_n}{\arg\max} \max_{i \in \mathcal{I}} \left(u_n^i(\theta) - l_n^i(\theta)\right). \tag{8}$$

The optimization process continues until the maximum number of iterations N_{max} is reached or a predefined convergence criterion is met. Starting from the initial safe set $\mathcal{S}_0$, this process identifies the optimum of (3) within the "reachable" region, which corresponds to the portion of the parameter space that can be safely explored from $\mathcal{S}_0$ without violating the safety constraints (Sui et al., 2015).

3. TWO-STEP CONTROLLER TUNING APPROACH

This section presents our two-step approach for tuning a controller for a generic nonlinear system, expressed as $\dot{x}(t) = f(x(t), u(t))$, where x is the state of the system and u is the control input. The procedure has two phases:

(1) First, H_∞ synthesis is employed to design a baseline controller that delivers conservative yet satisfactory performance, establishing an initial safe set of controller parameters.
(2) Next, the safeBO procedure is initialized with these safe parameters, which are then refined to improve performance for a specific task.

3.1 Step 1: H_∞ synthesis for baseline controller tuning

The first step in structured H_∞ synthesis involves linearizing the nonlinear system at an equilibrium point:

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + B\tilde{u}(t), \tag{9}$$

where the matrices A and B are typically obtained through system identification techniques. Next, a structured controller $K(\theta)$ parameterized by $\theta \in \Theta$ is defined. Given the linearized model and controller, the control design problem is formulated to minimize the H_∞ norm of the closed-loop transfer function while ensuring stability and performance specifications:

$$\min_{\theta \in \Theta} \|T_{w \to z}(P, K(\theta))\|_\infty, \tag{10}$$

where $P(s)$ is the plant transfer matrix, combining process and filter functions to define performance and robustness requirements, and $T_{w\to z}(K)$ represents the closed-loop transfer function from disturbances w to performance outputs z. This non-convex, non-smooth optimization problem can be addressed with methods like non-smooth optimization (Apkarian and Noll, 2006) or randomized approaches (Tempo et al., 2013).

The H_∞ synthesis process determines locally optimal controller parameters subject to frequency-domain performance specifications. These specifications are imposed through weighting functions on the sensitivity $W_S(s)$ and control sensitivity $W_Q(s)$, which shape the system to achieve satisfactory performance while limiting control effort. The inverses of these functions define upper bounds on the corresponding frequency response magnitudes, enforcing constraints on low-frequency gain (DC gain), high-frequency attenuation (HF gain), and bandwidth (crossover frequency ω_{cross}).

At this point, we have determined a baseline set of controller parameters, θ_{BL}, that ensures stable and safe system operation, while providing conservative yet satisfactory performance.

3.2 Step 2: Safe BO for task-specific controller refinement

The second step of our proposed two-step tuning procedure aims to fine-tune the baseline controller parameters, θ_{BL}, to better suit task-specific requirements that may go beyond the limitations of the initial H_∞ design. In such cases, the H_∞ controller can result in overly conservative actions, leading to suboptimal performance.

At this stage, the user specifies the task the system must perform, along with a performance objective and any relevant constraints tailored to that task. Unlike H_∞ synthesis, which relies on an explicit linear system model, BO treats the system as a black box (potentially nonlinear). It leverages input-output data from real-world experiments or simulations to iteratively guide the optimization. The task-specific controller refinement is therefore formulated as the following constrained optimization problem:

$$\max_{\theta \in \Theta} J(x(\theta, \tau), u(\theta, \tau)) \tag{11}$$

$$\text{s.t.} \quad g_i(x(\theta, \tau), u(\theta, \tau)) \geq 0 \quad \forall \quad i = 1, \ldots, m, \tag{12}$$

where τ represents the task specification (e.g., a desired reference trajectory). The cost function $J(x(\theta, \tau), u(\theta, \tau))$ depends on system states x and control inputs u, both of which are functions of the controller parameters θ and the task specification τ. This function quantifies system performance during task execution, such as tracking error, energy consumption, or other relevant metrics. The constraints $g_i(x(\theta, \tau), u(\theta, \tau))$ enforce safety limits, such as bounds on states or control inputs.

This optimization problem is solved using the `SAFEOPT-MC` procedure outlined in Algorithm 1. The controller parameters of the H_∞ synthesis define the initial safe set, $\mathcal{S}_0 = \theta_{BL}$, ensuring that constraints are satisfied at the beginning of optimization. By iteratively refining the controller parameters, task-specific performance is improved while maintaining safety. The process continues until convergence or until a given evaluation budget is reached. Since BO is employed to optimize costly-to-evaluate functions, the process is usually constrained by an optimization budget, which limits the number of function evaluations.

After optimization, the new controller parameters, θ_{BO}, can be analyzed to assess their impact on system stability. Given that the system model used for the H_∞ design is available, the new gain margin and phase margin can be computed. Alternatively, if explicit stability conditions are known, they can be added as constraints in the optimization; otherwise, stability can be encouraged through penalty terms in the cost function.

4. SIMULATION AND EXPERIMENTAL RESULTS

In this section, we evaluate the effectiveness of the proposed two-step approach for tuning the position controller of an ANT-X[1] fixed-pitch quadrotor.

4.1 Quadrotor control scheme

The dynamics of the quadrotor are characterized by 12 states: positions in the North-East-Down (NED) inertial frame (x, y, z), velocities (v_x, v_y, v_z), ZYX Euler angles (ψ, θ, ϕ), and angular velocities in the body frame (p, q, r).

PID controllers are commonly employed for quadrotor position control because of their simplicity and ease of implementation. To simplify the analysis, we focus on tuning the controller for the x- and y- directions, assuming the controllers for the z-direction and yaw are predefined and fixed. Due to the symmetric nature of the platform, the controller for the y-direction is identical to the one used for the x-direction, and both controllers consist of two main components: the position-velocity controller and the attitude-rate controller, as shown in Figure 1. The position-velocity controller is a cascaded loop, where the position control loop employs a proportional controller, $C_p(s) = K_p$, and the velocity control loop uses a PID controller, $C_v(s) = K_v + K_{iv}/s + K_{dv}s$. The velocity controller then provides the reference for the attitude-rate controller.

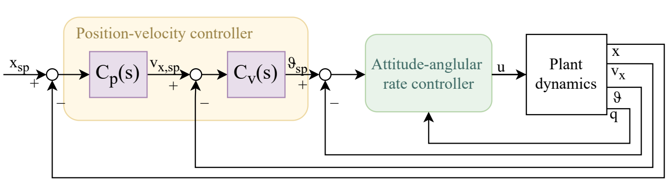


Fig. 1. Controller structure.

In this work, we focus on tuning the parameters of the position and velocity controllers along the x- and y- axes, denoted as $C_p(s)$ and $C_v(s)$, respectively, while assuming that the attitude and rate control loops are fixed. These controllers are parameterized by $\theta = \{K_p, K_v, K_{iv}, K_{dv}\}$.

4.2 Two-step controller tuning for quadrotor

1st step: H_∞ synthesis. An accurate system model is essential for H_∞ synthesis. To achieve this, we first linearize the model of the quadrotor around a reference steady-flight condition. Next, we identify a grey-box model for the longitudinal dynamics using the output-error method. With the linearized model and the controller structure from Section 4.1, the locally optimal controller parameters θ_{BL} are determined to achieve two objectives:

- Performance: Defined by a weighting function on the position sensitivity, $W_S(s)$, which represents the transfer function from the desired x-axis position x_d to the position error $e_x := x - x_d$.
- Control moderation: Defined by a weighting function on the control sensitivity, $W_Q(s)$, representing the transfer function from x_d to the control input M_x.

The parameters used to construct these weighting transfer functions, with poles and zeros following a Butterworth pattern, are summarized in table 1. In this case, we applied a non-smooth optimization approach to solve the H_∞ control design problem. The chosen control synthesis tool is the hinfstruct function from the Matlab Robust Control Toolbox. The resulting controller gain are: $\theta_{BL} = \{0.902, 0.153, 0.215, 0.0109\}$.

Table 1. Parameters for the structured H_∞ synthesis weighting functions.

	DC gain	HF gain	ω_{cross} [rad/s]	order
W_S	10^{-5}	2.5	1	1
W_Q	10^{-5}	3	3	3

2nd step: Safe Bayesian Optimization. The quadrotor is tasked with executing a circular trajectory defined by the reference position $p_d(t) = [R\sin(\Omega t), R\cos(\Omega t), -1]^\top$ m, where $R = 1$ m is the radius and $\Omega = 2$ rad/s is the angular velocity. This maneuver represents an aggressive flight path, involving high angular velocities and significant attitude changes that exceed the limitations of the initial H_∞ design.

The parameter space is defined as $\Theta = [0, 2] \times [0, 0.3] \times [0, 0.3] \times [0, 0.012]$. To explore this space, we generate 25 uniformly distributed samples per interval, resulting in a total of 390'625 possible parameter combinations. The controller parameters obtained from H_∞ synthesis are used to start the optimization procedure, specifically $\mathcal{S}_0 = \theta_{BL} = \{0.902, 0.153, 0.215, 0.0109\}$.

For this position-tracking task, we define a cost function that penalizes deviations from the reference trajectory:

$$J(\theta) = -\frac{w_p}{N}\sum_{i=1}^{N}\left((x_i - x_{d_i})^2 + (y_i - y_{d_i})^2\right), \qquad (13)$$

where $w_p = 500$. To ensure the safety of the quadrotor during performance evaluation, we enforce a constraint on its angular velocity. Specifically, we limit the maximum angular velocity around the x- and y-axes to $\omega_{max} = 10.5$ rad/s, reflecting the actual physical limits of the ANT-X quadrotor used in this study. This safety requirement is enforced through the constraint $g(\theta) = \omega_{max} - \max(\max_i |p_i|, \max_i |q_i|)$. At each iteration of the `SAFEOPT-MC` algorithm, the cost function and safety constraint are evaluated by running Simulink simulations of the quadrotor, providing the input-output data needed to guide the optimization process.

[1] https://antx.it/

Safe BO models both the cost function J and the safety constraint g as a GP. Without loss of generality, we assume zero-mean priors, *i.e.*, $\mu_J(\theta) = 0$ and $\mu_g(\theta) = 0$. The kernel functions $k_J(\theta, \theta')$ and $k_g(\theta, \theta')$ are problem-specific and encode prior knowledge about the objective and constraints. We selected the Matérn kernel with $\nu = 3/2$, which has proven effective for noisy, differentiable functions, combined with automatic relevance determination to assign individual length scales to each input dimension.

$$k(\theta, \theta') = \sigma_f^2 \left(1 + \sqrt{3}\, r(\theta, \theta')\right) \exp\left(-\sqrt{3}\, r(\theta, \theta')\right) \quad (14)$$

$$r(\theta, \theta') = (\theta - \theta')^T M^{-2} (\theta - \theta'). \quad (15)$$

This kernel is parameterized by three hyperparameters: measurement noise σ_ν^2, process variation σ_f^2 , and length-scales, $l \in \mathbb{R}_+^{|\Theta|}$, which form the diagonal elements of the matrix M. To improve the accuracy of the GP model, these hyperparameters are optimized by maximizing the marginal likelihood. In practice, an initial dataset $\mathcal{D}_0$ of 35 samples, obtained via Latin Hypercube Sampling, is used to estimate these hyperparameters.

The `SAFEOPT-MC` algorithm is then applied iteratively to refine the controller parameters. With an optimization budget of $N_{max} = 30$ experiments, the final gains after 30 iterations are: $\theta_{BO} = \{1.833, 0.288, 0.013, 0.005\}$.

4.3 Results

Controller performance evaluation. We assess the effectiveness of the proposed two-step approach on circular trajectories with varying angular velocities in both simulations and real-word experiments. The controller tracking performance is measured using the position Root Mean Square Error (RMSE) as an evaluation metric:

$$\text{RMSE} = \sqrt{\frac{1}{M} \sum_{k=1}^{M} \|p_k - p_{d_k}\|^2}, \quad (16)$$

where $p_k = (x_k, y_k)$ is the actual position, $p_{d_k} = (x_{d_k}, y_{d_k})$ is the desired position, and M is the size of dataset collected during flights.

Experimental setup. Flight tests are conducted within the Flying Arena for Rotorcraft Technologies (FlyART) facility at Politecnico di Milano, equipped with a Motion Capture (Mo-Cap) system. The drone is a fixed-pitch quadrotor designed by ANT-X [2] . Data is collected by the Mo-Cap system, which consists of 12 cameras tracking markers attached to the drone. These measurements are transmitted to a ground station, which then sends them to the drone at a rate of 100Hz, along with the trajectory instructions defined by specific waypoints at 50Hz.

Simulation and experimental results. Figure 2 shows the tracking performance of the baseline H_∞ controller in a real-world experiment, where the quadrotor follows a circular trajectory with $R = 1$ m and $\Omega = 1$ rad/s. In this scenario, the quadrotor operates within the linear dynamics assumed in H_∞ synthesis. As a result, the controller demonstrates satisfactory tracking performance.

[2] https://antx.it/educational-products-antx/dronelab-antx/

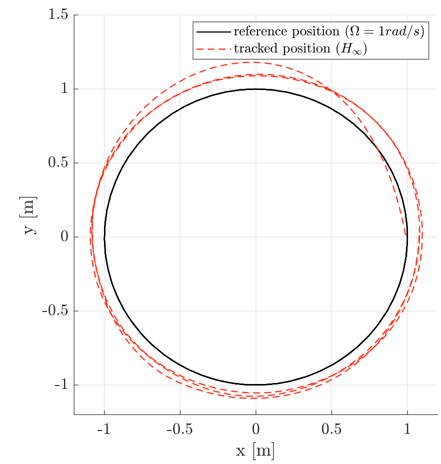


Fig. 2. Tracking performance of the H_∞ controller for a circular trajectory with $R = 1$ m and $\Omega = 1$ rad/s in a real-word experiment.

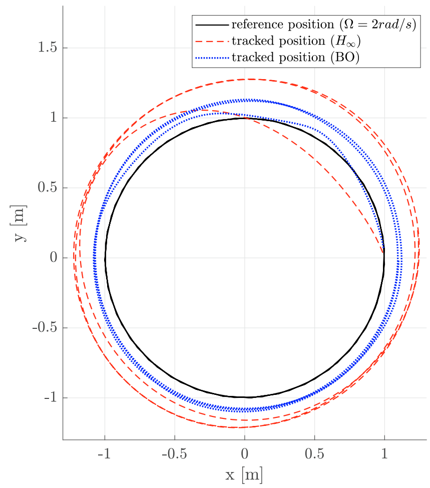
Table 2. RMSE for circular trajectories at varying angular velocities.

Ω [rad/s]	Simulation RMSE [m]		Simulation % ↓	Experiment RMSE [m]		Experiment % ↓
	H_∞	BO		H_∞	BO	
1	0.073	–	–	0.100	–	–
2	0.205	0.073	61.3%	0.376	0.148	60.7%
2.5	0.259	0.124	52.4%	0.475	0.217	54.3%

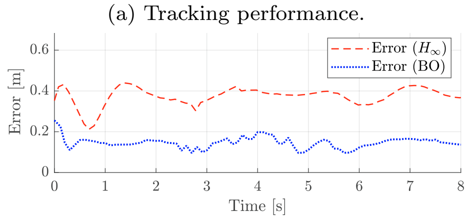
For the BO-optimized controller, we evaluate performance on two circular trajectories with the same radius ($R = 1$ m) but higher angular velocities: $\Omega = 2$ rad/s and $\Omega = 2.5$ rad/s. The $\Omega = 2$ rad/s trajectory, shown in Figure 3a, was used for training in simulations, while the $\Omega = 2.5$ rad/s trajectory tests the ability of the controller to generalize beyond its training conditions while remaining in a similar regime. During these trajectories, the quadrotor exceeds the linear dynamics assumed in H_∞, resulting in higher angular velocities and significant attitude changes. Consequently, the H_∞ controller struggles to accurately track the high-speed trajectories. In contrast, the BO-optimized controller achieves significantly lower position tracking errors, as shown in Figure 3b. The RMSE values in table 2 further highlight this improvement, showing error reductions of up to 60% in both simulations and experiments. Notably, the BO-optimized controller maintains superior tracking performance not only on the training trajectory ($\Omega = 2$ rad/s) but also on a different trajectory beyond its training conditions ($\Omega = 2.5$ rad/s), while remaining in a similar operational regime.

Table 3. Gain and phase margins of the open-loop systems.

Controller	Gain margin [dB]	Phase margin [deg]
H_∞	22.5	57.2
BO	16.2	61.6



(a) Tracking performance.



(b) Position tracking error norm.

Fig. 3. Tracking performance (a) and error (b) for the H_∞ controller and the BO controller on a circular trajectory with $R = 1$ m and $\Omega = 2$ rad/s in real-word experiments.

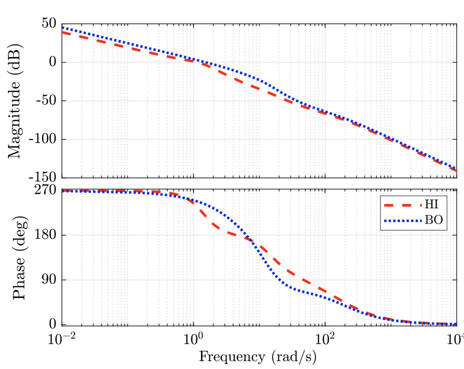


Fig. 4. Bode plot of the open-loop systems.

Figure 4 shows the Bode plots of the open-loop system for both controllers, with gain and phase margin values summarized in table 3. Despite a slight reduction in gain margin, the BO-optimized controller maintains stability within acceptable limits.

5. CONCLUSIONS

In this paper, we proposed a two-step controller tuning approach combining structured H_∞ synthesis with safe Bayesian Optimization. The H_∞ synthesis guarantees stability and reliable performance across a wide range of operating conditions, while safe Bayesian Optimization refines controller parameters to improve task-specific performance with safety guarantees at each iteration. Simulations and experiments on a quadrotor demonstrate the effectiveness of the method in achieving safe, efficient, and automated performance optimization, reducing position tracking error by up to 60%. Future work will focus on improving generalization via multi-task BO and overcoming scalability challenges by leveraging high-dimensional BO techniques capable of tuning controllers with larger parameter sets.

REFERENCES

Apkarian, P. and Noll, D. (2006). Nonsmooth H_∞ synthesis. *IEEE Transactions on Automatic Control*, 51(1), 71–86.

Berkenkamp, F., Krause, A., and Schoellig, A.P. (2023). Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics. *Machine Learning*, 112(10), 3713–3747.

Berkenkamp, F., Schoellig, A.P., and Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *2016 IEEE international conference on robotics and automation (ICRA)*, 491–496. IEEE.

Campi, M.C., Lecchini, A., and Savaresi, S.M. (2002). Virtual reference feedback tuning: a direct method for the design of feedback controllers. *Automatica*, 38(8), 1337–1346.

Hou, Z.S. and Wang, Z. (2013). From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235, 3–35.

Marco, A., Hennig, P., Bohg, J., Schaal, S., and Trimpe, S. (2016). Automatic LQR tuning based on Gaussian process global optimization. In *2016 IEEE international conference on robotics and automation (ICRA)*, 270–277. IEEE.

Novara, C. and Formentin, S. (2017). Data-driven inversion-based control of nonlinear systems with guaranteed closed-loop stability. *IEEE Transactions on Automatic Control*, 63(4), 1147–1154.

Pintelon, R. and Schoukens, J. (2012). *System identification: a frequency domain approach*. John Wiley & Sons.

Riccardi, F. and Lovera, M. (2014). Robust attitude control for a variable-pitch quadrotor. In *2014 IEEE Conference on Control Applications (CCA)*, 730–735. IEEE.

Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., and De Freitas, N. (2015). Taking the human out of the loop: a review of Bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Sui, Y., Gotovos, A., Burdick, J., and Krause, A. (2015). Safe exploration for optimization with Gaussian processes. In *International conference on machine learning*, 997–1005. PMLR.

Tempo, R., Calafiore, G., Dabbene, F., et al. (2013). *Randomized algorithms for analysis and control of uncertain systems: with applications*, volume 7. Springer.

Van Heusden, K., Karimi, A., and Bonvin, D. (2011). Data-driven model reference control with asymptotically guaranteed stability. *International Journal of Adaptive Control and Signal Processing*, 25(4), 331–351.