Trading-off safety, exploration, and exploitation in learning-based optimization: a Set Membership approach

Lorenzo Sabug Jr., Fredy Ruiz, and Lorenzo Fagiano

***Abstract*— We propose a technique for global optimization considering black-box cost function and constraints, which have to be learned from data during the optimization process, arising for example in plant-control co-design of complex systems or controller tuning based on experiments. Assuming Lipschitz continuity of the cost function and constraints, we build a surrogate model and derive tight bounds on such functions based on a Set Membership framework. An exploitation step is designed to improve on the current best feasible candidate solution, searching in regions where all constraints are estimated as fulfilled, thus preserving safety. On the other hand, an exploration routine aims to discover the shape of the cost and constraint functions by picking points with large uncertainty, prioritizing regions where more constraints are predictably satisfied. The proposed algorithm can intuitively trade-off safety, exploration, and exploitation. The performance is evaluated on the problem of model predictive control tuning for a servomechanism with plant uncertainties and task-level constraints.**

I. Introduction

Black-box optimization techniques address problems where the objective function is not available analytically, rather it is evaluated via (possibly time-consuming) experiments, and is generally non-convex. These methods attempt to simultaneously learn the function and optimize it, balancing exploitation and exploration to ensure fast convergence and coverage of the search space, respectively, [1]–[6].

Many black-box optimization problems require also the satisfaction of one or more constraints. In most approaches, these are assumed to be analytically available, for example as known convex sets or nonlinear differentiable constraint functions. However, in many cases, the constraints are also not known a priori or too complex to be handled analyticallyi.e., they are black box, too. In this paper, we consider a framework in which sensible objective and constraints' values are returned also when the test point is outside the feasible set. This is in contrast to non-relaxable problems, in which a violated constraint means a crashed simulation or a failed experiment, and the returned objective and/or constraint values are undefined or not meaningful. Using the taxonomy of black-box constraints introduced in [7], we deal with `QRSK` problems (Quantifiable, Relaxable, Simulation, Known).

All authors are with Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milan, Italy Email: {lorenzojr.sabug, fredy.ruiz, lorenzo.fagiano}@polimi.it

This research has been supported by the Italian Ministry of University and Research (MIUR) under the PRIN 2017 grant n. 201732RS94 "Systems of Tethered Multicopters"

The first author would like to acknowledge the support of the Department of Science and Technology–Science Education Institute (DOST-SEI) of the Philippines for his research

There have been relatively scarce previous efforts in treating black-box constraints in optimization, with most of the proposed techniques based on Bayesian optimization (BO). Early attempts to integrate black-box constraints with BO, like [8], require a feasible initial point, which may be limiting and require an ad-hoc initial phase to estimate the feasible region. For example, [9] deals with black-box constraints using support vector machines, in two phases. The first one is the "training phase", which estimates the feasible regions by evaluating the constraint at different points. In the second phase, a Bayesian optimizer is used to optimize over the estimated feasible space. While sensible with cheap-to-evaluate constraint functions, a separate training phase may significantly increase the experiment time before the actual optimization is attempted.

The proposal from [10], on the other hand, uses a sum of the objective and constraint values in the acquisition function, which removes the need for existence of a feasible solution. Another approach [11] employs the alternating direction method of multipliers (ADMM) in conjunction with BO, which recasts the constrained problem into an unconstrained formulation and introduces an ADMM-inherited stopping criterion. However, an overarching concern with all of the discussed techniques, as well as other BO-based methods, is their limitation only to low dimensions (usually up to 5-6 decision variables), mainly due to the computational burden required by Gaussian modelling for BO.

The present paper proposes a new approach to deal with black-box constraints in optimization. Furthering our recent work in [5], [6], we use the Set Membership framework [12] to build models of both the cost and the constraints. These approximations are used to predict the fulfillment of the black-box constraints in unsampled areas. Our design of the exploration routine allows the user to tune the exploration behavior, ranging from "cautious" to "risky". Furthermore, we do not require an initial feasible point, rather we automatically prioritize regions where more constraints are estimated as fulfilled, encouraging the search inside the feasible region. The proposed technique is described, and tested for the tuning of a model predictive controller for a servomechanism with plant parameter uncertainties and task-level constraints.

II. Problem Statement

We consider the minimization of a cost function $z = f(\boldsymbol{x}),\ f(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$, where the decision variable $\boldsymbol{x}$ is an element of a compact and convex search set $\mathcal{X} \subset \mathbb{R}^D$. This minimization is subject to the constraints $g_s, s = 1, \ldots, S$. A constraint g_s is satisfied at $\boldsymbol{x}$ when $g_s(\boldsymbol{x}) \geq 0$.

Neither f nor any g_s are assumed to be known. The only *a priori* assumptions about f and all g_s are given as follows:

Assumption 1: f and $g_s, s = 1, \ldots, S$ are Lipschitz continuous functions over $\mathcal{X}$ with unknown Lipschitz constants $\gamma, \rho_1, \ldots, \rho_S : f \in \mathcal{F}(\gamma), g_1 \in \mathcal{F}(\rho_1) \ldots g_S \in \mathcal{F}(\rho_S)$ where $\mathcal{F}(\eta) \doteq \{h : |h(\boldsymbol{x}_1) - h(\boldsymbol{x}_2)| \leq \eta \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}\}$.

Assumption 2: The values of the objective and of all constraints can be obtained without noise for any $\boldsymbol{x} \in \mathcal{X}$.

Both assumptions are realistic in many cases of practical interest, such as simulation-based optimization or experiments where sensors return accurate and valid measurements also when the black-box constraints are violated.

Finally, denoting the feasible set pertaining to the s-th constraint as $\mathcal{G}_s = \{\boldsymbol{x} \in \mathcal{X} : g_s(\boldsymbol{x}) \geq 0\}$, we assume a non-empty finite feasible region:

Assumption 3:

$$\bigcap_{s=1}^{S} \mathcal{G}_s \neq \varnothing.$$

Now we state the problem addressed in this paper.

Problem 1: Design an algorithm that generates a sequence of points $\left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots\right\}, \boldsymbol{x}^{(i)} \in \mathcal{X}$, to search for a minimizer point $\boldsymbol{x}^*$ of f, such that

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in \bigcap_{s=1}^{S} \mathcal{G}_s$$

III. Proposed Optimization Approach with Black-Box Constraints

The search for the minimizer $\boldsymbol{x}^*$ is approached by a sequential sampling procedure, wherein the next point for sampling is decided by the existing data. The inclusion of black-box constraints means that there are additional considerations in the algorithm design, as discussed in the following subsections.

A. Evaluation of objective and constraints, data update

At each iteration n, the evaluation of the objective and the constraints is performed at the test point $\boldsymbol{x}^{(n)}$. We denote the obtained cost and constraints values as:

$$\begin{aligned} z^{(n)} &= f(\boldsymbol{x}^{(n)}) \\ c_s^{(n)} &= g_s(\boldsymbol{x}^{(n)}),\ s = 1, \ldots, S \end{aligned}$$

According to Assumption 2, these values can be evaluated for any $\boldsymbol{x}^{(n)} \in \mathcal{X}$, and the corresponding data tuple $(\boldsymbol{x}^{(n)}, z^{(n)}, c_1^{(n)}, \ldots, c_S^{(n)})$ is iteratively added to build the data-set $\boldsymbol{X}^{\langle n \rangle}$, i.e.

$$\boldsymbol{X}^{\langle n \rangle} = \boldsymbol{X}^{\langle n-1 \rangle} \cup (\boldsymbol{x}^{(n)}, z^{(n)}, c_1^{(n)}, \ldots, c_S^{(n)}).$$

From the existing data-set $\boldsymbol{X}^{\langle n \rangle}$, we update iteratively the Lipschitz constants estimates $\gamma^{\langle n \rangle}$ and $\rho_s^{\langle n \rangle}$ (see Assumption 1) as in [5]:

$$\gamma^{\langle n \rangle} = \max\left(\gamma^{\langle n-1 \rangle}, \max_{k \in [1\ \ldots\ n-1]} \frac{|z^{(n)} - z^{(k)}|}{\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^{(k)}\|}\right), \quad (1)$$

$$\rho_s^{\langle n \rangle} = \max\left(\rho_s^{\langle n-1 \rangle}, \max_{k \in [1\ \ldots\ n-1]} \frac{|c_s^{(n)} - c_s^{(k)}|}{\|\boldsymbol{x}^{(n)} - \boldsymbol{x}^{(k)}\|}\right). \quad (2)$$

where $\gamma^{\langle n-1 \rangle}$, $\rho_s^{\langle n-1 \rangle}$ are the estimates computed at iteration $n-1$. The estimated Lipschitz constants are, by construction, unfalsified by the available data. At $n = 1$, one can set these estimates to zero, and choose the test point $\boldsymbol{x}^{(1)}$ with a strategy of choice (for example a random starting point, or, as discussed in our example, a sensible point for the application at hand). We can now build the following upper- and lower-bound functions, $\overline{f}^{\langle n \rangle}(\boldsymbol{x})$ and $\underline{f}^{\langle n \rangle}(\boldsymbol{x})$, resorting to a Set Membership (SM) approach [12]:

$$\overline{f}^{\langle n \rangle}(\boldsymbol{x}) \triangleq \min_{k=1,\ldots,n} \left(z^{(k)} + \mu\gamma^{\langle n \rangle}\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|\right), \quad (3)$$

$$\underline{f}^{\langle n \rangle}(\boldsymbol{x}) \triangleq \max_{k=1,\ldots,n} \left(z^{(k)} - \mu\gamma^{\langle n \rangle}\|\boldsymbol{x} - \boldsymbol{x}^{(k)}\|\right). \quad (4)$$

where $\mu > 1$ is introduced due to $\gamma^{\langle n \rangle}$ and $\rho_s^{\langle n \rangle}$ being underestimates of true Lipschitz constants γ and ρ_s. Furthermore, we build the central approximation of the objective function,

$$\tilde{f}^{\langle n \rangle}(\boldsymbol{x}) = \frac{1}{2}\left(\overline{f}^{\langle n \rangle}(\boldsymbol{x}) + \underline{f}^{\langle n \rangle}(\boldsymbol{x})\right)$$

and the uncertainty measure

$$\lambda^{\langle n \rangle}(\boldsymbol{x}) = \overline{f}^{\langle n \rangle}(\boldsymbol{x}) - \underline{f}^{\langle n \rangle}(\boldsymbol{x}).$$

The same is performed for each constraint function g_s, to calculate the corresponding upper and lower bounds, $\overline{g}_s^{\langle n \rangle}(\boldsymbol{x})$, $\underline{g}_s^{\langle n \rangle}(\boldsymbol{x})$, nominal estimate, $\tilde{g}_s^{\langle n \rangle}(\boldsymbol{x})$, and uncertainty interval, $\pi_s^{\langle n \rangle}(\boldsymbol{x})$.

The tuple describing the best sample $\boldsymbol{x}^{*\langle n \rangle}$ is

$$\boldsymbol{x}^{*\langle n \rangle} = \underset{\boldsymbol{x}^{(i)} \in \boldsymbol{X}^{\langle n \rangle}}{\arg\min}\ z^{(i)} \text{s.t. } c_s^{(i)} \geq 0, s = 1, \ldots, S, \quad (5)$$

which is the feasible sample with lowest objective value. A lexicographic criterion is used to sort out possible multiple feasible points with the same cost. Note that $\boldsymbol{x}^{*\langle n \rangle}$ might not exist particularly in the iterations where no feasible points were sampled yet.

After updating the Lipschitz constant estimates, the upper and lower bounds, central estimates, uncertainty estimates, and current best point, the proposed optimization algorithm attempts first an exploitation strategy, possibly followed (if not satisfactory enough) by an exploration one.

B. Exploitation with constraints

In this subroutine, we try to improve on the current best value by searching $\mathcal{X}$ for a better candidate point according to a multi-objective optimization problem involving the central approximation and estimated uncertainty, while satisfying the estimated constraints:

$$\begin{aligned} \boldsymbol{x}_\theta^{\langle n \rangle} = &\underset{\boldsymbol{x} \in \boldsymbol{E}^{\langle n \rangle}}{\arg\min} \left(\tilde{f}^{\langle n \rangle}(\boldsymbol{x}) - \beta\lambda^{\langle n \rangle}(\boldsymbol{x})\right) \\ &\text{s.t. } \tilde{g}_s^{\langle n \rangle}(\boldsymbol{x}) \geq 0, s = 1, \ldots, S \end{aligned} \quad (6)$$

where β is a user-defined weighting parameter (set here as 0.1), and $\boldsymbol{E}^{\langle n \rangle} \in \mathcal{X}$ is a finite set of candidate points, selected according to a gridding strategy described in Section III-D. If (6) is feasible and a $\boldsymbol{x}_\theta^{\langle n \rangle}$ is selected, we test its expected improvement with respect to the current best value. As in [6], we test if the lower bound $\underline{z}^{\langle n \rangle}(\boldsymbol{x}_\theta^{\langle n \rangle})$ improves w.r.t. $z^{*\langle n \rangle}$ by a certain threshold, i.e.

$$\underline{z}^{\langle n \rangle}(\boldsymbol{x}_\theta^{\langle n \rangle}) \leq z^{*\langle n \rangle} - \eta, \quad (7)$$

where $\eta = \alpha\gamma^{\langle n \rangle}$ is the expected improvement threshold. If this test is passed, we sample the selected point, i.e. $\boldsymbol{x}^{(n+1)} = \boldsymbol{x}_{\theta}^{\langle n \rangle}$. On the other hand, if (6) is infeasible or condition (7) is not met, we proceed to exploration.

C. Exploration by uncertainty and constraint violations

The exploration routine tries to probe the areas of the search space where cost function uncertainty is largest, while at the same time penalizing possible constraint violations. In the presence of black-box constraints, we prioritize choosing feasible candidate points, i.e. where all constraints are expected to be fulfilled using the central approximation. The exploration point $\boldsymbol{x}_{\psi}^{\langle n \rangle}$ is chosen as

$$\boldsymbol{x}_{\psi}^{\langle n \rangle} = \underset{\boldsymbol{x} \in \boldsymbol{E}^{\langle n \rangle}}{\arg\max} \left((1-\delta) w_{\lambda}(\boldsymbol{x}) + \delta w_{\pi}(\boldsymbol{x}) w_g(\boldsymbol{x}) \right) \tag{8}$$

where

$$w_{\lambda}(\boldsymbol{x}) = \begin{cases} \lambda^{\langle n \rangle}(\boldsymbol{x}) & \text{if } \tilde{g}_s^{\langle n \rangle}(\boldsymbol{x}) \geq 0, s = 1, \dots, S \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

$$w_{\pi}(\boldsymbol{x}) = \sum_{s=1}^{S} \frac{\pi_s^{\langle n \rangle}(\boldsymbol{x})}{\rho_s^{\langle n \rangle}}, \tag{10}$$

$$w_g(\boldsymbol{x}) = \prod_{s=1}^{S} \left(\mathbb{1}_s(\boldsymbol{x}) + 1 \right), \tag{11}$$

$$\mathbb{1}_s(\boldsymbol{x}) = \begin{cases} 1 & \tilde{g}_s^{\langle n \rangle}(\boldsymbol{x}) \geq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The objective w_{λ} in (8) aims to find the test point where uncertainty λ is maximum, but this term is factored in only if the candidate point fulfills all estimated constraints (see (9)). On the other hand, the right-hand expression inside the $\arg\max$ in (8) finds the candidate point with the maximum total (normalized) uncertainty w.r.t. all g_s (see (10)). However, we design it to double for every additional constraint fulfilled, in force of (11)-(12). This way, the selection prioritizes points that are predicted to satisfy a larger number of constraints, according to the information collected up to iteration n. The user parameter $\delta \in (0, 1]$ describes the level of conservativeness or caution in exploring outside the estimated feasible region, with $\delta \to 0$ resulting to higher conservativeness.

The selected point $\boldsymbol{x}_{\psi}^{\langle n \rangle}$ is then directly assigned as the next sample point $\boldsymbol{x}^{(n+1)}$.

D. Generation of candidate points

In Sections III-B-III-C we consider a set of candidate points $\boldsymbol{E}^{\langle n \rangle}$, which is systematically generated based on the existing data. Such generation eliminates the need for an external optimization algorithm to solve (6) and (8) and makes the algorithm repeatable.

The candidate points generation method proposed in our previous works [5], [6] results in a exponential complexity w.r.t. D in the common case of hyperrectangular $\mathcal{X}$. Hence we propose another candidate points generation system with improved complexity.

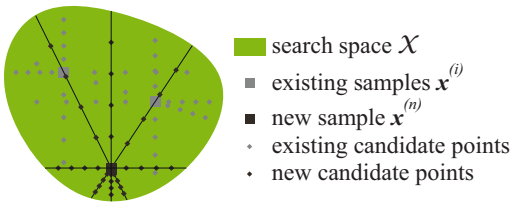

Fig. 1: Proposed method for introduction of new candidate points ($D = 2$ and $B = 5$)

Given an incoming sample $\boldsymbol{x}^{(n)}$, we iteratively add candidate points

1) along the coordinate directions stemming from $\boldsymbol{x}^{(n)}$,
2) in the directions pointing from each sampled point to all the other ones.

To generate 1), using the basic directions $\pm\hat{\boldsymbol{a}}_d, d = 1, \dots, D$, we define

$$\boldsymbol{b}_{+d}^{(n)} = \max_{b \in [0, \infty)} b \text{ s.t. } \boldsymbol{x}^{(n)} + b\hat{\boldsymbol{a}}_d \in \mathcal{X}, \tag{13}$$

$$\boldsymbol{b}_{-d}^{(n)} = \max_{b \in [0, \infty)} b \text{ s.t. } \boldsymbol{x}^{(n)} - b\hat{\boldsymbol{a}}_d \in \mathcal{X}. \tag{14}$$

These are the lengths of the longest segments inside the search set $\mathcal{X}$ departing from $\boldsymbol{x}^{(n)}$. Given the previous segment lengths, the set of candidate points generated by $\boldsymbol{x}^{(n)}$ along the coordinate directions are

$$\boldsymbol{Y}_{+d}^{(n)} = \left\{ \boldsymbol{x}^{(n)} + \frac{k}{B} \boldsymbol{b}_{+d}^{(n)} \hat{\boldsymbol{a}}_d, k \in \{1, \dots, B-1\} \right\} \tag{15}$$

$$\boldsymbol{Y}_{-d}^{(n)} = \left\{ \boldsymbol{x}^{(n)} - \frac{k}{B} \boldsymbol{b}_{-d}^{(n)} \hat{\boldsymbol{a}}_d, k \in \{1, \dots, B-1\} \right\} \tag{16}$$

where B is the grid granularity.

To generate 2), we denote $\hat{\boldsymbol{a}}^{(nj)}$ as the unit vector from $\boldsymbol{x}^{(n)}$ to each other existing sample $\boldsymbol{x}^{(j)}, j = 1, \dots, n-1$. Then, in a similar fashion as (13)-(14), we define

$$\boldsymbol{b}_{+}^{(nj)} = \max_{b \in [0, \infty)} b \text{ s.t. } \boldsymbol{x}^{(n)} + b\hat{\boldsymbol{a}}^{(nj)} \in \mathcal{X}, \tag{17}$$

$$\boldsymbol{b}_{-}^{(nj)} = \max_{b \in [0, \infty)} b \text{ s.t. } \boldsymbol{x}^{(n)} - b\hat{\boldsymbol{a}}^{(nj)} \in \mathcal{X}, \tag{18}$$

and the set of candidate points generated by $\boldsymbol{x}^{(n)}$ along the segments pointing to each other sample are

$$\boldsymbol{Y}_{+}^{(nj)} = \left\{ \boldsymbol{x}^{(n)} + \frac{k}{B} \boldsymbol{b}_{+}^{(nj)} \hat{\boldsymbol{a}}_d, k \in \{1, \dots, B-1\} \right\}, \tag{19}$$

$$\boldsymbol{Y}_{-}^{(nj)} = \left\{ \boldsymbol{x}^{(n)} - \frac{k}{B} \boldsymbol{b}_{-}^{(nj)} \hat{\boldsymbol{a}}_d, k \in \{1, \dots, B-1\} \right\}. \tag{20}$$

Finally, the set of candidate points generated by $\boldsymbol{x}^{(n)}$ is

$$\boldsymbol{Y}^{(n)} = \left(\bigcup_{d=1}^{D} \left\{ \boldsymbol{Y}_{+d}^{(n)}, \boldsymbol{Y}_{-d}^{(n)} \right\} \right) \bigcup \left(\bigcup_{j=1}^{n-1} \left\{ \boldsymbol{Y}_{+}^{(nj)}, \boldsymbol{Y}_{-}^{(nj)} \right\} \right). \tag{21}$$

Lastly, the aggregate collection of candidate points at iteration n is now

$$\boldsymbol{E}^{\langle n \rangle} = \boldsymbol{E}^{\langle n-1 \rangle} \cup \boldsymbol{Y}^{(n)}.$$

This results in a total of $2n(B-1)(D + \frac{(n-1)}{2})$ candidate points for any iteration n, which is $\mathcal{O}(Dn^2)$. The number of candidate points is polynomial w.r.t. n, however only linear in D, thus providing a good tradeoff between complexity and accuracy. Furthermore, unlike our proposal in [6], we do not require anymore a polytopic search set $\mathcal{X}$, but only a convex and compact one.

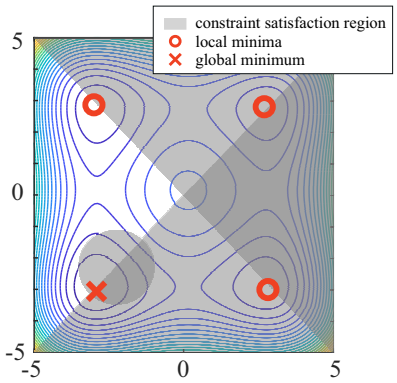

Fig. 2: Top view of Styblinski-Tang function, with constraint satisfaction regions

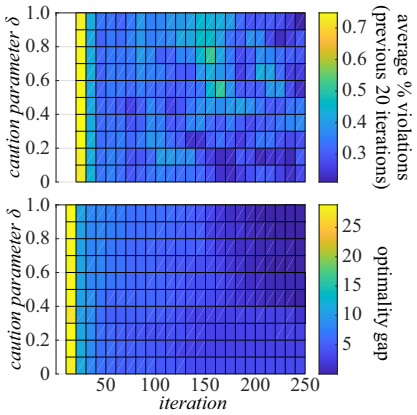

Fig. 3: Effect of increasing δ: (top) frequency of violations, (bottom) optimality gap

E. On the influence of caution parameter δ

We illustrate the effect of changing the "caution parameter" δ by testing SMGO-δ for the constrained optimization of the 2-dimensional Styblinski-Tang function:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \frac{1}{2}\sum_{i=1}^{2}\left(x_i^4 - 16x_i^2 + 5x_i\right) \tag{22}$$

$$\text{s.t.} \quad \max\left(4 - \sum_{i=1}^{2}(x_i+2)^2, \sum_{i=1}^{2} x_i\right) \geq 0, \tag{23}$$

$$x_1 - x_2 \geq 0, \tag{24}$$

with search space $\mathcal{X} = [-5\ 5]^2$. The global minimum is located in a disjoint half-circle on the lower left side, see Fig. 2. In this simple test we vary δ from 10^{-6} to 1, in increments of 0.1. For each tested δ, 50 independent optimization runs/trials (250 iterations) with random starting points are performed. In Fig. 3 we show the frequency of constraint violation w.r.t. n and δ. For this purpose, we count the average number of violation occurrences (i.e. $\boldsymbol{x}^{(n)} \notin \bigcap_{s=1}^{2}\mathcal{G}_s$) in the previous 20 iterations, starting from $n = 20$. Furthermore, we present the optimality gap w.r.t. $z^* = -78.332$ in Fig. 3 as well.

Caution parameter value $\delta =$ 1E-6 (marked in the axis as 0) resulted in least frequency of constraint violations. Violations increase w.r.t. larger δ, represented by brighter colors towards the upper right on the top graph of Fig. 3. On the other hand, higher δ led to better optimality gaps, which means that they are more likely to find the $\boldsymbol{x}^*$ in the disjoint semi-circle. Hence, the user can consider a trade-off between discovery of other feasible regions (and a chance at better minima, "high risk, high reward"), and the need to stay in a feasible region.

IV. Application to MPC Tuning

Consider a servomotor mechanism, where θ_i and θ_o are the input and output shaft angular positions, V is the input voltage, T is the torsional moment between the input and output shafts, d is the load (disturbance) on the output shaft. R_m is the motor electrical resistance, K_t is the motor constant, J_i and J_o the moments of inertia of the input and output shafts, β_i and β_o the shaft viscous friction coefficients, τ_g is the gear ratio, and K_θ is the torsional stiffness of the output shaft. Their nominal values are reported in Table I.

Assuming linearity of all the components, the system dynamics are described by the following equations (the continuous time variable is omitted for brevity):

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{-K_\theta}{J_i\tau_g^2} & \frac{-K_\theta}{J_i\tau_g} & -\frac{\beta_i+\frac{K_t^2}{R_m}}{J_i} & 0 \\ \frac{K_\theta}{J_o\tau_g} & \frac{-K_\theta}{J_o} & 0 & \frac{-\beta_o}{J_o} \end{bmatrix}\boldsymbol{\xi} + \begin{bmatrix} 0 \\ 0 \\ \frac{K_t}{J_i R_m} \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_o} \end{bmatrix} d, \tag{25}$$

$$y = \begin{bmatrix} \frac{K_\theta}{\tau_g} & -K_\theta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\boldsymbol{\xi} \tag{26}$$

where $\boldsymbol{\xi} = [\theta_i\ \theta_o\ \dot{\theta}_i\ \dot{\theta}_o]^T$ is the state, and $\boldsymbol{y} = [T\ \theta_o\ \dot{\theta}_i\ \dot{\theta}_o]^T$ the output.

A. Controller structure

We aim to design a model-predictive controller (MPC, [13]) capable of tracking a reference $\hat{\theta}_o$ for the output shaft angle. An MPC strategy is chosen due to the explicit consideration of input and state constraints.

After setting a sampling time $T_s = 0.1$ s, at each time step t the following finite horizon optimal control problem is solved:

$$\min_{U\in\mathbb{R}^N} \sum_{i=0}^{N}(y_{ref} - y(i|t))^T\boldsymbol{Q}(y_{ref} - y(i|t)) + \sum_{i=0}^{N-1} Ru(i|t)^2 \tag{27}$$

$$\text{s.t.} \quad \boldsymbol{\xi}(i+1|t) = \boldsymbol{A}\boldsymbol{\xi}(i|t) + Bu(i|t) + B_d d(i|t) \tag{28}$$

$$y(i|t) = \boldsymbol{C}\boldsymbol{x}(i|t) \tag{29}$$

$$\boldsymbol{\xi}(0|t) = \boldsymbol{\xi}(t) \tag{30}$$

$$\left|u(i|t)\right| \leq \overline{V} \tag{31}$$

$$\left|T(i|t)\right| \leq \tilde{T}, \tag{32}$$

where for each signal, the time instant $(i|t)$ indicates the value of the signal at time $t+i$ predicted at time t, (28) is the discrete-time equivalent of (25) considering a sampling time of 0.1 s, (29) is the output equation given in (26), (29) is the known system state at time step t, (31) declares the input constraint with $\overline{V} = 220$ V, and (32) limits the torsional

Variable	Description	Nominal value
R_m	Motor electrical resistance	20 Ω
K_t	Motor constant	10 $\frac{Nm}{A}$
K_θ	Output shaft torsional stiffness	1280 $\frac{Nm}{rad}$
J_i	Input shaft moment of inertia	0.5 $kg\ m^2$
J_o	Output shaft moment of inertia	25 $kg\ m^2$
β_i	Input shaft friction coefficient	0.1 $\frac{Nm\ s}{rad}$
β_o	Output shaft friction coefficient	25 $\frac{Nm\ s}{rad}$
τ_g	Gear ratio (input/output)	20

TABLE I: Servo motor specifications

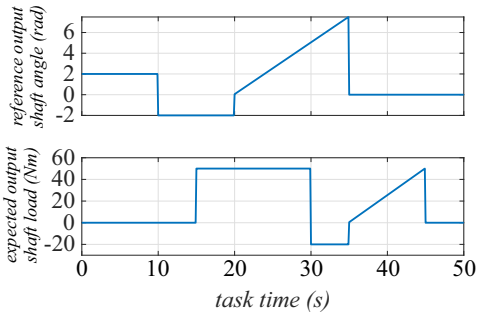

Fig. 4: Reference trajectory and expected load for the servomotor problem

moment between the gears. N is the prediction and control horizon, selected as $N = 50$. Matrix $\boldsymbol{Q}$ is a diagonal matrix of tunable parameters q_1, q_2, q_3, q_4, while R is fixed as 1.

B. Black-box optimization for controller tuning

In the context of factory automation, we encounter machines performing repetitive tasks that usually involve a fixed reference trajectory $\hat{\theta}_o$ and load profile d. Fig. 4 shows the task profile we consider, having a duration of 50 s.

Once the controller architecture is defined, we aim to tune the MPC parameters to guarantee good tracking performance during the complete task, considering the following practically-motivated issues:

1) MPC cost function matrix $\boldsymbol{Q}$ needs to be tuned to maximize tracking performance on $\hat{\theta}_o$.
2) The nameplate (nominal) motor parameters are known (as in Table I); however, the real parameters differ due to manufacturing tolerances and/or previous usage. In this paper, $R_m, K_t, \beta_i, \beta_o$ may differ up to $\pm 12.5\%$ w.r.t. nameplate values. The remaining parameters are assumed exact.
3) To protect the motor windings and prolong its service lifetime, the average power consumption of the system throughout the task cycle, must not exceed a maximum $\overline{P}$, set as 25 W. At the same time, constraints (31) and (32) must be satisfied for the complete task duration.

Based on the previous considerations, the following black-box optimization problem is formulated:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \sum_{t=0}^{t_{max}} (\theta_o(t;\boldsymbol{x}) - \hat{\theta}_o(t))^2 \tag{33}$$

$$\text{s.t.} \max_{0\le t\le t_{max}} \left|T(t)\right| \le \overline{T}, \tag{34}$$

$$\frac{1}{t_{max}+1} \sum_{t=0}^{t_{max}-1} \frac{u(t)\left(u(t) - K_t\dot{\theta}_i(t)\right)}{R_m} \le \overline{P} \tag{35}$$

where the vector of decision variables is $\boldsymbol{x} = [log(q_1)\ \ log(q_2)\ \ log(q_3)\ \ log(q_4)\ \ \tilde{R}_m\ \ \tilde{K}_t\ \ \tilde{\beta}_i\ \ \tilde{\beta}_o\ \ \tilde{T}]$. The tilde on 4th to 8th variables of $\boldsymbol{x}$ denote parameter estimates, while $\tilde{T}$ is the imposed torsional moment limit in (32). $\theta_o(t;\boldsymbol{x})$ is the resulting output shaft angular position during the complete task, when the MPC controller is applied to the servo system using the parameters $\boldsymbol{x}$. Constraint (34) sets the maximum torsional moment for the experiment, while (35) imposes the maximum average motor power for the entire duration.

The search intervals for the first four parameters is $[-7,\ 7]$, while intervals $\pm 25\%$ w.r.t. nominal values are imposed for the next four (see Table I for the nominal values), and $[50\ 100]$ is fixed for $\tilde{T}$. 50 trials of the SMGO-δ algorithm are executed with $N = 250$ iterations each. The starting test point $\boldsymbol{x}^{(1)}$ for each optimization is composed of $log(q_i) = 0$, the nominal values for each motor parameter, and $\overline{T}$ for $\tilde{T}$.

For comparison, we also consider a Finite Horizon Optimal Control Problem (FHOCP) that optimizes the input sequence over the whole task duration, having complete knowledge of the actual plant parameters. In the same manner as (33), the FHOCP solves

$$\min_{u(0),\ldots,u(t_{max})} \sum_{t=0}^{t_{max}} (\theta_o(t) - \hat{\theta}_o(t))^2$$

considering the constraints (34)-(35). The cost value computed by the FHOCP is denoted as z^*_{FHOCP}.

C. Results and discussion

It was observed that after at most 10 iterations, all 50 trials have found feasible points, implying the existence of a best sample $\boldsymbol{x}_\theta^{\langle n\rangle}$. Fig. 5 shows the spread of the cost function for the best (feasible) solutions found across the 50 trials, compared to the FHOCP resulting cost. Furthermore, it shows the corresponding values of the constraints at those best solutions. Note that the cost value improves approaching the optimal solution of the FHOCP and also the constraints exhibit a generally decreasing trend. Indeed, SMGO-δ finds better points which tend to approach the boundary of the feasible set, i.e. resulting in most aggressive control actions but still within the requirements imposed in (34) and (35).

An inspection of the distribution of the estimated plant parameters (across all trials) for the best sample $\boldsymbol{x}_\theta^{\langle n\rangle}$, not shown for brevity, leads to interesting considerations. In fact, the distribution of the estimated parameters does not match at all the real parameters, which shows that the tuning process did not consider the motor parameter estimation accuracy, but only the actual performance based on the experiments. Furthermore, it is interesting to note that while R_m and K_t estimates had discernible changes w.r.t. iterations, β_i and β_o did not change appreciably, with β_o estimates barely changing. This means that the performance is highly sensitive to R_m and K_t estimates, while the β_i and β_o did not have a significant impact.

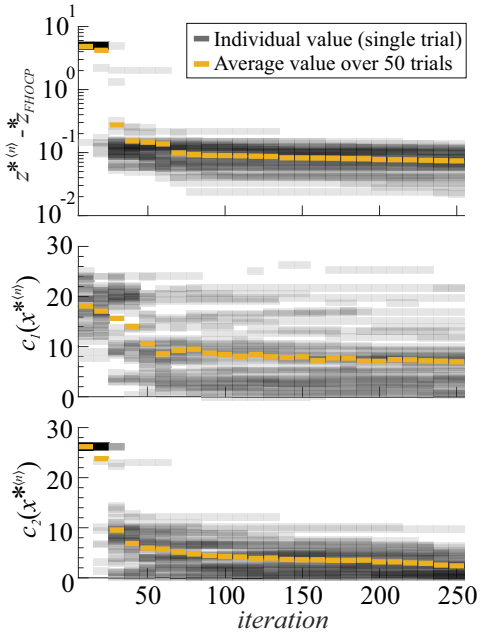

Fig. 5: Distribution of (top) best objective history across 50 runs with randomly-generated motor specifications, (center, bottom) corresponding constraint values for the best point found. Constraints are satisfied if positive.

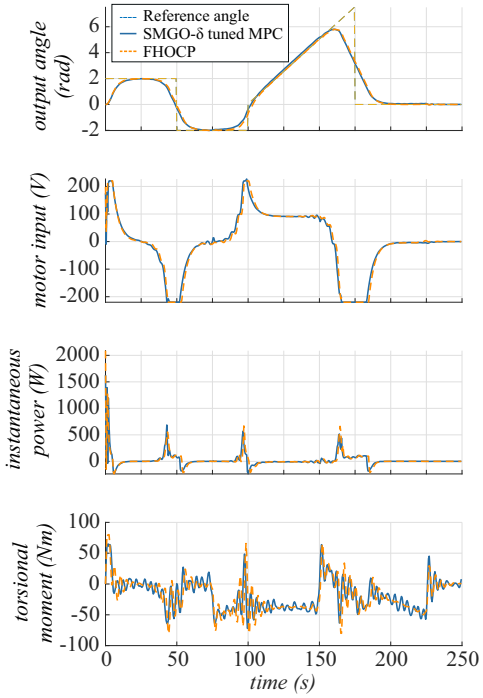

Fig. 6: Comparison of SMGO-δ tuned MPC versus FHOCP for a sample optimization run

For graphical comparison, we consider an example SMGO-δ run and take its best resulting MPC. We now compare its performance to the ideal FHOCP with complete system model information and optimized over the entire task time, which provides a global minimum. Fig. 6 shows that the resulting input trajectory of our optimized MPC did not have significant difference with the FHOCP. However, the torsional moments for the FHOCP showed much larger swings than for our resulting MPC, due to a more aggressive performance. This is in turn caused by the FHOCP pushing into the bounds of feasibility w.r.t. torsional moment constraints. Similar results are observed for the SMGO-δ-optimized MPC of all the other runs and the corresponding FHOCP results, not shown here for brevity.

V. Conclusions and Further Work

We introduce in this paper a black-box optimization algorithm which also considers black-box constraints, named SMGO-δ. Rooted in Set Membership approximation theory, it features exploitation and exploration routines that do not require an initial feasible point, but only that a feasible region exists in the search space. We discussed implementation issues, and presented the performance obtained in an MPC tuning problem for an industrial servomotor system, with plant parameter uncertainties and task-specific constraints that cannot be addressed in the design phase. Our results show that the MPC tuned by SMGO-δ results in performance that is highly comparable to an FHOCP with perfect knowledge of the plant parameters and optimized over the whole process time. Investigation of theoretical properties, as well as experimental applications are the subject of current research.

References

[1] S. Z. Khong, D. Nešić, C. Manzie, and Y. Tan, "Multidimensional global extremum seeking via the direct optimisation algorithm," *Automatica*, vol. 49, no. 7, pp. 1970 – 1978, 2013.

[2] C. Malherbe and N. Vayatis, "Global optimization of Lipschitz functions," *34th International Conference on Machine Learning, ICML 2017*, vol. 5, pp. 3592–3601, 2017.

[3] S. Gao, L. Shi, and Z. Zhang, "A peak-over-threshold search method for global optimization," *Automatica*, vol. 89, pp. 83 – 91, 2018.

[4] A. Bemporad, "Global optimization via inverse distance weighting and radial basis functions," *Computational Optimization and Applications*, vol. 77, no. 2, pp. 571–595, 2020.

[5] L. Sabug, F. Ruiz, and L. Fagiano, "On the use of Set Membership theory for global optimization of black-box functions," in *2020 59th IEEE Conference on Decision and Control (CDC)*, vol. 2020-Decem. IEEE, dec 2020, pp. 3586–3591.

[6] ——, "SMGO: A Set Membership Approach to Data-Driven Global Optimization," *Automatica*, vol. 133, p. 109890, 2021.

[7] S. L. Digabel and S. M. Wild, "A Taxonomy of Constraints in Simulation-Based Optimization," pp. 1–14, 2015. [Online]. Available: http://arxiv.org/abs/1505.07881

[8] M. A. Gelbart, J. Snoek, and R. P. Adams, "Bayesian Optimization with Unknown Constraints," pp. 1–14, mar 2014. [Online]. Available: http://arxiv.org/abs/1403.5607

[9] C. Antonio, "Sequential model based optimization of partially defined functions under unknown constraints," *Journal of Global Optimization*, vol. 79, no. 2, pp. 281–303, 2019.

[10] J. M. Hernández-Lobato, M. A. Gelbart, M. W. Hoffman, R. P. Adams, and Z. Ghahramani, "Predictive entropy search for bayesian optimization with unknown constraints," in *Proc. 32nd Int. Conf. Mach. Learn.*, ser. ICML'15. JMLR.org, 2015, p. 1699–1707.

[11] S. Ariafar, J. Coll-Font, D. Brooks, and J. Dy, "ADMMBO: Bayesian Optimization with Unknown Constraints using ADMM." *Journal of machine learning research : JMLR*, vol. 20, no. 10, pp. 139–148, 2019.

[12] M. Milanese and C. Novara, "Set Membership identification of nonlinear systems," *Automatica*, vol. 40, no. 6, pp. 957–975, 2004.

[13] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.