ISA Transactions xxx (xxxx) xxx

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatra

A general surrogate-based optimization algorithm for problems with computationally-expensive constraints with application to μ-synthesis control design

Davide Previtali, Mirko Mazzoleni*, Nicholas Valceschini, Fabio Previdi

University of Bergamo, Department of Management, Information and Production Engineering, Via G. Marconi 5, 24044, Dalmine (BG), Italy

HIGHLIGHTS

- The paper focuses on SISO LTI passive fault-tolerant control systems that should be robust against multiplicative faults, modeled as parametric uncertainties.
- The μ-synthesis tool is adopted for control design. In particular, we aim to automatically define the performance weight W_p that dictates the closed-loop performance based on the sensitivity function of the closed-loop system.
- The optimization problem for the automatic definition of W_p is a nested optimization problem that includes a μ-synthesis constraint. The problem is characterized by a computationally-light cost and a set of constraints, where only the μ-synthesis one is computationally-expensive. Thus, we propose a novel surrogate-based optimization scheme able to effectively and efficiently handle such problems. A proof of convergence is provided.
- Numerical Monte Carlo results show the benefits of the proposed approach with respect to two other global optimization schemes.

ARTICLE INFO

Keywords:
Passive fault-tolerant control
Robust control
Surrogate-based optimization

ABSTRACT

The design of robust and passive fault-tolerant controllers can be performed with several approaches, such as μ-synthesis. These control design tools can also include manually set weight functions for different control performance specifications. Although the literature already includes automatic tuning methods for such weights, these methods involve a nested optimization problem whose resolution is hindered by the presence of local minima and the evaluation of a computationally-expensive constraint. In this paper, we propose a novel surrogate-based global optimization algorithm to overcome the aforementioned limitations, also demonstrating its convergence. The effectiveness and efficiency of our proposal are compared against a local optimization method with multi-start and a global genetic algorithm, showing superior performance on both fronts. The engineering importance of the proposed algorithm is motivated by the design of passive fault-tolerant controllers that are robust to multiplicative faults, modeled as parametric uncertainties on the plant. In particular, we focus on the automatic tuning of the performance weight on the sensitivity function in the μ-synthesis control design procedure. Numerical results are provided to assess the susceptibility with respect to the order of the plant and the uncertainty level. The robustness properties of the designed controller are also evaluated on a simulated mechatronic system subject to a multiplicative fault.

1. Introduction

The *μ-synthesis* method is an optimization-based approach for robust control design in the face of structured and unstructured uncertainty when a measure of closed-loop performance can be expressed in terms of the $\mathcal{H}_\infty$-norm of (possibly) weighted closed-loop transfer functions [38, Chapter 10], [33, Chapter 8]. However, suitable weights are in general *manually* tuned by the control designer.

Research for the *automatic* (and optimal in a certain sense) tuning of performance weights has shown promising results for μ-synthesis [19, 20] and other mixed-sensitivity control design methods [29,35]. In particular, the authors of [20] propose a *nested optimization problem* for

* Corresponding author.
Email addresses: davide.previtali@unibg.it (D. Previtali), mirko.mazzoleni@unibg.it (M. Mazzoleni), nicholas.valceschini@unibg.it (N. Valceschini), fabio.previdi@unibg.it (F. Previdi).

https://doi.org/10.1016/j.isatra.2026.03.047
Received 6 October 2025; Received in revised form 30 March 2026; Accepted 30 March 2026
Available online 2 April 2026

tuning the performance weights subject to a (relaxed) *μ-synthesis constraint* that is solved by a modified DK-iteration procedure [12]. Still, especially for mixed real and complex perturbations for which devised DK-iteration-like schemes were proposed [2,13,37], the *resolution of this μ-synthesis constraint is computationally-expensive* (i.e., time-consuming to evaluate) and not simultaneously convex in all the optimization variables [1]. *Global* optimization algorithms [34] should thus be investigated for *efficiently* (as in minimizing the number of computationally intensive constraint evaluations) and *effectively* (as in finding accurate solutions to the optimization problem) solving the performance weights optimization problem subject to the μ-synthesis constraint.

Popular global derivative-free optimization procedures include `DIRECT` [17], Particle Swarm Optimization (PSO) [18], and genetic algorithms [4]. Alternatively, local derivative-based optimization procedures, such as interior-point methods [25], can also be used for global optimization purposes by running the local solver multiple times from different starting points,[1] leading to the so-called multi-start methods [22]. Nonlinear constraints (such as the μ-synthesis one) can be handled via penalty [25] or barrier functions [9]. Nonetheless, the cited global optimization methods are unsuited for problems involving computationally-expensive cost and/or constraint functions, as they typically require a *large* number of evaluations of these functions to attain accurate solutions.

Several global optimization methods were proposed for solving optimization problems whose cost function and (possibly) constraint functions are expensive to evaluate, giving rise to the so-called *Black-Box*[2] *Optimization (BBO)* framework [16,30,36], [27, Chapter 2]. Most BBO algorithms belong to the family of *Surrogate-Based Methods (SBMs)*, which approximate the computationally-expensive cost and constraint functions with surrogate models that are cheap to evaluate. The goal of SBMs is to *find a sufficiently accurate global solution of an optimization problem while evaluating the fewest possible tunings of the decision variables*, achieving a good balance between effectiveness and efficiency. To that end, SBMs iteratively propose new calibrations to try by trading off *exploitation*, i.e., using the surrogate models as proxies for the time-consuming functions, and *exploration*, i.e., promoting the evaluation of tunings in those regions of the decision space where less information is available. This iterative process is stopped once a maximum number of calibrations (i.e., the *budget*) is reached. To the best of the authors' knowledge, there does not exist a BBO algorithm in the literature that addresses the case where the cost function is cheap to evaluate, making the construction of its surrogate unnecessary, while only some constraints are computationally-expensive.

1.1. Contributions

This paper proposes a global surrogate-based optimization algorithm specifically designed to handle optimization problems with a *computationally-light cost function* and a set of *computationally-expensive constraint functions*. A relevant instance of such problems is the one in [20], which considers the automatic tuning of performance weights in the context of μ-synthesis, with subsequent design of a robust controller with maximally achievable closed-loop performance. Different from [20], the methodology proposed in this paper *does not require a reformulation of the optimization problem to make its resolution computationally tractable*. Specifically, as a practical engineering application, we focus on maximizing the performance of Passive Fault-Tolerant (e.g., robust) Controllers (PFTCs) [7] for Single-Input Single-Output Linear Time-Invariant (SISO LTI) plants subject to multiplicative faults modeled as parametric variations, assuming that the possible values of the parameters following a multiplicative fault lie within a known bounded range.[3]

1.2. Paper organization

The remainder of the paper is as follows. Section 2 provides preliminary notions about plant models with uncertain parameters and the μ-synthesis framework with automatic performance weight selection. Section 3 presents the proposed surrogate-based optimization algorithm. A proof of its convergence is provided in Appendix B. A comparative analysis on numerical case studies of the proposed optimization algorithm with respect to genetic and multi-start global optimization algorithms is given in Section 4, along with a closed-loop evaluation of the PFTC properties of the designed controller for a three-mass mechatronic system. The paper ends with some concluding remarks in Section 5. Finally, Appendix A presents a supplementary empirical study on the hyperparameters of our proposal.

1.3. Notation

Let $\mathbb{C}$, $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote the sets of complex, real, integer, and natural numbers, respectively, with $0 \in \mathbb{N}$. $\mathbb{R}_{>0}$ and $\mathbb{R}_{\geq 0}$ stand for the sets of positive and non-negative real numbers, respectively. The imaginary unit is denoted by j. Unless otherwise stated, scalar variables x are denoted with lowercase letters, vectors $\boldsymbol{x}$ with bold lowercase letters, matrices X with uppercase letters, and sets $\mathcal{X}$ with uppercase calligraphy letters. Transfer functions and transfer matrices $X(s)$ are represented in uppercase with $s \in \mathbb{C}$ being the Laplace variable. Signals are denoted as $x(t)$ with $t \in \mathbb{R}_{\geq 0}$ (in seconds, s) being the continuous-time variable. The $n \times n$, $n \in \mathbb{N}$, identity matrix is denoted as I_n while the n-dimensional vector of zeros is denoted as $\mathbf{0}_n$. The symbol $\|\cdot\|_\infty$ indicates the ∞-norm of signals or vectors, and the $\mathcal{H}_\infty$-norm of transfer functions and transfer matrices. Meanwhile, $\|\cdot\|_2$ denotes the 2-norm of signals or vectors, and the $\mathcal{H}_2$-norm of transfer functions and transfer matrices. Diagonal stacking of two matrices X, Y is denoted as $\mathrm{diag}(X, Y)$. Infinite sequences are denoted as $\langle \cdot \rangle_{i \geq 1}$.

2. Preliminaries

In the following, Section 2.1 presents the considered plant model with parametric uncertainties. This description is used to represent multiplicative faults that may occur in the plant. Then, Section 2.2 reviews the μ-synthesis approach to robust control design, while Section 2.3 motivates the proposed optimization algorithm for solving the computationally-expensive μ-synthesis problem with automatic performance weight optimization.

2.1. Plant model

We consider a SISO LTI plant model with input $u(t)$ and output $y(t)$ subject to *multiplicative faults* (i.e., variations of plant parameters) represented as structured (parametric) uncertainties [33, Chapter 7]. Let $g \in \mathbb{Z}$ be the system type, $n_{z,\mathbb{R}}, n_{z,\mathbb{C}} \in \mathbb{N}$ be the number of real zeros and the number of pairs of complex conjugate zeros, respectively, and $n_{p,\mathbb{R}}, n_{p,\mathbb{C}} \in \mathbb{N}$ be the number of real poles and the number of pairs of complex conjugate poles, respectively. Then, the *perturbed plant model*

[1] In this paper, we use the terms "point", "sample", "calibration", and "tuning" interchangeably when referring to a specific set of values for the decision variables of an optimization problem.

[2] As a matter of fact, in the relevant literature, a black-box function (such as a cost function or a constraint function) is a function that is unknown (in the sense that no analytical formulation is available) and/or computationally-expensive to evaluate.

[3] In this paper, the difference between PFTC and robust control is merely a matter of interpretation about the origin of the uncertainty in the plant. In robust control, parameter variations follow epistemic uncertainty about the model. Instead, in PFTC, parameter variations are caused by multiplicative faults.

$G_p(s)$ amounts to:

$$G_p(s) := \frac{\lambda_p}{s^g} \frac{\prod_{i=1}^{n_{z,\mathbb{R}}} (\varphi_{i,p} s + 1)}{\prod_{i=1}^{n_{p,\mathbb{R}}} (\tau_{i,p} s + 1)} \frac{\prod_{i=1}^{n_{z,\mathbb{C}}} \left(\frac{s^2}{\alpha_{i,p}^2} + 2\frac{\zeta_{i,p}}{\alpha_{i,p}} s + 1 \right)}{\prod_{i=1}^{n_{p,\mathbb{C}}} \left(\frac{s^2}{\beta_{i,p}^2} + 2\frac{\xi_{i,p}}{\beta_{i,p}} s + 1 \right)}, \tag{1}$$

where the subscript p indicates a perturbation, either in the plant model (i.e., $G_p(s)$), or in the parameters (e.g., λ_p). In particular, the perturbed plant parameters in (1) are the gain $\lambda_p \in \mathbb{R}_{>0}$, the time constants of the real zeros and poles, i.e., $\varphi_{i,p} \in \mathbb{R}_{>0}$ and $\tau_{i,p} \in \mathbb{R}_{>0}$ (in s), the natural frequencies of the complex conjugate zeros and poles, i.e., $\alpha_{i,p} \in \mathbb{R}_{>0}$ and $\beta_{i,p} \in \mathbb{R}_{>0}$ (in $\frac{\text{rad}}{\text{s}}$), and the damping ratios $\zeta_{i,p}, \xi_{i,p} \in [0, 1)$. Each parameter $\rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{p,\mathbb{C}}}\}$ is bounded within a region $[\rho_{\min}, \rho_{\max}]$, $\rho_{\min}, \rho_{\max} \in \mathbb{R}_{>0}$, leading to the uncertain parameter set [33, Chapter 7]:

$$\rho_p := \bar{\rho}\left(1 + r_\rho \delta_\rho\right) \quad \text{for any } \delta_\rho \in \mathbb{R}, \left|\delta_\rho\right| \le 1, \tag{2}$$

where $\bar{\rho}$ is the *mean parameter value*, $r_\rho := \frac{\rho_{\max} - \rho_{\min}}{\rho_{\max} + \rho_{\min}}$ is the *relative uncertainty* in the parameter, and δ_ρ is its *perturbation*. Thus, we can define the set of considered perturbed systems as:

$$\mathcal{G} := \{G_p(s) \text{ in (1): } \delta_\rho \in \mathbb{R}, \left|\delta_\rho\right| \le 1, \forall \rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{p,\mathbb{C}}}\}\}. \tag{3}$$

Further, the nominal ("average") plant model $G(s) \in \mathcal{G}$ with no perturbation amounts to:

$$G(s) := \frac{\bar{\lambda}}{s^g} \frac{\prod_{i=1}^{n_{z,\mathbb{R}}} (\bar{\varphi}_i s + 1)}{\prod_{i=1}^{n_{p,\mathbb{R}}} (\bar{\tau}_i s + 1)} \frac{\prod_{i=1}^{n_{z,\mathbb{C}}} \left(\frac{s^2}{\bar{\alpha}_i^2} + 2\frac{\bar{\zeta}_i}{\bar{\alpha}_i} s + 1 \right)}{\prod_{i=1}^{n_{p,\mathbb{C}}} \left(\frac{s^2}{\bar{\beta}_i^2} + 2\frac{\bar{\xi}_i}{\bar{\beta}_i} s + 1 \right)}. \tag{4}$$

As an example, following a physical fault, the description (1)–(4) can be used to represent the degradation of an actuator (e.g., the value of a pole time constant increases or the gain decreases), or to approximate the introduction of a time delay $\tau_d \in \mathbb{R}_{>0}$ by augmenting (1) with a Padé approximation [26, Chapter 2], treating τ_d as an uncertain parameter bounded within a region $[\tau_{d_{\min}}, \tau_{d_{\max}}]$, where $\tau_{d_{\min}} \to 0^+$ (approximately no delay) and $\tau_{d_{\max}}$ is the maximum time delay due to a fault. A practical example is also given in Section 4.5, where we model the degradation of a component of a three-mass mechatronic system using parametric uncertainties.

2.2. The μ-synthesis framework with performance weight optimization

We consider the design of a controller $K(s)$ that guarantees *Robust Performance (RP)* for a generalized LTI plant $P(s)$ comprising (1) and a *performance weight* $W_P(s)$ encoding performance specifications on the *sensitivity function* (e.g., control bandwidth or disturbance rejection)

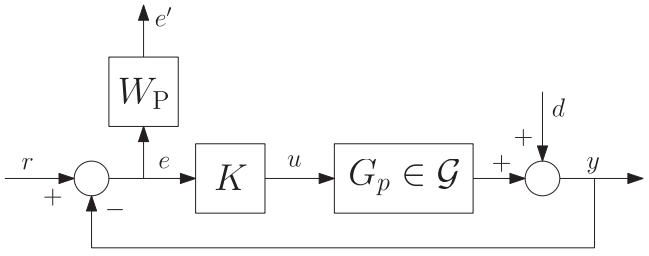


Fig. 1. Considered closed-loop scenario for μ-synthesis. $K(s)$ is a feedback controller, $G_p(s)$ is a system affected by parametric uncertainty as in (1)–(4), $W_P(s)$ is a weight function that encodes performance specifications by weighing the sensitivity function of the closed-loop system. The reported signals are the reference $r(t)$, measured output $y(t)$, tracking error $e(t)$, control input $u(t)$, disturbance $d(t)$, and fictitious error $e'(t)$ arising from $W_P(s)$.

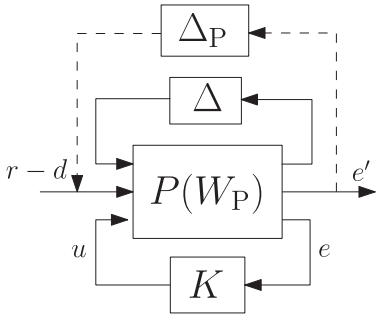


Fig. 2. General configuration for robust control analysis and synthesis. $P(s; W_P)$ is the generalized plant that considers a transfer function $W_P(s)$ (i.e., the performance weight) to weigh the sensitivity function of the closed-loop system, $\Delta(s)$ is a block-diagonal matrix representing perturbations in the parameters of the plant, $\Delta_P(s)$ is a full complex perturbation matrix encoding performance specifications, $K(s)$ is a feedback controller, $r(t) - d(t)$ are the external contributions, $u(t)$ is the control input, $e(t)$ is the tracking error to be minimized, and $e'(t)$ is $e(t)$ filtered by $W_P(s)$.

as in Figs. 1 and 2. The RP specification amounts to imposing [33, Chapter 8], [39, Chapter 10]:

$$\left\|S(s; W_P, K, \Delta)\right\|_\infty < 1, \quad \forall \Delta(s) \in \mathcal{B}_\Delta, \tag{5a}$$

$$S(s; W_P, K, \Delta) := \mathcal{F}_u\left(N(s; W_P, K), \Delta(s)\right), \tag{5b}$$

$$N(s; W_P, K) := \mathcal{F}_l\left(P(s; W_P), K(s)\right), \tag{5c}$$

where $\mathcal{B}_\Delta$ is the structured set of allowed perturbations, which is derived from (3) and reads as:

$$\mathcal{B}_\Delta = \left\{ \Delta(s) : \|\Delta(s)\|_\infty \le 1, \Delta(s) = \text{diag}\left(\delta_\lambda I_{m_\lambda}, \delta_{\varphi_1} I_{m_{\varphi_1}}, \dots, \delta_{\xi_{n_{p,\mathbb{C}}}} I_{m_{\xi_{n_{p,\mathbb{C}}}}} \right) \right\},$$

$m_\rho \in \mathbb{N}$ being the multiplicity of the uncertain parameter $\rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{p,\mathbb{C}}}\}$, while $\mathcal{F}_u(\cdot,\cdot)$ and $\mathcal{F}_l(\cdot,\cdot)$ denote the upper and lower Linear Fractional Transformation (LFT) representations, respectively [38, Chapter 9]. Provided that $N(s; W_P, K)$ is internally stable, the RP condition in (5) is equivalent to [20], [33, Theorem 8.7]

$$\mu_{\hat{\Delta}}\left(N(j\omega; W_P, K)\right) < 1, \quad \forall \omega \in \mathbb{R}_{\ge 0}, \tag{6}$$

where the structured singular value $\mu_{\hat{\Delta}}(\cdot)$ is computed with respect to $\hat{\Delta}(s) := \text{diag}(\Delta(s), \Delta_P(s))$, $\Delta_P(s)$ being a full complex perturbation matrix such that $\left\|\Delta_P(s)\right\|_\infty \le 1$ that allows treating performance constraints as stability ones. Thus, *given* a performance weight $W_P(s)$, *μ-synthesis* aims to find a nominally stabilizing controller $K^*(s)$ that minimizes the just-mentioned structured singular value across frequency [33, Chapter 8]:

$$K^*(s) = \arg \min_{K(s) \in \mathcal{K}} \left[\sup_{\omega \in \mathbb{R}_{\ge 0}} \mu_{\hat{\Delta}}\left(N(j\omega; W_P, K)\right) \right], \tag{7}$$

where $\mathcal{K}$ is the set of all nominally stabilizing controllers. Then, if $K^*(s)$ satisfies (6), robust performance is achieved. Problem (7) is typically solved via *DK*-iteration-like schemes [2,13,37].

As multiplicative faults might directly influence the stability of the closed-loop system, a PFTC must at least guarantee stability for all faulty behaviors. Nonetheless, it would be beneficial to strive for maximum closed-loop performance even in spite of faults. Thus, in this paper, we focus on the *automatic* tuning of $W_P(s)$ for attaining the maximally possible performance, in line with [20,35], by solving the following *nested optimization problem*

$$\min_{W_P(s)} \bar{\mathcal{L}}\left(W_P(s)\right) := \left\| \frac{1}{W_P(s)} \right\|_2 \tag{8a}$$

$$\text{s.t.} \quad \min_{K(s)\in\mathcal{K}} \left[\sup_{\omega\in\mathbb{R}_{\geq 0}} \mu_{\hat{\Delta}} \left(N\left(j\omega; W_{\mathrm{P}}, K\right)\right)\right] < 1, \tag{8b}$$

where the cost function $\bar{\mathcal{L}}\left(W_{\mathrm{P}}(s)\right)$ represents the $\mathcal{H}_2$-norm of the inverse of the performance weight $W_{\mathrm{P}}(s)$ while the constraint involves solving a μ-synthesis problem as in (7). Typically, to ease the resolution of Problem (8), $W_{\mathrm{P}}(s)$ is defined as a parametric transfer function with fixed structure that depends on a set of parameters θ that can assume values within a set Ω, as we will see in Section 4.1.

Remark 1. Different representations that describe plant uncertainty exist [33]. In this work, we focus on a specific description just to exemplify the proposed optimization algorithm, which, in reality, can handle any type of norm-bounded block-diagonal $\Delta(s)$, i.e., possibly including both parametric and unstructured uncertainty. Further, parametric uncertainty is not necessarily restricted to the form in (1). Rather, it can take any suitable structure (e.g., uncertainty on the parameters of a state-space model as in Section 4.5), as those shown in [38].

Remark 2. Problem (8) can be extended to include more than one performance weight, as in [20]. In this setting, the transfer function $W_{\mathrm{P}}(s)$ is replaced by a transfer matrix $W(s)$ that incorporates the additional performance weights. Then, the definition of the generalized plant $P(s)$ in Fig. 2 should account for the additional weights and related closed-loop transfer functions, see [14, Chapter 4], [33, Chapter 9]. Considering additional performance weights, such as the one on the control sensitivity function, prevents the synthesized controller from saturating the actuators and accounts for their physical limitations.

2.3. Motivation of the work

Problem (8) exhibits two key properties:

(p1) A *computationally-light* cost function;
(p2) A *computationally-expensive* constraint.

The nature of (8b) makes the resolution of (8) particularly challenging as noted in [20]:

(c1) Solving for $K(s)$ in (8b) is a nonlinear and non-convex optimization problem;
(c2) The mixed-μ synthesis procedure involved in (8b) is computationally-expensive (i.e., time-consuming).

Despite solving Problem (8) in a *global* fashion might overcome Challenge (c1), traditional global optimization algorithms may fail due to Challenge (c2), as finding accurate global solutions involves evaluating the cost function and the computationally-expensive constraint in (8b) at a number of points that scales *exponentially* with the number of optimization variables [24]. To mitigate this issue, Challenge (c2) can be tackled by surrogate-based methods. However, to the best of the authors' knowledge, there does not exist a black-box optimization algorithm in the literature explicitly tailored to optimization problems with Properties (p1)–(p2). Thus, in the next Section, we propose a novel surrogate-based method suited to this optimization problem setting and prove its convergence.

3. Global optimization under black-box constraints (GOBBC)

This section presents the *Global Optimization under Black-Box Constraints* (GOBBC) *algorithm*, a novel surrogate-based method for solving optimization problems with Properties (p1)–(p2) in a global fashion. In particular, Section 3.1 defines the problem statement in the general framework. Then, Section 3.2 describes the GOBBC optimization algorithm in detail. Finally, Section 3.3 analyses the convergence properties of our proposal.

3.1. Problem statement

3.1.1. Considered optimization problem

Consider the following global optimization problem:

$$\Theta^* = \arg\min_{\theta} \mathcal{L}(\theta) \tag{9a}$$

$$\text{s.t.} \quad \theta \in \Omega \cap \Xi, \tag{9b}$$

where $\theta \in \mathbb{R}^d$, $d \in \mathbb{N}$, are the decision variables, $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ is a *computationally-light* cost function, $\Omega \subset \mathbb{R}^d$ is a set of *inexpensive* constraints (e.g., bound constraints on θ),

$$\Xi := \left\{\theta : \boldsymbol{g}_{\Xi}(\theta) \leq \boldsymbol{0}_q\right\} \tag{10}$$

is a set of $q \in \mathbb{N}$ *black-box* constraints with $\boldsymbol{g}_{\Xi} : \mathbb{R}^d \to \mathbb{R}^q$, and

$$\Theta^* = \left\{ \theta_i^* : \theta_i^* \in \Omega \cap \Xi, \mathcal{L}\left(\theta_i^*\right) = \min_{\theta\in\Omega\cap\Xi} \mathcal{L}(\theta) \right\} \tag{11}$$

is the set of global minimizers of Problem (9).

We make the following assumption to ensure that Θ^* is nonempty (recall the Extreme Value Theorem [3, Chapter 2]).

Assumption 1. The optimization problem in (9) is such that:

1. The cost function $\mathcal{L}(\theta)$ is a continuous function;
2. The sets of constraints Ω and Ξ are nonempty and compact;
3. $\Omega \cap \Xi \neq \varnothing$.

3.1.2. Feasibility function

Rather than dealing with the set of black-box constraints Ξ in (10) directly, we define the Ξ-*feasibility function* as:

$$u_{\Xi}(\theta) := \begin{cases} 1 & \text{if } \theta \in \Xi, \\ 0 & \text{if } \theta \notin \Xi, \end{cases} \tag{12}$$

i.e., such that $u_{\Xi}(\theta) = 0$ if any of the constraints in (10) is violated, and $u_{\Xi}(\theta) = 1$ otherwise. Consistent with [40], in this general setting, we assume that *we can only assess the* Ξ-*feasibility function* in (12) rather than the value taken by each of the constraint functions $\boldsymbol{g}_{\Xi}(\theta)$.

Notation-wise, we say that a point $\theta \in \mathbb{R}^d$ is

- Ξ-*feasible* (respectively, Ξ-infeasible) w.r.t. Problem (9) if $u_{\Xi}(\theta) = 1$ ($u_{\Xi}(\theta) = 0$);
- *Feasible* (respectively, infeasible) if $\theta \in \Omega \cap \Xi$ ($\theta \notin \Omega \cap \Xi$).

3.1.3. Data collection

As with any surrogate-based method [27, Chapter 2], [36], GOBBC progressively evaluates different calibrations of the decision variables, keeping track of the set of points for which the values of the expensive functions (in our case, the Ξ-feasibility function in (12)) are available. In particular, let $\ell \in \mathbb{N}$ be the number of points $\theta_i \in \Omega$, $i \in \{1, \ldots, \ell\}$, tested by GOBBC at a certain iteration, and let $u_i := u_{\Xi}\left(\theta_i\right)$ be their feasibility w.r.t. Ξ. This information is grouped in the set

$$\mathcal{D}_{\ell} := \left\{ \left(\theta_i, u_i\right) : \theta_i \in \Omega, u_i = u_{\Xi}\left(\theta_i\right), i \in \{1, \ldots, \ell\} \right\}. \tag{13}$$

Further, we define the *best candidate point* among the ℓ available as:

$$\theta^*(\ell) := \begin{cases} \arg\min_{\theta_i : (\theta_i, u_i)\in\mathcal{D}_{\ell}, u_i = 1} \mathcal{L}\left(\theta_i\right) & \text{if } \exists \left(\theta_i, u_i\right) \in \mathcal{D}_{\ell} \text{ s.t. } u_i = 1, \\ \arg\min_{\theta_i : (\theta_i, u_i)\in\mathcal{D}_{\ell}} \mathcal{L}\left(\theta_i\right) & \text{otherwise,} \end{cases} \tag{14}$$

i.e., $\theta^*(\ell)$ is the tested point that attains the lowest cost function value and can either be feasible, if at least one Ξ-feasible tuning was tried, or infeasible, if $\mathcal{D}_{\ell}$ does not contain a Ξ-feasible point.

3.1.4. Aim of GOBBC

The aim of the proposed GOBBC algorithm is to solve Problem (9) effectively and efficiently. From a practical perspective, this entails

- Finding a *feasible* calibration $\boldsymbol{\theta}^*(\ell)$ as in (14) that is close to (at least) one of the global minimizers in (11), i.e., ensuring that $\min_{\boldsymbol{\theta}_i^* \in \Theta^*} \left\| \boldsymbol{\theta}^*(\ell) - \boldsymbol{\theta}_i^* \right\|_2$ is small, or, equivalently, striving for a sufficiently small $\left| \mathcal{L}\left(\boldsymbol{\theta}^*(\ell)\right) - \min_{\boldsymbol{\theta} \in \Omega \cap \Xi} \mathcal{L}(\boldsymbol{\theta}) \right|$ (*effectiveness*);
- Simultaneously, testing the smallest possible number of calibrations, i.e., keeping ℓ small (*efficiency*).

3.2. The GOBBC algorithm in detail

The GOBBC algorithm is an *iterative* procedure that consists of two main phases:

1. *Experimental design phase*, where an initial set of $\ell_{\text{init}} \in \mathbb{N}$ candidate points is sampled.
2. *Active search phase*, where the algorithm iteratively searches for a minimizer of Problem (9). To achieve this, given the data related to the previously-tested calibrations, i.e., $\mathcal{D}_\ell$ in (13) with $\ell \geq \ell_{\text{init}}$, GOBBC proposes new calibrations to try by alternating between three rationales:
   (r1) *Pure exploration*, i.e., probing those regions of Ω where few tunings have been tested;
   (r2) *Pure exploitation*, i.e., minimizing the cost function $\mathcal{L}(\boldsymbol{\theta})$ in (9) without considering the black-box constraints in (10). This is a degenerate case, as will be explained in Section 3.2.2.
   (r3) *Trading off exploration and exploitation*, i.e., looking for a new calibration that achieves a low cost function value while being Ξ-feasible according to a surrogate model that approximates the feasibility region described by Ξ in (10).

   Thus, at each iteration, GOBBC returns a tuning $\boldsymbol{\theta}_{\ell+1}$ to evaluate according to one of the just-mentioned rationales. Afterwards, we assess the Ξ-feasibility of the new sample $\boldsymbol{\theta}_{\ell+1}$, obtaining

   $$u_{\ell+1} = u_\Xi\left(\boldsymbol{\theta}_{\ell+1}\right).$$

   The pair $\left(\boldsymbol{\theta}_{\ell+1}, u_{\ell+1}\right)$ is then added to the available data, obtaining

   $$\mathcal{D}_{\ell+1} = \mathcal{D}_\ell \cup \left(\boldsymbol{\theta}_{\ell+1}, u_{\ell+1}\right).$$

   The active search phase is repeated until $\ell_{\max} \in \mathbb{N}$, $\ell_{\max} > \ell_{\text{init}}$, points have been tested ($\ell_{\max}$ representing the so-called *budget* of the optimization procedure).

The output of GOBBC is $\boldsymbol{\theta}^*\left(\ell_{\max}\right)$ as in (14), i.e., the best candidate point found within the prescribed budget. Algorithm 1 summarizes the GOBBC method. We now proceed to discuss the experimental design phase and the active search phase in detail.

3.2.1. Experimental design phase

Like any surrogate-based method [27, Chapter 2], [36], the GOBBC algorithm requires an initial sampling phase during which a set of points $\boldsymbol{\theta}_i \in \Omega$, $i \in \{1, \dots, \ell_{\text{init}}\}$, is generated in a *space-filling* and *non-collapsing* manner, i.e., in such a way that the obtained calibrations are uniformly spread in Ω and no two tunings share any coordinate value. This can be done via a Latin Hypercube Design (LHD) [23]. After generating the ℓ_{init} calibrations, the Ξ-feasibility of each $\boldsymbol{\theta}_i$ is assessed, leading to the initial dataset $\mathcal{D}_{\ell_{\text{init}}}$ as in (13).

3.2.2. Active search phase

Consider the set $\mathcal{D}_\ell$ in (13) with $\ell_{\text{init}} \leq \ell < \ell_{\max}$. At each iteration of the active search phase, GOBBC proposes a new calibration $\boldsymbol{\theta}_{\ell+1}$ to try based on $\mathcal{D}_\ell$ and according to either one of the Rationales (r1), (r2), or (r3), which are discussed next.

Algorithm 1 The GOBBC algorithm.

Input: (i) Cost function $\mathcal{L}(\boldsymbol{\theta})$ and constraint set Ω of Problem (9); (ii) Number of initial samples $\ell_{\text{init}} \in \mathbb{N}$; (iii) Choice of the surrogate model $p_\ell(\boldsymbol{\theta})$ (e.g., the IDWI function in (20)); (iv) Threshold $\eta \in [0,1]$ for the surrogate model $p_\ell(\boldsymbol{\theta})$; (v) Budget $\ell_{\max} \in \mathbb{N}, \ell_{\max} > \ell_{\text{init}}$; (vi) Number of iterations $k_{\text{exp}} \in \mathbb{N}$ after which to enforce exploration.
Output: (i) Best sample obtained by the procedure $\boldsymbol{\theta}^*\left(\ell_{\max}\right)$.

(Optional) Rescale the considered optimization problem as in [5, Section 3]
▷ *Experimental design phase* ◁
Generate a set of ℓ_{init} starting points $\boldsymbol{\theta}_i$, $i \in \{1, \dots, \ell_{\text{init}}\}$, in a space-filling and non-collapsing manner (e.g., via an LHD [23])
Assess the Ξ-feasibility of each $\boldsymbol{\theta}_i$, $i \in \{1, \dots, \ell_{\text{init}}\}$, obtaining $u_i = u_\Xi\left(\boldsymbol{\theta}_i\right)$
Build the initial dataset $\mathcal{D}_{\ell_{\text{init}}}$ as in (13)
Set $\ell = \ell_{\text{init}}$
Initialize the iteration counter for exploration: $k = 0$
▷ *Active search phase* ◁
while $\ell < \ell_{\max}$ **do**
  if $\nexists \left(\boldsymbol{\theta}_i, u_i\right) \in \mathcal{D}_\ell$ s.t. $u_i = 1$ **then** ▷ *Pure exploration*
    Solve Problem (16) to get $\boldsymbol{\theta}_{\ell+1}$
  else if $\nexists \left(\boldsymbol{\theta}_i, u_i\right) \in \mathcal{D}_\ell$ s.t. $u_i = 0$ **then** ▷ *Pure exploitation*
    Solve Problem (17) to get $\boldsymbol{\theta}_{\ell+1}$
  else
    if $k = k_{\text{exp}}$ **then** ▷ *Enforce exploration*
      Solve Problem (16) to get $\boldsymbol{\theta}_{\ell+1}$
      Reset the iteration counter: $k = 0$
    else ▷ *Trade off exploration and exploitation*
      Build the surrogate model $p_\ell(\boldsymbol{\theta})$ using the available data $\mathcal{D}_\ell$ (e.g., as in (20) in the case of GOBBC-IDWI)
      Solve Problem (19) to get $\boldsymbol{\theta}_{\ell+1}$, which relies on $p_\ell(\boldsymbol{\theta})$ to approximate the Ξ-feasibility region
      Increase the iteration counter: $k = k + 1$
  Assess the Ξ-feasibility of $\boldsymbol{\theta}_{\ell+1}$, obtaining $u_{\ell+1} = u_\Xi\left(\boldsymbol{\theta}_{\ell+1}\right)$
  Update the data, i.e., $\mathcal{D}_{\ell+1} = \mathcal{D}_\ell \cup \left(\boldsymbol{\theta}_{\ell+1}, u_{\ell+1}\right)$
  Increase the number of tested samples: $\ell = \ell + 1$
return $\boldsymbol{\theta}^*\left(\ell_{\max}\right)$ as in (14)

Pure exploration (r1). Whenever $\mathcal{D}_\ell$ in (13) does not contain a Ξ-feasible point, i.e., $u_i = 0$, $\forall i \in \{1, \dots, \ell\}$, emphasis should be placed on finding a feasible point for Problem (9) rather than minimizing the cost function $\mathcal{L}(\boldsymbol{\theta})$. Therefore, we should promote the exploration of those regions of Ω where relatively few points have been tried. For that purpose, we consider the *Inverse Distance Weighting (IDW) distance function* $z_\ell : \mathbb{R}^d \to (-1, 0]$ defined as [5]:

$$z_\ell(\boldsymbol{\theta}) := \begin{cases} 0 & \text{if } \exists \left(\boldsymbol{\theta}_i, u_i\right) \in \mathcal{D}_\ell \text{ s.t. } \boldsymbol{\theta}_i = \boldsymbol{\theta}, \\ -\frac{2}{\pi} \arctan\left(\frac{1}{\sum_{i=1}^{\ell} w_i(\boldsymbol{\theta})}\right) & \text{otherwise,} \end{cases} \tag{15}$$

$$w_i(\boldsymbol{\theta}) := \frac{1}{\left\| \boldsymbol{\theta} - \boldsymbol{\theta}_i \right\|_2}.$$

In practice, $z_\ell(\boldsymbol{\theta})$ in (15) assumes lower values at points that are furthest away from the samples $\boldsymbol{\theta}_i$ in $\mathcal{D}_\ell$, and for this reason the IDW distance function was used to promote exploration in several SBMs [5,6,27,28]. Consequently, in this work, we propose to select $\boldsymbol{\theta}_{\ell+1}$ as follows:

$$\boldsymbol{\theta}_{\ell+1} = \arg\min_{\boldsymbol{\theta}} z_\ell(\boldsymbol{\theta}) \tag{16}$$
$$\text{s.t.} \quad \boldsymbol{\theta} \in \Omega.$$

Given that $z_\ell(\boldsymbol{\theta})$ in (15) is a continuous function [5, Lemma 2], under Assumption 1, Problem (16) always admits a solution due to the Extreme Value Theorem [3, Chapter 2].

Remark 3. For ease of discussion, the optimization problems that concern the active search phase of GOBBC (e.g., Problem (16)) are presented as having a single global minimizer, although this may not be the case. That is because, at each iteration, we are only interested in finding a single point $\boldsymbol{\theta}_{\ell+1}$ to evaluate, which may be any one of the global minimizers of the considered problem, favoring those that were not previously tested. In any case, if there exists a $(\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell$ such that $\boldsymbol{\theta}_{\ell+1} = \boldsymbol{\theta}_i$, we can introduce a slight perturbation to any of the values that make up $\boldsymbol{\theta}_{\ell+1}$ to avoid repeating evaluations.

Pure exploitation (r2). In case all the tested points in $\mathcal{D}_\ell$ in (13) are feasible, i.e., $u_i = 1$, $\forall i \in \{1, \dots, \ell\}$, it could be that $\Omega \cap \Xi = \Omega$ (degenerate case) for Problem (9). This motivates the search for a point that minimizes $\mathcal{L}(\boldsymbol{\theta})$ subject to only the inexpensive constraints, namely

$$\begin{aligned} \boldsymbol{\theta}_{\ell+1} = \arg\min_{\boldsymbol{\theta}} \ & \mathcal{L}(\boldsymbol{\theta}) \\ \text{s.t.} \quad & \boldsymbol{\theta} \in \Omega. \end{aligned} \tag{17}$$

Note that, under Assumption 1, Problem (17) always admits a solution due to the Extreme Value Theorem [3, Chapter 2]. Clearly, $\min_{\boldsymbol{\theta}\in\Omega} \mathcal{L}(\boldsymbol{\theta}) \leq \min_{\boldsymbol{\theta}\in\Omega\cap\Xi} \mathcal{L}(\boldsymbol{\theta})$, so that $\boldsymbol{\theta}_{\ell+1}$ found in (17) attains the lowest possible cost function value for Problem (9). Furthermore, if $\boldsymbol{\theta}_{\ell+1}$ were to be Ξ-feasible, then it would be one of the global minimizers of Problem (9), and GOBBC should be stopped immediately as a consequence.

Trading off exploration and exploitation (r3). Whenever $\mathcal{D}_\ell$ in (13) contains both Ξ-feasible and Ξ-infeasible points, the search for $\boldsymbol{\theta}_{\ell+1}$ should look for a point such that $\mathcal{L}(\boldsymbol{\theta}_{\ell+1}) \leq \mathcal{L}(\boldsymbol{\theta}^*(\ell))$ while avoiding points such that $u_\Xi(\boldsymbol{\theta}_{\ell+1}) = 0$. To do so, we propose to construct a cheap *surrogate model* $p_\ell : \mathbb{R}^d \to [0, 1]$ that *describes the probability of a point being Ξ-feasible given the available information*, namely

$$p_\ell(\boldsymbol{\theta}) := \mathbb{P}\left(u_\Xi(\boldsymbol{\theta}) = 1 \,\middle|\, \mathcal{D}_\ell, \boldsymbol{\theta}\right), \tag{18}$$

where $\mathbb{P}\left(u_\Xi(\boldsymbol{\theta}) = 1 \,\middle|\, \mathcal{D}_\ell, \boldsymbol{\theta}\right)$ represents the just-mentioned probability. Then, in line with traditional machine learning approaches for classification [8], we can define a classifier that distinguishes between Ξ-feasible and Ξ-infeasible points according to a *threshold* $\eta \in [0, 1]$ for $p_\ell(\boldsymbol{\theta})$ in (18). Particularly, points such that $p_\ell(\boldsymbol{\theta}) \geq \eta$ are likely to be Ξ-feasible according to the data in $\mathcal{D}_\ell$ in (13), and vice versa whenever $p_\ell(\boldsymbol{\theta}) < \eta$. Then, $p_\ell(\boldsymbol{\theta})$ in (18) can be used as a proxy for the expensive constraint set in (10), replacing Problem (9) with the optimization problem

$$\begin{aligned} \boldsymbol{\theta}_{\ell+1} = \arg\min_{\boldsymbol{\theta}} \ & \mathcal{L}(\boldsymbol{\theta}) \\ \text{s.t.} \quad & \boldsymbol{\theta} \in \Omega \\ & p_\ell(\boldsymbol{\theta}) \geq \eta. \end{aligned} \tag{19}$$

The choice of the surrogate model is covered in detail in Section 3.2.3.

Remark 4. Different from Problem (9), the optimization problems solved during the active search phase of GOBBC, i.e., (16), (17), and (19), involve only the inexpensive cost function $\mathcal{L}(\boldsymbol{\theta})$, the constraints Ω, and the surrogate model $p_\ell(\boldsymbol{\theta})$, making them relatively cheap to solve globally. Consequently, we suggest using general-purpose global optimization procedures, such as DIRECT [17], PSO [18], and genetic algorithms [4] (see Section 1).

On the importance of exploration. Pure exploration (Problem (16)) plays a fundamental role in the GOBBC algorithm and should not be restricted to only when $\nexists (\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell$ in (13) such that $u_i = 1$. That is because the quality of the solutions returned by Problem (19) depends on how well the constraint $p_\ell(\boldsymbol{\theta}) \geq \eta$ approximates the feasibility region associated with Ξ. Furthermore, assuming that $\mathcal{D}_\ell$ contains both Ξ-feasible and Ξ-infeasible samples, finding $\boldsymbol{\theta}_{\ell+1}$ as in (19) at every iteration may result in an *overly conservative* black-box optimization procedure. Indeed, the algorithm would be prone to focusing its search efforts in those regions of Ω where Ξ-feasible samples have been tested, possibly neglecting portions of the search space where the global minimizers of (9) are actually located. To avoid this issue, we propose to *resort to pure exploration as in Problem (16) every $k_{exp} \in \mathbb{N}$ iterations*, leading to surrogate models $p_\ell(\boldsymbol{\theta})$ in subsequent iterations that are likely more accurate over the whole Ω and, ultimately, do not leave any promising region of the search space unexplored. This design choice is also *closely tied to the global convergence of GOBBC*, as detailed in Section 3.3.

3.2.3. Definition of the surrogate model

So far, we have yet to define a surrogate model $p_\ell(\boldsymbol{\theta})$ that can be used in Problem (19). In practice, GOBBC can use any function $p_\ell(\boldsymbol{\theta})$ for which the following assumption holds.

Assumption 2. The surrogate model $p_\ell(\boldsymbol{\theta})$ in (18)

1. Is a continuous function;
2. Has codomain bounded and equal to $[0, 1]$;
3. Is such that $p_\ell\left(\boldsymbol{\theta}^*(\ell)\right) \geq \eta$, $\forall \ell \in \mathbb{N}$, $\ell_{\text{init}} \leq \ell < \ell_{\max}$.

From a practical perspective, Assumption 2 ensures that: (i) $p_\ell(\boldsymbol{\theta})$ is a valid probability function, (ii) Problem (19) always admits a solution due to the Extreme Value Theorem [3, Chapter 2] (when combined with Assumption 1), (iii) Problem (19) does not discourage the search in a neighborhood of the best candidate point since we guarantee at least that[4] $p_\ell\left(\boldsymbol{\theta}^*(\ell)\right) \geq \eta$, otherwise we could miss a solution of the global optimization problem in (9).

In this work, similarly to [40], we suggest using the *Inverse Distance Weighting Interpolation (IDWI) function* as the surrogate model, which is defined as

$$p_\ell(\boldsymbol{\theta}) := \sum_{i=1}^{\ell} v_i(\boldsymbol{\theta})\, u_i, \tag{20}$$

$$v_i(\boldsymbol{\theta}) := \begin{cases} 1 & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}_i, (\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell, \\ 0 & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}_{i'}, (\boldsymbol{\theta}_{i'}, u_{i'}) \in \mathcal{D}_\ell, i' \neq i, \\ \frac{\tilde{w}_i(\boldsymbol{\theta})}{\sum_{i'=1}^{\ell} \tilde{w}_{i'}(\boldsymbol{\theta})} & \text{otherwise,} \end{cases}$$

$$\tilde{w}_i(\boldsymbol{\theta}) := \frac{\exp\left(-\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\|_2\right)}{\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\|_2}.$$

The IDWI function in (20) is a valid surrogate model according to Assumption 2 since it is continuous but also [5, Lemma 1]:

- It assumes the correct codomain when $\mathcal{D}_\ell$ in (13) contains both Ξ-feasible and Ξ-infeasible points, i.e.,

$$0 = \min_{(\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell} u_i \leq p_\ell(\boldsymbol{\theta}) \leq \max_{(\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell} u_i = 1, \quad \forall \boldsymbol{\theta} \in \mathbb{R}^d.$$

- $p_\ell(\boldsymbol{\theta}_i) = u_i$, $\forall i \in \{1, \dots, \ell\}$, implying that $p_\ell(\boldsymbol{\theta}_i) = 0$ for those points $\boldsymbol{\theta}_i$ in $\mathcal{D}_\ell$ that are Ξ-infeasible, and $p_\ell(\boldsymbol{\theta}_i) = 1$ for the Ξ-feasible $\boldsymbol{\theta}_i$'s. In turn, this means that surely $p_\ell\left(\boldsymbol{\theta}^*(\ell)\right) \geq \eta$.

In what follows, we will refer to the implementation of GOBBC that employs the surrogate model in (20) as GOBBC-IDWI. Nonetheless, other valid surrogate models exist. One such model is the modified probabilistic support vector machines classifier proposed in [27, Chapter 6].

3.2.4. On the choice of the hyperparameters of GOBBC

Besides the choice of the surrogate model explained in the previous section, the GOBBC method (Algorithm 1) requires tuning four *hyperparameters*: the budget $\ell_{\max}$, the number of initial points ℓ_{init}, the threshold η for $p_\ell(\boldsymbol{\theta})$, and how often to enforce pure exploration as defined by k_{exp}.

[4] Note that $\boldsymbol{\theta}^*(\ell) \in \Xi$ if $\exists (\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_\ell$ s.t. $u_i = 1$, see (14).

In practice, the budget should be determined by the time constraints of the application under consideration. For instance, assuming that $t_\Xi \in \mathbb{R}_{>0}$ (in s) is an approximation of the time needed for an evaluation of the computationally-expensive constraints encapsulated in (12), and that $t_{\max} \in \mathbb{R}_{>0}$ is the maximum time allocatable to the optimization procedure, the budget can be chosen as $\ell_{\max} = \lfloor \frac{t_{\max}}{t_\Xi} \rfloor$, $\lfloor \cdot \rfloor$ being the floor function.

Concerning ℓ_{init}, there is no consensus in the SBM literature regarding this choice. Popular choices for methods that do not consider black-box constraints, e.g. [5,15,28,32], range from $2d$ to $4d$ in the case of LHDs, or 2^d when starting from the vertices of the hypercube defined by the bound constraints in Ω in (9). In our case, due to the presence of the black-box constraints in (10), we suggest generating a higher number of initial samples, as it is desirable to start the active search phase with at least one Ξ-feasible calibration. Based on our empirical findings, a good rule of thumb is to set ℓ_{init} between 15% and 50% of $\ell_{\max}$.

Moving on, the threshold η can be viewed as a tuning knob that regulates how much we are willing to explore Ω to find better Ξ-feasible points (see Problem (19)). Setting $\eta \to 1$ results in a conservative (or "safe") algorithm that tends to search only in a neighborhood of the available Ξ-feasible points, as the constraint $p_\ell(\theta) \geq \eta$ will be satisfied only when it is very likely (according to $p_\ell(\theta)$) that $\theta \in \Xi$. Choosing $\eta \to 0$ essentially means that Ξ-feasibility is only a minor concern, making Problem (19) very similar to the pure exploitation one in (17), penalizing the search only in those regions of Ω where samples are very likely to be Ξ-infeasible (i.e., $p_\ell(\theta) \to 0$). Instead, $\eta \approx 0.5$ offers a good trade-off between these two rationales (in fact, it is a common choice when dealing with machine learning classification tasks [8]), and we suggest it as the standard value to use, motivated by our empirical results.

Finally, k_{exp} is sort of a fail-safe that ensures that GOBBC is ultimately convergent (see Section 3.3) or, roughly speaking, that the method does not leave any promising region of Ω behind. We suggest choosing a low k_{exp} (i.e., promote exploration often) when $\eta \to 1$ to prevent overconservativeness of the algorithm, while k_{exp} should be high (i.e., rarely explore) when $\eta \to 0$, as exploration is intrinsically carried out already due to this choice of η (although in a completely different fashion).[5] When $\eta \approx 0.5$, the choice of k_{exp} is less critical but depends on the problem at hand. For more details, Appendix A analyzes different values of η and k_{exp} for Algorithm 1 on a synthetic optimization benchmark.

3.2.5. Rescaling the optimization problem

The IDW distance function in (15) and the IDWI function in (20) are fairly sensitive to the ranges of the decision variables since they are based on Euclidean norms. Thus, to avoid numerical issues in the presence of decision variables that assume vastly different ranges, we resort to the rescaling strategy proposed in [5, Section 3], which rescales each variable so that it assumes values between -1 and 1 within the constraint set of interest for the optimization problem.

3.3. Global convergence analysis

In the global optimization literature, convergence is typically analyzed *asymptotically*, i.e., as the number of tested candidate points $\ell \to \infty$. Clearly, this contradicts the goal of surrogate-based methods, whose aim is to find a sufficiently accurate calibration by trying the least number of points (see Section 3.1.4). Yet, convergence results for SBMs are available in the literature, see e.g. [15,27,28,32]. The underlying motivation behind these convergence analyses is proving that the methods are not "flawed" in the sense that, if allowed to run indefinitely, they can return the global minimizers of the considered optimization problem (under Assumption 1). To address the global convergence of GOBBC, recall the following fundamental result from the global optimization literature.

Theorem 1 (**Convergence of a global optimization algorithm [34]**). *An algorithm converges to the global minimum of every continuous function over any compact set $\mathcal{S} \subset \mathbb{R}^d$ if and only if its sequence of tested points,*

$$\langle \boldsymbol{\theta}_i \rangle_{i \geq 1} := \langle \boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \rangle,$$

is dense in $\mathcal{S}$.

Roughly speaking, ensuring the denseness of the sequence of tested points of a global optimization algorithm means that the procedure must generate calibrations that differ from all the samples that were previously tested every once in a while, highlighting the importance of a *thorough exploration of the decision space* $\mathcal{S}$. Nonetheless, to promote efficiency, the detailed exploration of $\mathcal{S}$ can be arbitrarily delayed, favoring the search for promising solutions in the first iterations of the algorithm (e.g., in those regions of $\mathcal{S}$ that were previously sampled), and leaving the exhaustive search for the global minimizers for later iterations.

We can now state the convergence result of GOBBC, which leverages Theorem 1.

Theorem 2 (**Convergence of GOBBC**). *Consider the optimization problem in (9) under Assumption 1. A sufficient condition for the convergence of GOBBC to the global minimum of Problem (9) is that k_{exp} is finite, the surrogate model $p_\ell(\theta)$ satisfies Assumption 2, and $\ell_{max} \to \infty$.*

Proof. The proof is reported in Appendix B. □

Theorem 2 highlights the importance of carrying out pure exploration every once in a while for GOBBC. Then, k_{exp} represents a tuning parameter that regulates how often we are willing to forego the search for a potentially better solution via the exploitation of the cost function $\mathcal{L}(\theta)$ and surrogate model $p_\ell(\theta)$ in favor of the thorough exploration of Ω, which is required for global convergence.

Remark 5. Theorem 2 ensures that, asymptotically, GOBBC generates a set of points that is dense in $\Omega \cap \Xi$. In turn, this means that the algorithm finds calibrations that are arbitrarily close to the global minimizers Θ^* of Problem (9). However, GOBBC does not intrinsically make a choice among those points that attain the minimum $\min_{\theta \in \Omega \cap \Xi} \mathcal{L}(\theta)$. If that is a concern, Algorithm 1 can be easily modified to return also the set $\mathcal{D}_{\ell_{\max}}$ other than $\boldsymbol{\theta}^*(\ell_{\max})$, as well as the cost function value for each of the tested θ_i's, $1 \leq i \leq \ell_{\max}$. Then, in a fashion similar to multi-objective optimization [21], a choice among the solutions obtained by GOBBC that attain the minimum cost function value can be made by a human decision-maker.

4. Numerical results

This section applies the GOBBC algorithm presented in Section 3 to solve the μ-synthesis with automatic performance weight selection problem in (8). In particular, Section 4.1 defines the details and practicalities of the optimization problem. Then, Section 4.2 presents the settings used in the numerical experiments, the compared optimization algorithms, and the considered indicators of performance. The susceptibility of the results w.r.t. the plant order and uncertainty level is assessed in Sections 4.3 and 4.4, respectively. Finally, Section 4.5 evaluates the robustness of the designed PFTC on a simulated mechatronic system in closed-loop subject to a multiplicative fault.

All the numerical experiments presented in this section were conducted in MATLAB on a workstation with a 64-core 3.20 GHz CPU and 255 GB of RAM.

[5] Solving $\theta_{\ell+1} = \arg\min_{\theta \in \Omega, p_\ell(\theta) \geq \eta} \mathcal{L}(\theta)$ for $\eta \to 0$ makes GOBBC explore mostly those regions of Ω where the minimizers of $\mathcal{L}(\theta)$ s.t. only $\theta \in \Omega$ are located. Instead, pure exploration, as in Problem (16), samples Ω in a space-filling and non-collapsing manner.

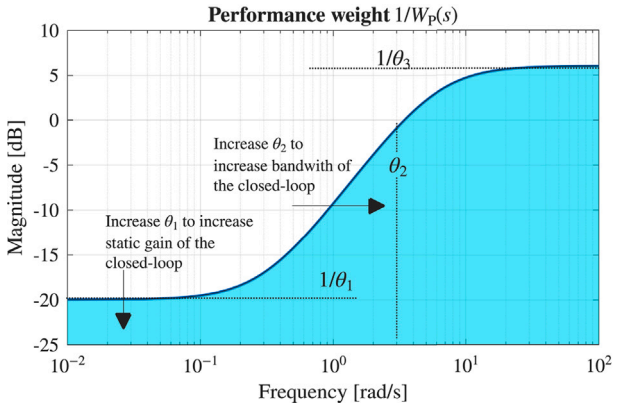


Fig. 3. Minimizing the $\mathcal{H}_2$-norm (shaded area) of $1/W_P(s)$ involves optimizing for a performance weight $W_P(s)$ that weighs the sensitivity function of the closed-loop system, striving for higher bandwidth and higher disturbance attenuation.

4.1. Application of GOBBC to a μ-synthesis problem with performance weight optimization

In its current formulation, Problem (8) requires some caveats to make it a valid optimization problem. First, we define the performance weight $W_P(s)$ as the transfer function with parameters $\boldsymbol{\theta} = \begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}^\top \in \mathbb{R}^3_{>0}$ as in [33, Chapter 2]:

$$W_P(s;\boldsymbol{\theta}) := \frac{\left(\theta_3^{1/n_W} s + \theta_2\right)^{n_W}}{\left(s + \dfrac{\theta_2}{\theta_1^{1/n_W}}\right)^{n_W}}, \tag{21}$$

where $n_W \in \mathbb{N}$ is the order of the weight, θ_1 is the *static gain*, θ_2 (in $\frac{\text{rad}}{\text{s}}$) is (approximately) the *bandwidth* requirement for the closed-loop system, and θ_3 is the *high-frequency gain* such that $\theta_3 < \theta_1$. The set of possible values for $\boldsymbol{\theta}$ in (21) is defined as:

$$\Omega := \left\{\boldsymbol{\theta} : \boldsymbol{\theta}_l \le \boldsymbol{\theta} \le \boldsymbol{\theta}_u\right\} \tag{22}$$

with $\boldsymbol{\theta}_l = \begin{bmatrix}\theta_{l_1} & \theta_{l_2} & \theta_{l_3}\end{bmatrix}^\top \in \mathbb{R}^3_{>0}$, $\boldsymbol{\theta}_u = \begin{bmatrix}\theta_{u_1} & \theta_{u_2} & \theta_{u_3}\end{bmatrix}^\top \in \mathbb{R}^3_{>0}$. Second, as (21) is biproper, the cost function in (8a) practically reads as

$$\mathcal{L}(\boldsymbol{\theta}) := \sqrt{\frac{1}{2\pi}\int_{\omega_{\min}}^{\omega_{\max}} \left|\frac{1}{W_P(j\omega;\boldsymbol{\theta})}\right|^2 \mathrm{d}\omega}, \tag{23}$$

where $\omega_{\min}, \omega_{\max} \in \mathbb{R}_{\ge 0}$ (in $\frac{\text{rad}}{\text{s}}$) are such that $\omega_{\min} < \theta_{l_2} < \theta_{u_2} < \omega_{\max}$. The cost function in (23) can be viewed as the $\mathcal{H}_2$-norm of the inverse of the performance weight computed over the frequency range $[\omega_{\min}, \omega_{\max}]$ rather than $(-\infty, \infty)$ [33, Chapter 2], see Fig. 3. Third, the constraint in (8b) should be converted into a non-strict inequality constraint via the addition of an arbitrarily small constant $\varepsilon \in \mathbb{R}_{>0}$. Then, Problem (8) reads as the nested optimization problem[6]

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \tag{24a}$$

$$\text{s.t.} \quad \boldsymbol{\theta} \in \Omega \tag{24b}$$

$$\min_{K(s)\in\mathcal{K}} \left[\sup_{\omega\in\mathbb{R}_{\ge 0}} \mu_{\hat{\Delta}}\left(N\left(j\omega; W_P(\boldsymbol{\theta}), K\right)\right)\right] \le 1-\varepsilon, \tag{24c}$$

[6] For ease of discussion, we consider only one ($\boldsymbol{\theta}^*$) of the possible global minimizers of Problem (24), although the discussion can be extended in that regard, as was done in Section 3.

where (24b) is the set of inexpensive constraints in (22), and (24c) is the computationally-expensive constraint solved by DK-iteration-like algorithms [2,13,37] for a fixed performance weight $W_P(s;\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$. Clearly, Problem (24) has the same structure as Problem (9), making GOBBC (Algorithm 1) a suitable candidate solver. Once Problem (24) is solved, the best-performing closed-loop controller $K^*(s)$ is attained as in (7) using the optimal parameters $\boldsymbol{\theta}^*$ for the performance weight $W_P(s;\boldsymbol{\theta})$.

4.2. Comparison methods, simulation settings, and performance indicators

We compare the GOBBC-IDWI algorithm in Section 3.2 with two popular global optimization algorithms:

1. Augmented Lagrangian Genetic Algorithm (referred to as GA) [4, 10];
2. Interior-Point method with Multi-Start (referred to as IP-MS) [22, 25].

The settings for GOBBC-IDWI (Algorithm 1) are $\eta = 0.5$, $\ell_{\max} = 100$, $\ell_{\text{init}} = 30$, and $k_{\exp} = 5$. The initial samples are generated via a Latin Hypercube Design [23]. The optimization problems that concern the active search phase of GOBBC-IDWI (Section 3.2.2) are solved by the PSO global optimization algorithm [18]. Instead, GA and IP-MS use the standard settings from their respective MATLAB implementations.

To make the comparison between the optimization algorithms fair, the *same* initial samples are used as the dataset $\mathcal{D}_{\ell_{\text{init}}}$ in (13) for GOBBC-IDWI, the initial population of GA, and the starting points of the local optimization procedures for IP-MS. To maximize the time efficiency of GA and IP-MS, these methods are *parallelized* across CPU cores. In turn, this means that GA simultaneously updates its population of ℓ_{init} points while IP-MS solves the ℓ_{init} optimization problems in parallel. Clearly, this should favor GA and IP-MS from an efficiency standpoint compared to GOBBC-IDWI, since the latter is intrinsically a *sequential* method that cannot be parallelized. Finally, GA and IP-MS are stopped only once their respective convergence criteria are satisfied, allowing us to assess their most accurate solutions. Clearly, this may involve testing many more points than the $\ell_{\max}$ allotted to GOBBC-IDWI. Nevertheless, despite these seemingly unfavorable comparison conditions, in Sections 4.3 and 4.4 we will see that GOBBC-IDWI can achieve satisfactory levels of effectiveness and efficiency, even surpassing the other two methods.

Concerning the μ-synthesis problem with performance weight optimization in (24), the order of the performance weight $W_P(s;\boldsymbol{\theta})$ in (21) is set to $n_W = 2$, while the bounds for the tunable parameters are set to $\theta_{l_1} = 1\,\text{dB}$, $\theta_{u_1} = 60\,\text{dB}$ (static gain), $\theta_{l_2} = 10^{-8}\,\frac{\text{rad}}{\text{s}}$, $\theta_{u_2} = 10^{8}\,\frac{\text{rad}}{\text{s}}$ (bandwidth),[7] and $\theta_{l_3} = -60\,\text{dB}$, $\theta_{u_3} = -1\,\text{dB}$ (high-frequency gain), comprising a vast range. The frequency range over which to compute the cost function in (23) is set to $\omega_{\min} = 10^{-10}\,\frac{\text{rad}}{\text{s}}$ and $\omega_{\max} = 10^{10}\,\frac{\text{rad}}{\text{s}}$. Finally, $\varepsilon = 10^{-8}$ for the constraint in (24c).

To compare the considered optimization algorithms, we consider three performance indicators:

1. The *(relative) value of the cost function* achieved by the best-found calibration $\boldsymbol{\theta}_b^*$ (e.g., $\boldsymbol{\theta}_b^* = \boldsymbol{\theta}^*\left(\ell_{\max}\right)$ in the case of GOBBC-IDWI, see Algorithm 1) with respect to the best possible parameterization $\boldsymbol{\theta}_u$ attainable by $\mathcal{L}(\boldsymbol{\theta})$ in (23) within Ω in (22), i.e.,[8]

$$\tilde{\mathcal{L}}\left(\boldsymbol{\theta}_b^*\right) := \frac{\mathcal{L}\left(\boldsymbol{\theta}_b^*\right) - \mathcal{L}\left(\boldsymbol{\theta}_u\right)}{\mathcal{L}\left(\boldsymbol{\theta}_u\right)}. \tag{25}$$

[7] We apply a logarithmic transformation to θ_2 to assist the optimizers in handling its large range.

[8] Note that $\boldsymbol{\theta}_u = \arg\min_{\boldsymbol{\theta}\in\Omega} \mathcal{L}(\boldsymbol{\theta})$, i.e., if the constraint in (24c) were omitted, minimizing the $\mathcal{H}_2$-norm of $1/W_P(s;\boldsymbol{\theta})$ would amount to choosing the maximal static gain, bandwidth, and high-frequency gain for the performance weight.

This indicator measures the *effectiveness* of the optimizers in finding the global minimum of (24) (see Section 3.1.4) and should be as low as possible.

2. The *time required by the optimization algorithm to reach the solution* θ_{b}^{*} along with the *number of evaluated points*, which matches the number of times the μ-synthesis procedure was executed (see the constraint in (24c)). This indicator is a measure of *efficiency*, as explained in Section 3.1.4, and should be as low as possible.
3. The *degree of conservativeness* of the μ-synthesis solution in (7). Specifically,

$$\gamma\left(\theta_{\mathrm{b}}^{*}\right) := \min_{K(s)\in\mathcal{K}} \left[\sup_{\omega\in\mathbb{R}_{\geq 0}} \mu_{\hat{\Delta}}\left(N\left(j\omega; W_{\mathrm{P}}\left(\theta_{\mathrm{b}}^{*}\right), K\right)\right)\right] \tag{26}$$

is an indicator of the closed-loop performance attained by the optimal controller synthesized according to the tuned performance weight $W_{\mathrm{P}}\left(s;\theta_{\mathrm{b}}^{*}\right)$ [33, Chapter 8]. In particular, $\gamma\left(\theta_{\mathrm{b}}^{*}\right) \to 1$ represents a controller that achieves the desired performance, $\gamma\left(\theta_{\mathrm{b}}^{*}\right) \to 0$ marks a controller that still has margins of improvement with respect to the desired performance, and $\gamma\left(\theta_{\mathrm{b}}^{*}\right) \geq 1$ implies that the optimizer was not able to find a performance weight that guarantees robust performance (as reviewed in Section 2.2), i.e., it failed to return a feasible solution of Problem (24).

4.3. Numerical results for varying plant orders

First, we assess how the order of the plant model $G_p(s)$ in (1) affects the effectiveness and efficiency of the considered optimization algorithms, as well as the impact it has on the robust control performance. For this purpose, for each system order, we carry out $n_{\mathrm{MC}} = 10$ Monte Carlo (MC) simulations by varying the mean parameter values of $G_p(s)$ but keeping the relative uncertainties constant and equal to 10%, i.e. $r_\rho = 0.1$ in (2), $\forall \rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{\mathrm{p,\mathbb{C}}}}\}$. In detail, the considered $G_p(s)$'s amount to:

1. A *first-order* system with $g = 0$, $n_{\mathrm{z,\mathbb{R}}} = n_{\mathrm{z,\mathbb{C}}} = 0$, $n_{\mathrm{p,\mathbb{R}}} = 1$, and $n_{\mathrm{p,\mathbb{C}}} = 0$. The mean gains are such that $\bar{\lambda} \in [10, 100]$, while the mean pole time constants $\bar{\tau}_1$ belong to $[0.1, 10]$.
2. A *second-order* system with $g = 0$, and $n_{\mathrm{z,\mathbb{R}}}, n_{\mathrm{p,\mathbb{R}}} \in \{0, 1, 2\}$, $n_{\mathrm{z,\mathbb{C}}}, n_{\mathrm{p,\mathbb{C}}} \in \{0, 1\}$, randomly extracted in such a way as to have $n_{\mathrm{p,\mathbb{R}}} + 2n_{\mathrm{p,\mathbb{C}}} = 2$ and $n_{\mathrm{z,\mathbb{R}}} + 2n_{\mathrm{z,\mathbb{C}}} \leq 2$. The mean parameter values are such that $\bar{\lambda} \in [10, 100]$, $\bar{\tau}_i, \bar{\varphi}_{i'} \in [0.1, 10]$, $\forall i \in \{1, \dots, n_{\mathrm{p,\mathbb{R}}}\}$, $\forall i' \in \{1, \dots, n_{\mathrm{z,\mathbb{R}}}\}$, and $\bar{\beta}_i, \bar{\alpha}_{i'} \in [0.1, 10]$, $\bar{\xi}_i, \bar{\zeta}_{i'} \in [0.1, 0.9]$, $\forall i \in \{1, \dots, n_{\mathrm{p,\mathbb{C}}}\}$, $\forall i' \in \{1, \dots, n_{\mathrm{z,\mathbb{C}}}\}$.
3. A *third-order* system with $g \in \{0, 1\}$, $n_{\mathrm{z,\mathbb{R}}}, n_{\mathrm{p,\mathbb{R}}} \in \{0, \dots, 3\}$, and $n_{\mathrm{z,\mathbb{C}}}, n_{\mathrm{p,\mathbb{C}}} \in \{0, 1\}$, randomly extracted in such a way as to have $g + n_{\mathrm{p,\mathbb{R}}} + 2n_{\mathrm{p,\mathbb{C}}} = 3$ and $n_{\mathrm{z,\mathbb{R}}} + 2n_{\mathrm{z,\mathbb{C}}} \leq 3$. The mean parameter values share the same ranges as the previous case.
4. A *fourth-order* system with $g \in \{0, 1\}$, $n_{\mathrm{z,\mathbb{R}}}, n_{\mathrm{p,\mathbb{R}}} \in \{0, \dots, 4\}$, and $n_{\mathrm{z,\mathbb{C}}}, n_{\mathrm{p,\mathbb{C}}} \in \{0, 2\}$, randomly extracted in such a way as to have $g + n_{\mathrm{p,\mathbb{R}}} + 2n_{\mathrm{p,\mathbb{C}}} = 4$ and $n_{\mathrm{z,\mathbb{R}}} + 2n_{\mathrm{z,\mathbb{C}}} \leq 4$. The mean parameter values share the same ranges as the second-order system.

For each MC simulation, the values $\bar{\rho}$, $\rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{\mathrm{p,\mathbb{C}}}}\}$, are extracted from the uniform distribution with supports chosen as above. Given the limited orders of the considered systems, we employ simple *fixed-structure* PID controllers, namely

$$K(s;\boldsymbol{\psi}) := \psi_1 + \frac{\psi_2}{s} + \frac{\psi_3 s}{1 + \psi_4 s}, \tag{27}$$

with parameters $\boldsymbol{\psi} = \begin{bmatrix}\psi_1 & \psi_2 & \psi_3 & \psi_4\end{bmatrix}^{\top} \in \mathbb{R}^4$. Then, the μ-synthesis procedure in (7) will return the optimal parameters $\boldsymbol{\psi}^*$ for the PID controller in (27) given the performance weight $W_{\mathrm{P}}\left(s;\theta_{\mathrm{b}}^{*}\right)$ found by the optimizer.

Fig. 4(a) and Table 1 show the performance indicators described in Section 4.2 in this setting. The results are outstanding: *GOBBC-IDWI matches the effectiveness of* GA*, offers greater efficiency, and produces performance weights that yield less conservative robust controllers*. Indeed, we would expect a general-purpose global optimization algorithm to return an accurate global solution of Problem (24) when allowed to run until convergence. For this reason, the solutions returned by GA can be regarded as being very close to the true global minimum of Problem (24), aside from the occasional outliers. Yet, GOBBC-IDWI is able to reach similar values of $\tilde{\mathcal{L}}\left(\theta_{\mathrm{b}}^{*}\right)$ in (25) despite evaluating (roughly) *one hundredth* fewer tunings than GA. This translates into a significant saving of computational time, as GOBBC-IDWI takes nearly one order of magnitude less time than GA. To put it into perspective, considering the fourth-order system (which is the most computationally-intensive), GOBBC-IDWI runs for 676 minutes in the worst case among the n_{MC} ones while GA needs 1691 minutes to converge, a significant difference.

Table 1
Number of times the μ-synthesis procedure was executed by the optimization procedures when solving Problem (24), which is a measure of efficiency (see Section 4.2). *GOBBC-IDWI always tests the same number of calibrations, which amounts to $\ell_{\max} = 100$.

	System order ($r_\rho = 0.1$)			
Method	1	2	3	4
GOBBC-IDWI*	100	100	100	100
GA	7288	7190	7190	7205
IP-MS	2862	2682	2452	2364

	Relative uncertainty (2nd order system)				
Method	5%	10%	15%	20%	25%
GOBBC-IDWI*	100	100	100	100	100
GA	7190	7190	7190	7190	7190
IP-MS	2618	2682	2633	2626	2577

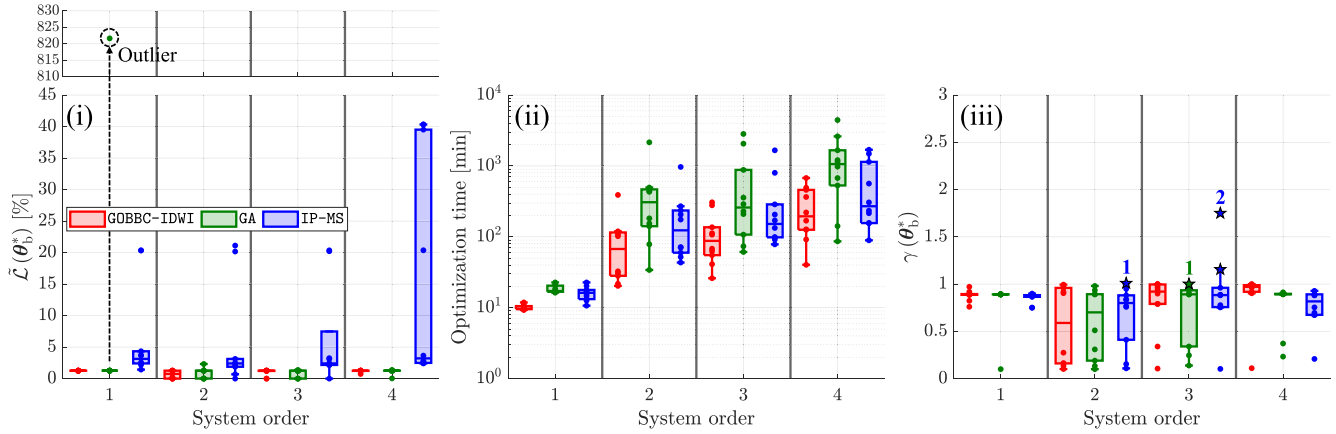
Concerning IP-MS, this optimization algorithm severely underperforms in terms of effectiveness. Indeed, due to the nonlinearity and nonconvexity of Problem (24) (Section 2.3), a proper initialization of this procedure is crucial, as it involves running local solvers. In our simulations, despite employing the multi-start methodology, IP-MS lags behind in terms of $\tilde{\mathcal{L}}\left(\theta_{\mathrm{b}}^{*}\right)$ in (25), especially for higher plant orders. In terms of efficiency, this method is in between GOBBC-IDWI and GA both in terms of optimization time and number of evaluations.

Regarding $\gamma\left(\theta_{\mathrm{b}}^{*}\right)$ in (26), we can see that GOBBC-IDWI usually attains the best results in the sense that its corresponding distributions are usually less dispersed and closer to 1. Remarkably, *GOBBC-IDWI is able to achieve robust performance ($\gamma\left(\theta_{b}^{*}\right) < 1$) in all the considered scenarios* while GA (respectively, IP-MS) fails to do so one (three) time(s). This motivates our proposal even further in the considered application setting, as attaining feasible solutions is of utmost importance for Problem (24).

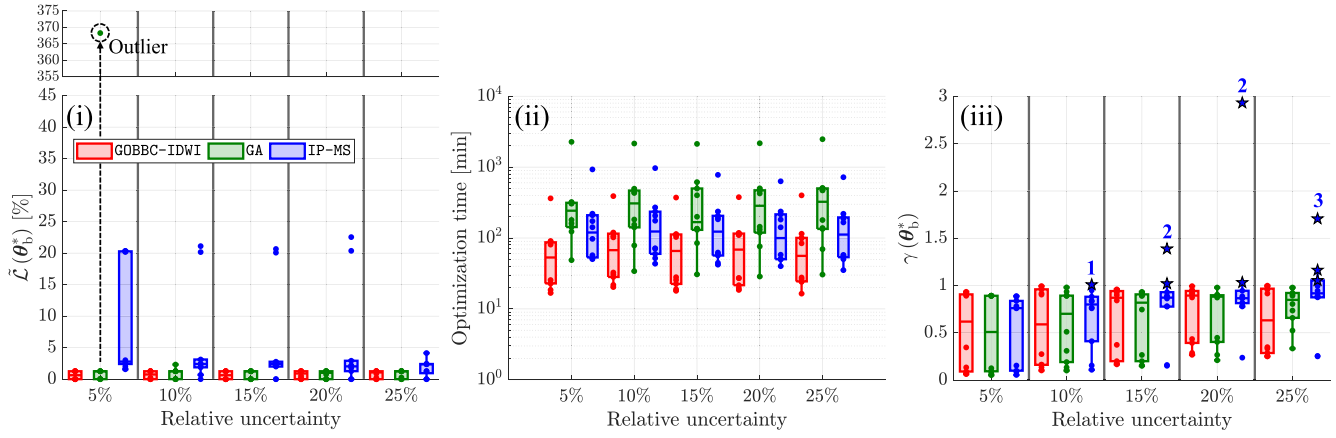
To conclude, restricting our analysis to GOBBC-IDWI, we can assess how the order of the plant affects the results of our proposal by examining Fig. 4(a). In our simulations, *there is no significant change in effectiveness ($\tilde{\mathcal{L}}\left(\theta_{b}^{*}\right)$ in (25)) as the plant order increases*, implying that we can find similarly performing performance weights in all cases. Concerning efficiency, *computational times tend to increase as the plant order increases* since the complexity of μ-synthesis scales with the number of uncertain parameters [33, Chapter 8]. Hence, this is not an issue of GOBBC-IDWI. Finally, no significant pattern is observed for the degree of conservativeness indicator in (26).

4.4. Numerical results for varying uncertainty levels

Next, we assess how the relative uncertainty in the parameters of the plant model $G_p(s)$ in (1) impacts the effectiveness, efficiency, and robust control performance of the considered optimization algorithms. To that end, in a fashion similar to Section 4.3, we conduct $n_{\mathrm{MC}} = 10$ Monte Carlo simulations this time by varying r_ρ in (2), $\rho \in \{\lambda, \varphi_1, \dots, \xi_{n_{\mathrm{p,\mathbb{C}}}}\}$, but



(a) Performance for varying plant orders.



(b) Performance for varying relative uncertainty levels.

Fig. 4. Distribution of the performance attained by the considered optimization algorithms for varying plant orders and relative uncertainty levels on the $n_{\text{MC}} = 10$ different systems. The distributions are represented using box plots; due to the low n_{MC}, we also display the specific results for each plant using filled circle markers. In particular, (i) shows the effectiveness indicator in (25); (ii) depicts the time required by the optimization algorithms to reach a solution. We remark that the optimization times are reported on a *logarithmic* scale and that `GA` and `IP-MS` evaluate 30 samples at a time (due to the *parallelization* explained in Section 4.2), whereas `GOBBC-IDWI` assesses only one calibration at a time. Finally, (iii) is the degree of conservativeness of the μ-synthesis procedure reported in (26). In this final plot, we use star markers to highlight those simulations such that $\gamma(\theta_b^*) \geq 1$, i.e., when the optimization solvers failed to find a performance weight that guarantees robust performance. Furthermore, we also report the number of times that is the case.

keeping the plant's order constant and equal to two. In particular, the $G_p(s)$'s are generated exactly as described in Section 4.3 (second-order system), but now we test five different levels of relative uncertainty $r_\rho \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$. Again, we employ fixed-structure PID controllers as in (27).

Fig. 4(b) and Table 1 show the performance indicators described in Section 4.2 for the considered simulations. Even in this setting, *`GOBBC-IDWI` outperforms the other two methods in the sense that it is as effective as `GA` but notably more efficient, also attaining satisfactory degrees of conservativeness with regard to the μ-synthesis solutions*. As a matter of fact, consistent with Section 4.3, `GOBBC-IDWI` achieves values of $\tilde{\mathcal{L}}(\theta_b^*)$ in (25) that are similar to those attained by `GA` while taking notably less time to run, evaluating (roughly) *one hundredth* fewer calibrations. `IP-MS` still lags behind `GOBBC-IDWI` and `GA` in terms of effectiveness, but it is more efficient than the genetic algorithm. However, `IP-MS` heavily struggles in finding feasible solutions ($\gamma(\theta_b^*) < 1$) for Problem (24) as the relative uncertainty increases, failing to achieve robust performance one time for $r_\rho = 0.1$, two times for $r_\rho \in \{0.15, 0.2\}$, and three times for $r_\rho = 0.25$. Instead, in this setting, `GOBBC-IDWI` and `GA` show similar distributions of $\gamma(\theta_b^*)$ in (26) and always yield robust-performing controllers.

In conclusion, from the results reported in Fig. 4(b), we can say that *changing the level of relative uncertainty does not noticeably impact the effectiveness or the efficiency* of the optimization algorithms. However, *increasing r_ρ can make it harder to find a controller that guarantees robust performance*, as the results of `IP-MS` suggest.

4.5. Passive fault-tolerant control of a three-mass mechatronic system

As a practical application, we consider the self-balancing manual manipulator system represented in Fig. 5, which assists human operators in handling heavy loads. Its main components are: (i) the *direct current motor* that actuates the load, (ii) the *steel wire rope*, wrapped around a drum, which transmits motion from the motor shaft to the handle, (iii) the *winding drum* on which the rope wraps around, (iv) the *load* to be moved, and (v) the *handle*, which is the mechanical device that holds the load and guides its motion. The system is controlled in closed-loop to modulate the motor speed. The wire rope is the *most critical component* of the system, as it degrades over time and eventually breaks after a certain number of maneuvers. It is thus important to guarantee the stability and control performance of the closed-loop system despite changes in the rope's properties caused by degradation.

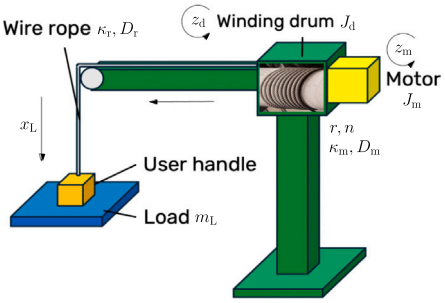


Fig. 5. Schematic representation of the considered mechatronic system, presenting the main components: motor (yellow), winding drum, wire rope (gray), and load (blue). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

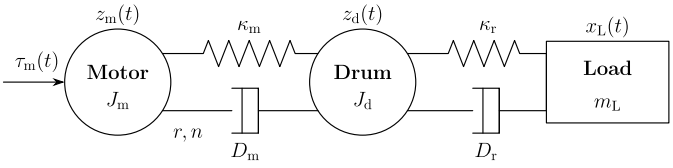


Fig. 6. Mass-spring–damper model of the system in Fig. 5. The motor and drum have inertia J_m and J_d, respectively, while the load has a mass m_L. The input is the motor torque $\tau_m(t)$. The rotational positions of the motor and drum are denoted as $z_m(t)$ and $z_d(t)$, respectively. The load translational position is denoted as $x_L(t)$. The transmission ratio between the motor and the winding drum is n. The winding drum has a radius r. The elements κ_m (rigidity) and D_m (viscous coefficient) characterize the elastic transmission between the motor and the winding drum, while κ_r and D_r characterize the wire rope, modeled as an elastic (although almost rigid) transmission.

Considering the motor torque $\tau_m(t)$ as the input and the motor rotational speed $\dot{z}_m(t)$ as the output, the system in Fig. 5 can be modeled as a mass-spring–damper LTI SISO system (Fig. 6) with state-space matrices (A, B, C, D) and state $\boldsymbol{x}(t) = \begin{bmatrix} z_m(t) & \dot{z}_m(t) & x_L(t) & \dot{x}_L(t) & z_d(t) & \dot{z}_d(t) \end{bmatrix}^\top \in \mathbb{R}^6$ with:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{\kappa_m}{J_m} & -\frac{D_m}{J_m} & 0 & 0 & \frac{n\kappa_m}{J_m} & \frac{nD_m}{J_m} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{\kappa_r}{m_L} & -\frac{D_r}{m_L} & \frac{r\kappa_r}{m_L} & \frac{rD_r}{m_L} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ \frac{n\kappa_m}{J_d} & \frac{nD_m}{J_d} & \frac{r\kappa_r}{J_d} & \frac{rD_r}{J_d} & -\frac{n^2\kappa_m}{J_d} - \frac{r^2\kappa_r}{J_d} & -\frac{n^2D_m}{J_d} - \frac{r^2D_r}{J_d} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & \frac{1}{J_m} & 0 & 0 & 0 & 0 \end{bmatrix}^\top$$

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}, \qquad D = 0.$$

The value of each parameter is reported in Table 2.

Having introduced the three-mass system under study, the aim is now to design a PFTC for the motor rotational speed $\dot{z}_m(t)$ by actuating the torque $\tau_m(t)$. The control performance should be maintained despite rope degradation. Specifically, the degradation of the wire rope could include loosening or breaking of some of its wires. We model such degradation as a decrease in the value of its stiffness κ_r. This modeling assumption is motivated by considering the relationship between the stiffness of a rope subject to axial loads and its constructive parameters, i.e.,

$$\kappa_r \approx \epsilon \frac{a}{c},$$

Table 2
Parameters of the system in Fig. 6.

Parameter	Symbol	Value	Unit
Motor inertia	J_m	0.05	$kg \cdot m^2$
Drum inertia	J_d	0.8	$kg \cdot m^2$
Load mass	m_L	60	kg
Motor-drum stiffness	κ_m	4000	Nm/rad
Motor-drum damping	D_m	0.15	Nm/(rad/s)
Rope stiffness	κ_r	15,000	N/m
Rope damping	D_r	0.8	N/(m/s)
Transmission ratio motor-drum	n	3	–
Drum radius	r	0.3	m

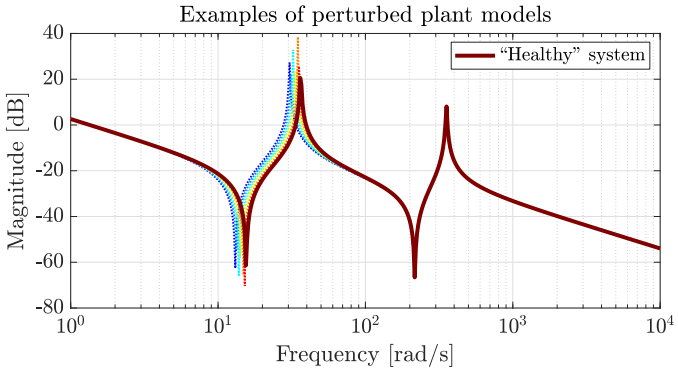


Fig. 7. Magnitude of the uncertain three-mass system in Fig. 5 for varying levels of rope stiffness as in (29). The model corresponding to a healthy rope is shown as a continuous line. The dashed lines represent different perturbed plant models with $\kappa_{r_{min}} \leq \kappa_{r_p} < \kappa_{r_{max}}$. Color intensity embeds the corresponding rope stiffness value κ_{r_p}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Results achieved by GOBBC-IDWI when applied to the design of a PFTC for the three-mass system in Fig. 5.

Indicator/parameter	Value
Relative cost $\tilde{\mathcal{L}}(\theta_b^*)$ in (25)	1.26%
Optimization time	11.37 min
Degree of conservativeness $\gamma(\theta_b^*)$ in (26)	0.99
Optimal static gain $\theta_{b_1}^*$ for (21)	60 dB
Optimal bandwidth $\theta_{b_2}^*$ for (21)	3.80 $\frac{rad}{s}$
Optimal high-frequency gain $\theta_{b_3}^*$ for (21)	−1 dB

where ϵ is the Young's modulus, a is the cross-sectional area of the rope, and c is a constant factor [11, Chapter 3.4]. Following the breaking of a wire, there is a decrease in the effective area of the rope a, and so κ_r decreases as a consequence. Then, to model rope degradation within the framework introduced in Section 2.1, we define its stiffness as an uncertain parameter like in (2), i.e.,

$$\kappa_{r_p} := \bar{\kappa}_r \left(1 + r_{\kappa_r} \delta_{\kappa_r}\right) \quad \text{for any } \delta_{\kappa_r} \in \mathbb{R}, \left|\delta_{\kappa_r}\right| \leq 1. \tag{29}$$

Specifically, let $\kappa_{r_{max}} = 15000\,N/m$ (as reported in Table 2) be the stiffness of the *healthy* rope. We consider the rope to have reached its *end-of-life* when its stiffness approaches the value $\kappa_{r_{min}} = 10500\,N/m$. Then, the mean rope stiffness in (29) is $\bar{\kappa}_r = \frac{\kappa_{r_{max}} + \kappa_{r_{min}}}{2} = 12750\,N/m$, while the relative uncertainty is $r_{\kappa_r} = \frac{\kappa_{r_{max}} - \kappa_{r_{min}}}{\kappa_{r_{max}} + \kappa_{r_{min}}} = 0.1765$. Fig. 7 shows the magnitude of several perturbed plant models, highlighting that κ_{r_p} in (29) acts on the low-frequency resonance of the considered three-mass system.

Given the uncertain plant model, we now employ the GOBBC-IDWI algorithm with the settings described in Section 4.2 to synthesize a

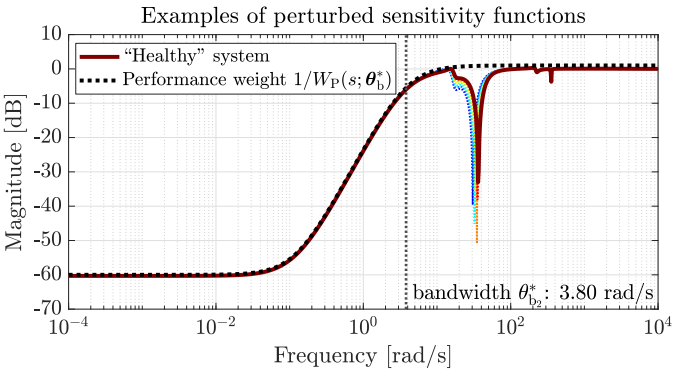


Fig. 8. Magnitude of the closed-loop sensitivity function of the uncertain three-mass system in Fig. 5 equipped with the optimal controller found by μ-synthesis with performance weight tuned by `GOBBC-IDWI` for varying levels of rope stiffness as in (29). The closed-loop model corresponding to a healthy rope is shown as a continuous line. The dashed colored lines represent different perturbed plant models with $\kappa_{r_{min}} \leq \kappa_{r_p} < \kappa_{r_{max}}$. Color intensity embeds the corresponding rope stiffness value κ_{r_p}. The dashed black line represents the magnitude of the inverse of the optimal performance weight. The figure also reports the desired bandwidth of the closed-loop system. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

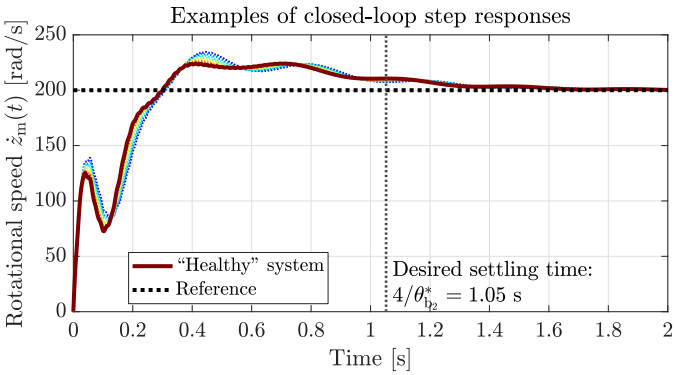


Fig. 9. Examples of closed-loop step responses of the uncertain three-mass system in Fig. 5 equipped with the optimal controller found by μ-synthesis with performance weight tuned by `GOBBC-IDWI` for varying levels of rope stiffness as in (29). In all cases, the reference is set to 200 rad/s (dashed black line). The closed-loop model corresponding to a healthy rope is shown as a continuous line. The dashed colored lines represent different perturbed plant models with $\kappa_{r_{min}} \leq \kappa_{r_p} < \kappa_{r_{max}}$. Color intensity embeds the corresponding rope stiffness value κ_{r_p}. The figure also reports a rough approximation of the desired settling time based on the bandwidth of the optimal performance weight $W_P(s;\theta_b^*)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

PFTC for the three-mass system under study, following the procedure in Section 4.1. Different from Sections 4.3 and 4.4, we design a controller $K(s)$ with *no structure imposed a-priori*. The results in terms of the indicators presented in Section 4.2 are reported in Table 3, including also the parameters θ_b^* of the optimal performance weight in (21). Note that robust performance is guaranteed since $\gamma(\theta_b^*) < 1$ as in (26). We can see that the limiting factor is the (approximate) bandwidth of the closed-loop system, which cannot be increased beyond 3.80 rad/s without affecting the RP of the controller. Interestingly, the optimization time is quite lower than those obtained in Fig. 4, despite the fact that we are dealing with a sixth-order system. However, we remark that the complexity of μ-synthesis scales heavily in the number of uncertain parameters, even more so than in the system's order [33, Chapter 8], making the problem tackled in this section (only one uncertain parameter) computationally-lighter than those in Sections 4.3 and 4.4.

To inspect the quality of the designed PFTC, Fig. 8 reports the magnitude of the closed-loop sensitivity function for varying perturbed plant models, comparing it against the inverse of the optimal performance weight $W_P(s;\theta_b^*)$. We can clearly see that the displayed sensitivity functions are always below $1/W_P(s;\theta_b^*)$, verifying the robust performance condition described in Section 2.2 from a practical perspective. For further validation, Fig. 9 displays several closed-loop step responses due to a 200 rad/s speed reference. We can see that the robust performance guaranteed during the PFTC design phase translates directly into a closed-loop system that maintains similar performance levels (e.g., settling times) despite the degradation of the steel wire rope.

5. Conclusions

In this paper, we propose a novel, globally-convergent, surrogate-based optimization method tailored for optimization problems with a computationally-light cost function and computationally-expensive constraints, which we refer to as the Global Optimization under Black-Box Constraints (`GOBBC`) algorithm. These types of optimization problems are particularly relevant in the context of μ-synthesis with automatic performance weight selection, where one has to solve a nested optimization problem to find the performance weight that maximizes closed-loop performance while preserving robust stability and robust performance of the system. Solving such a nested optimization problem is challenging due to the presence of local minima and a computationally-expensive μ-synthesis constraint, making `GOBBC` the ideal candidate for its resolution. Specifically, in this work, we rely on `GOBBC` to design passive fault-tolerant controllers for SISO LTI plants subject to multiplicative faults with maximally achievable closed-loop performance; the same rationale can be applied to synthesize robust controllers in general. Numerical results demonstrate that `GOBBC` is as effective as state-of-the-art general-purpose global optimization algorithms in finding the minima of the considered μ-synthesis optimization problem, while being remarkably more efficient in the sense that computational times are notably reduced, all the while yielding non-conservative PFTCs. The application to a three-mass mechatronic system practically proves that our proposal can maximize the closed-loop performance of a system whose components (respectively, parameters) degrade (change) over time.

Future directions include extending our proposal to the Multiple-Input Multiple-Output (MIMO) setting, considering also cases where users impose constraints on the complementary sensitivity and control sensitivity functions (or transfer matrices).

CRediT authorship contribution statement

Davide Previtali: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization. **Mirko Mazzoleni:** Writing – review & editing, Writing – original draft. **Nicholas Valceschini:** Software. **Fabio Previdi:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors confirm that a similar paper is neither under consideration nor already published in another venue.

Appendix A. Empirical case study on the hyperparameters of GOBBC

As an example, we consider the synthetic benchmark optimization problem `camelsixhumps - hard constrained` [40, Table I], a 2-dimensional optimization problem as in (9) that exhibits a narrow Ξ-feasibility region with respect to the bound constraints in Ω. This benchmark problem is such that the cost function $\mathcal{L}(\boldsymbol{\theta})$ is nonlinear and multimodal, and $\arg\min_{\boldsymbol{\theta}\in\Omega} \mathcal{L}(\boldsymbol{\theta}) = \{\begin{bmatrix}0.0898 & -0.7126\end{bmatrix}^\top, \begin{bmatrix}-0.0898 & 0.7126\end{bmatrix}^\top\}$ while $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\Omega\cap\Xi} \mathcal{L}(\boldsymbol{\theta}) = \begin{bmatrix}0.2126 & 0.5751\end{bmatrix}^\top$, i.e., the sets of global minimizers of the problem with and without the constraints in Ξ *do not match*.

We employ this benchmark optimization problem to test the performance of GOBBC-IDWI for different values of η and k_{exp}, thereby gaining insights into how these hyperparameters affect the optimization process. In all the simulations presented here, the budget is set to $\ell_{\max} = 200$, while the $\ell_{\text{init}} = 30$ initial samples are generated by an LHD [23]. All simulations employ the *same* initial starting samples, which already include two Ξ-feasible calibrations. We test all possible combinations of $\eta = 0.1$ (very low concern for Ξ-feasibility), $\eta = 0.5$ (standard setting), $\eta = 0.9$ (conservative), and $k_{\text{exp}} = 10$ (explore "often", i.e., $\lfloor(\ell_{\max} - \ell_{\text{init}})/(k_{\text{exp}}+1)\rfloor = 15$ times in total), $k_{\text{exp}} = 25$ (explore "sometimes", i.e., 6 times in total), and $k_{\text{exp}} = 40$ (explore "rarely", i.e., 4 times in total). The results are reported in Fig. A.10.

Analyzing the cases when $\eta = 0.1$ (Fig. A.10(a)), we can see that GOBBC-IDWI tests many Ξ-infeasible samples, especially because the Ξ-unconstrained global minimizers are outside of Ξ. As a matter of fact, most of the tested calibrations are close to $\begin{bmatrix}-0.0898 & 0.7126\end{bmatrix}^\top$. Varying k_{exp} in this setting changes how exploration is carried out. Particularly, $k_{\text{exp}} = 10$ allows GOBBC-IDWI to probe regions of Ω that are furthest away from the Ξ-unconstrained global minimizers, while the other choices lead the procedure to focus only on a neighborhood of $\begin{bmatrix}-0.0898 & 0.7126\end{bmatrix}^\top$. In any case, when $\eta = 0.1$, the setting $k_{\text{exp}} \in \{10, 25, 40\}$ does not particularly affect the convergence of the method (last row of Fig. A.10(a)), but *we believe that is the case solely because* $\begin{bmatrix}-0.0898 & 0.7126\end{bmatrix}^\top$ *is close to* $\boldsymbol{\theta}^*$*, ultimately allowing the procedure to find the feasible global minimizer*. Nonetheless, the approximate Ξ-feasibility boundaries in these simulations differ significantly from the real one.

Analyzing the cases when $\eta = 0.9$ (Fig. A.10(c)), we can observe a behavior that is opposite to the one previously described. Now, GOBBC-IDWI acts very conservatively and rarely samples far from the available Ξ-feasible calibrations. In fact, for $k_{\text{exp}} \in \{25, 40\}$ ($\eta = 0.9$), the method struggles to converge to the global minimizer, failing to do so within the allotted budget. The situation improves when $k_{\text{exp}} = 10$ ($\eta = 0.9$) as now the method is able to approach $\boldsymbol{\theta}^*$, although much more slowly than when $\eta \in \{0.1, 0.5\}$. In any case, the approximate Ξ-feasibility regions are extremely narrow when $\eta = 0.9$, explaining the conservative behavior of GOBBC-IDWI. We can conclude that k_{exp} *is a key parameter when* $\eta \to 1$*, and should be set low enough to prevent* GOBBC-IDWI *from getting stuck in a suboptimal region of the feasible space*.

Finally, the best trade-off is achieved when $\eta = 0.5$ (Fig. A.10(b)). In this setting, GOBBC-IDWI demonstrates a good balance of Ξ-feasible and Ξ-infeasible points, allowing the procedure to converge to the global minimizer relatively quickly while also producing an approximate Ξ-feasibility region $p_{\ell_{\max}-1}(\boldsymbol{\theta}) \geq \eta$ that closely resembles the actual one. The convergence rate in this setting (last row of Fig. A.10(b)) depends on the choice of k_{exp} (in this specific case, the best results are achieved by $k_{\text{exp}} = 25$), although no significant differences are observed.

Appendix B. Proof of Theorem 2

In what follows, we denote by Θ_ℓ the set comprising the first $\ell \in \mathbb{N}$ entries of the infinite sequence $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$, i.e. $\Theta_\ell = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_\ell\}$, while Θ_∞ represents the set containing all the elements of $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$. Besides Theorem 1, to prove the convergence of GOBBC we also leverage the following result from the global optimization literature, which provides a sufficient condition that ensures the denseness of Θ_∞ within a compact set.

Theorem 3 (A sufficient condition for the denseness of Θ_∞ [31]). *Let $S \subset \mathbb{R}^d$ be a compact set and $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$ be a sequence of points such that $\boldsymbol{\theta}_i \in S$, $\forall i \in \{1, 2, \dots\}$. Suppose that there exists a strictly increasing sequence of positive integers $\langle\ell_\iota\rangle_{\iota\geq 1}$, $\ell_\iota \in \mathbb{N}$, such that $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$ satisfies the following condition for some $\sigma \in (0, 1]$:*

$$\min_{1\leq i\leq \ell_\iota - 1} \left\|\boldsymbol{\theta}_{\ell_\iota} - \boldsymbol{\theta}_i\right\|_2 \geq \sigma \phi_S\left(\Theta_{\ell_\iota - 1}\right), \quad \forall \iota \in \mathbb{N}, \tag{B.1}$$

where:

$$\phi_S\left(\Theta_{\ell_\iota - 1}\right) := \max_{\boldsymbol{\theta}\in S} \min_{1\leq i\leq \ell_\iota - 1} \left\|\boldsymbol{\theta} - \boldsymbol{\theta}_i\right\|_2. \tag{B.2}$$

Then, Θ_∞ is dense in S.

Now, consider the sequence of tested points $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$ produced by GOBBC (Algorithm 1). The first ℓ_{init} elements of $\langle\boldsymbol{\theta}_i\rangle_{i\geq 1}$ are obtained by the chosen experimental design. Instead, each $\boldsymbol{\theta}_i \in \Theta_\ell$, $\ell > \ell_{\text{init}}$, is selected by either Problem (16), Problem (17), or Problem (19). We consider two cases.

Degenerate case. The first case is degenerate and considers Ξ to be a compact superset of Ω, i.e. $\Xi \supseteq \Omega$. Then, any $\boldsymbol{\theta} \in \Omega$ is Ξ-feasible and, consequently, $\boldsymbol{\theta}$ is also feasible w.r.t. Problem (9). In turn, this means that after the experimental design phase $\mathcal{D}_{\ell_{\text{init}}}$ in (13) is such that $\nexists\,(\boldsymbol{\theta}_i, u_i) \in \mathcal{D}_{\ell_{\text{init}}}$ with $u_i = 0$. Consequently, Algorithm 1 selects

$$\boldsymbol{\theta}_{\ell_{\text{init}}+1} = \arg\min_{\boldsymbol{\theta}\in\Omega} \mathcal{L}(\boldsymbol{\theta}),$$

which is such that $\boldsymbol{\theta}_{\ell_{\text{init}}+1} = \boldsymbol{\theta}_i^*$ for some $\boldsymbol{\theta}_i^* \in \Theta^*$ in (11) (recall Remark 3) due to $\Xi \supseteq \Omega$, attaining one of the global minimizers of Problem (9) and proving the global convergence of GOBBC (in finite time) in this case.

Non-degenerate case. The second case is when $\Omega \cap \Xi \subset \Omega$. To prove the convergence of GOBBC in this setting, we demonstrate that Θ_∞ is dense in Ω (recall Theorems 1 and 3). As a consequence, Θ_∞ is also dense in $\Omega \cap \Xi$. Note that, due to the finiteness of k_{exp}, Algorithm 1 solves Problem (16) an infinite number of times for $\ell \to \infty$, either:

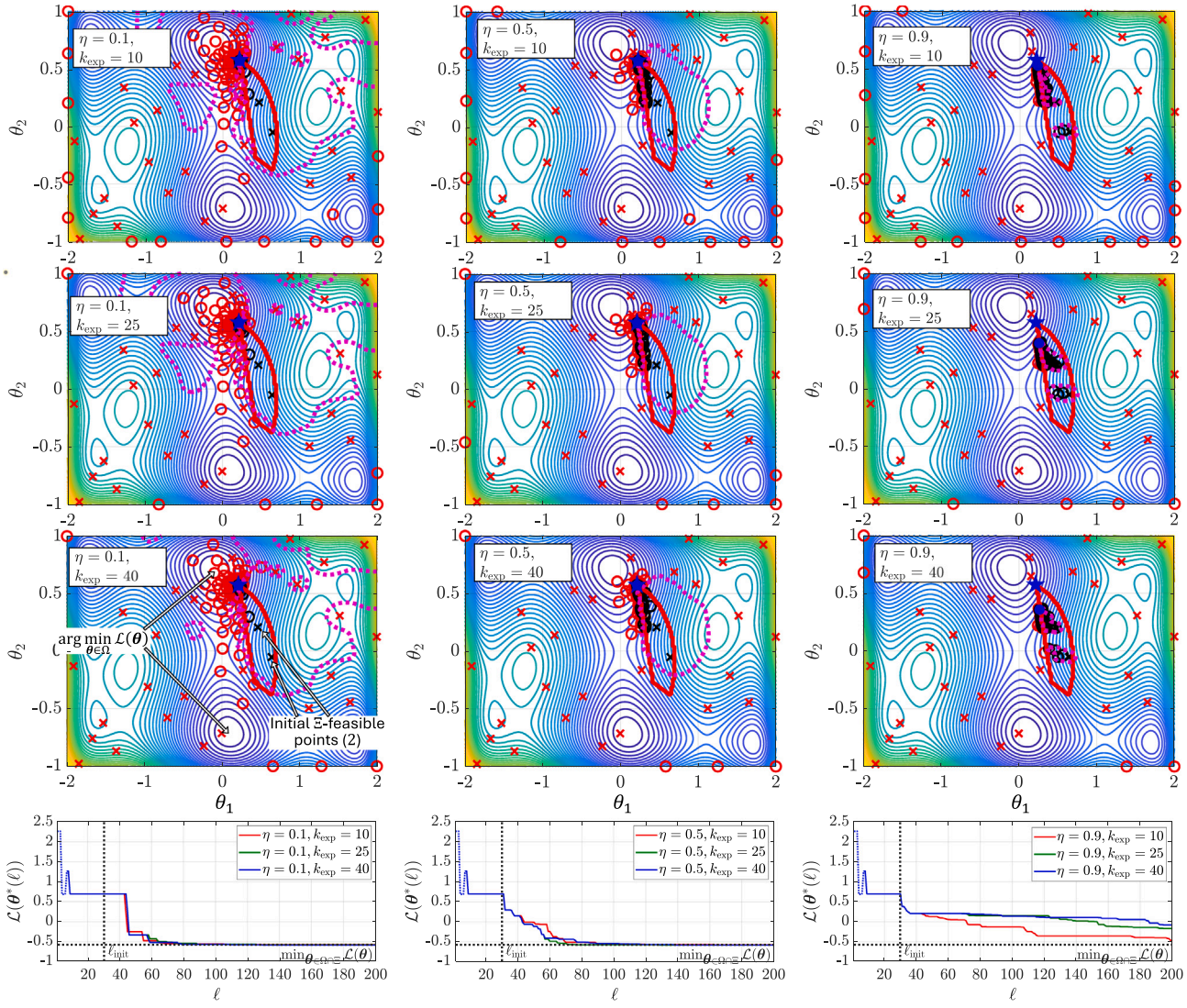
1. At every iteration until finding a Ξ-feasible point, and then every $k_{\text{exp}} + 1$ iterations;
2. Every $k_{\text{exp}} + 1$ iterations after an initial phase during which GOBBC solves Problem (17) instead of Problem (19), if $\mathcal{D}_{\ell_{\text{init}}}$ does not include a Ξ-infeasible point;
3. Every $k_{\text{exp}} + 1$ iterations if $\mathcal{D}_{\ell_{\text{init}}}$ in (13) is such that $\exists\,(\boldsymbol{\theta}_i, u_i), (\boldsymbol{\theta}_{i'}, u_{i'}) \in \mathcal{D}_{\ell_{\text{init}}}$, $i \neq i'$, with $u_i = 0$ and $u_{i'} = 1$.

Then, we can define a strictly increasing sequence of positive integers $\langle\ell_\iota\rangle_{\iota\geq 1}$, $\ell_\iota \in \mathbb{N}$, such that:

$$\boldsymbol{\theta}_{\ell_\iota} = \arg\min_{\boldsymbol{\theta}} z_{\ell_\iota - 1}(\boldsymbol{\theta}), \quad \forall \iota \in \mathbb{N} \tag{B.3}$$
$$\text{s.t.} \quad \boldsymbol{\theta} \in \Omega,$$

marking those candidate points found by solving Problem (16). Now, note that each $\boldsymbol{\theta}_i \in \Theta_{\ell_\iota - 1}$ is a global maximizer of Problem (B.3), as proven in [28, Proposition 4]. This result, together with the continuity of $z_{\ell_\iota - 1}(\boldsymbol{\theta})$ [5, Lemma 2], compactness of Ω (Assumption 1), and minimization over Ω as the constraint set, implies that:

$$\boldsymbol{\theta}_{\ell_\iota} \notin \Theta_{\ell_\iota - 1} \implies \exists \sigma_\iota \in (0, 1] \text{ s.t.} \tag{B.4}$$



(a) Results for $\eta = 0.1$. (b) Results for $\eta = 0.5$. (c) Results for $\eta = 0.9$.

Fig. A.10. Results achieved by GOBBC-IDWI (Algorithm 1) on the `camelsixhumps - hard constrained` [40, Table I] optimization benchmark for different thresholds $\eta \in \{0.1, 0.5, 0.9\}$ and exploration iterations $k_{\text{exp}} \in \{10, 25, 40\}$ values. The top three rows of plots display the level curves of the cost function $\mathcal{L}(\boldsymbol{\theta})$, the Ξ-feasibility boundary as a continuous red line, and the global minimizer as a blue star. The red and black markers denote all the $\ell_{\max}$ points tested by the GOBBC-IDWI algorithm, differentiating between the Ξ-feasible (black) and Ξ-infeasible (red) ones, as well as making a distinction between the calibrations tested during the experimental design phase (cross markers) and those found in the active search phase (circle markers). The blue filled circle marker shows $\boldsymbol{\theta}^*(\ell_{\max})$. Finally, the magenta dashed line is the approximate Ξ-feasibility boundary determined by $p_{\ell_{\max}-1}(\boldsymbol{\theta}) \geq \eta$ (i.e., at the last iteration of GOBBC-IDWI). The last row of plots shows the evolution of the cost function value of the best calibration $\boldsymbol{\theta}^*(\ell)$, $1 \leq \ell \leq \ell_{\max}$, iteratively found by GOBBC-IDWI. The colored dashed lines are associated with Ξ-infeasible $\boldsymbol{\theta}^*(\ell)$'s, while the continuous lines represent Ξ-feasible $\boldsymbol{\theta}^*(\ell)$'s. The dashed black vertical line marks the number of initial samples. The dashed black horizontal line is the global minimum of the benchmark optimization problem. Finally, the dashed black vertical line marks the number of initial samples. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

$$\min_{1 \leq i \leq \ell_\iota - 1} \left\| \boldsymbol{\theta}_{\ell_\iota} - \boldsymbol{\theta}_i \right\|_2 \geq \sigma_\iota \phi_\Omega \left(\Theta_{\ell_\iota - 1} \right), \quad \forall \iota \in \mathbb{N},$$

where $\phi_\Omega \left(\Theta_{\ell_\iota - 1} \right)$ is defined analogously to (B.2). Define

$$\sigma = \min_{\iota \in \mathbb{N}} \sigma_\iota,$$

and notice that (B.4) holds even if we set $\sigma_\iota = \sigma, \forall \iota \in \mathbb{N}$. Then, GOBBC generates a sequence of points that satisfies Condition (B.1) of Theorem 3, making Θ_∞ dense in Ω and, as a consequence, dense in Ω∩Ξ. This result, combined with Theorem 1, implies that GOBBC is a globally convergent optimization procedure. □

References

[1] Apkarian P, Dao MN, Noll D. Parametric robust structured control design. IEEE Trans Autom Control 2015;60:1857–69. https://doi.org/10.1109/TAC.2015.2396644

[2] Apkarian P, Noll D. Nonsmooth $\mathcal{H}_\infty$ synthesis. IEEE Trans Autom Control 2006;51:71–86. https://doi.org/10.1109/TAC.2005.860290

[3] Audet C, Hare W. Derivative-free and blackbox optimization. Springer; 2017. ISBN: 9783319689128.

[4] Back T. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford University Press; 1996. ISBN: 9780195099713.

[5] Bemporad A. Global optimization via inverse distance weighting and radial basis functions. Comput Optim Appl 2020;77:571–95. https://doi.org/10.1007/s10589-020-00215-w

[6] Bemporad A, Piga D. Global optimization based on active preference learning with radial basis functions. Mach Learn 2021;110:417–48. https://doi.org/10.1007/s10994-020-05935-y

[7] Benosman M. Passive fault tolerant control. Robust Control Theory and Applications; 2011. ISBN: 9789533072296. https://doi.org/10.5772/14334
[8] Bishop CM. Pattern recognition and machine learning. Springer; 2006. ISBN: 9780387310732.
[9] Conn A, Gould N, Toint P. A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. Math Comput 1997;66:261–88. https://doi.org/10.1090/S0025-5718-97-00777-1
[10] Conn AR, Gould NIM, Toint P. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. SIAM J Numer Anal 1991;28:545–72. https://doi.org/10.1137/0728030
[11] Costello GA. Theory of wire rope. Springer; 1997. ISBN: 9780387982021.
[12] Doyle JC. Structured uncertainty in control system design. In: 1985 24th IEEE conference on decision and control; 1985. p. 260–5. https://doi.org/10.1109/CDC.1985.268842
[13] Fan M, Tits AL, Doyle JC. Robustness in the presence of mixed parametric uncertainty and unmodeled dynamics. IEEE Trans Autom Control 1991;36:25–38. https://doi.org/10.1109/9.62265
[14] Gu D-W, Petkov PH, Konstantinov MM. Robust control design with MATLAB®. Springer; 2005. ISBN: 9788181285126.
[15] Gutmann H-M. A radial basis function method for global optimization. J Glob Optim 2001;19:201–27. https://doi.org/10.1023/A:1011255519438
[16] Jones DR. A taxonomy of global optimization methods based on response surfaces. J Glob Optim 2001;21:345–83. https://doi.org/10.1023/A:1012771025575
[17] Jones DR, Perttunen CD, Stuckman BE. Lipschitzian optimization without the lipschitz constant. J Optim Theory Appl 1993;79:157–81. https://doi.org/10.1007/BF00941892
[18] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks. IEEE; 1995. p. 1942–8. https://doi.org/10.1109/ICNN.1995.488968
[19] Lanzon A. Weight optimisation in $\mathcal{H}_\infty$ loop-shaping. Automatica 2005;41:1201–8. https://doi.org/10.1016/j.automatica.2005.01.010
[20] Lanzon A, Richards R. A frequency domain optimisation algorithm for simultaneous design of performance weights and controllers in /spl mu/-synthesis. In: Proceedings of the 38th IEEE conference on decision and control, vol. 5. 1999. p. 4523–8. https://doi.org/10.1109/CDC.1999.833254
[21] Liu GP, Yang JB, Whidborn JF. Multiobjective optimisation and control. Research Studies Press Ltd; 2002. ISBN: 9780863802645.
[22] Martí R, Lozano JA, Mendiburu A, Hernando L. Multi-start methods. Handbook of heuristics 2018:155–75. https://doi.org/10.1007/0-306-48056-5_12
[23] McKay MD, Beckman RJ, Conover WJ. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 2000;42:55–61. https://doi.org/10.2307/1268522
[24] Neumaier A. Complete search in continuous global optimization and constraint satisfaction. Acta Numer 2004;13:271–369. https://doi.org/10.1017/S0962492904000194
[25] Nocedal J, Wright SJ. Numerical optimization. Springer; 1999. ISBN: 0387987932.
[26] Normey-Rico JE, Camacho EF. Control of dead-time processes. Springer; 2007. ISBN: 9781846288289.
[27] Previtali D. Surrogate-based methods for black-box and preference-based optimization in control systems [Ph.D. thesis]. University of Bergamo; 2024. https://doi.org/10.13122/978-88-97413-93-6. ISBN: 9788897413936.
[28] Previtali D, Mazzoleni M, Ferramosca A, Previdi F. GLISp-r: a preference-based optimization algorithm with convergence guarantees. Comput Optim Appl 2023;86:383–420. https://doi.org/10.1007/s10589-023-00491-2
[29] Previtali D, Mazzoleni M, Valceschini N, Previdi F. Data-driven mixed-sensitivity structured control of SISO multi-model systems with application to a reconfigurable industrial oven. In: 23rd european control conference (ECC); 2025. https://doi.org/10.23919/ECC65951.2025.11187191
[30] Regis RG. A survey of surrogate approaches for expensive constrained black-box optimization. In: World congress on global optimization. Springer; 2019. p. 37–47. https://doi.org/10.1007/978-3-030-21803-4_4
[31] Regis RG, Shoemaker CA. Constrained global optimization of expensive black box functions using radial basis functions. J Glob Optim 2005;31:153–71. https://doi.org/10.1007/s10898-004-0570-0
[32] Regis RG, Shoemaker CA. A stochastic radial basis function method for the global optimization of expensive functions. INFORMS J Comput 2007;19:497–509. https://doi.org/10.1287/ijoc.1060.0182
[33] Skogestad S, Postlethwaite I. Multivariable feedback control: analysis and design, vol. 2. Wiley New York; 2007. ISBN: 9780470011683.
[34] Torn A, Zilinskas A. Global optimization. Lecture Notes in Computer Science; 1989. ISBN: 9783540508717.
[35] Valceschini N, Mazzoleni M, Formentin S, Previdi F. Data-driven mixed-sensitivity control with automated weighting functions selection. Int J Robust Nonlinear Control 2023;33:3458–70. https://doi.org/10.1002/rnc.6579
[36] Vu KK, d'Ambrosio C, Hamadi Y, Liberti L. Surrogate-based methods for black-box optimization. Int Trans Oper Res 2017;24:393–424. https://doi.org/10.1111/itor.12292
[37] Young PM. Controller design with real parametric uncertainty. Int J Control 1996;65:469–509. https://doi.org/10.1080/00207179608921707
[38] Zhou K, Doyle JC. Essentials of robust control, vol. 104. Prentice Hall Upper Saddle River, NJ; 1998. ISBN: 9780135258330.
[39] Zhou K, Doyle JC, Glover K. Robust and optimal control. USA: Prentice-Hall, Inc.; 1996. ISBN: 9780134565675.
[40] Zhu M, Piga D, Bemporad A. C-glisp: preference-based global optimization under unknown constraints with applications to controller calibration. IEEE Trans Control Syst Technol 2021. https://doi.org/10.1109/TCST.2021.3136711