

Advanced Manufacturing Configuration by Sample-efficient Batch Bayesian Optimization

Xavier Guidetti^{1,2}, Alisa Rupenyan^{1,2}, Lutz Fassl³, Majid Nabavi³, and John Lygeros¹

Abstract—We propose a framework for the configuration and operation of expensive-to-evaluate advanced manufacturing methods, based on Bayesian optimization. The framework unifies a tailored acquisition function, a parallel acquisition procedure, and the integration of process information providing context to the optimization procedure. The novel acquisition function is demonstrated, analyzed and compared on state-of-the-art benchmarking problems. We apply the optimization approach to atmospheric plasma spraying and fused deposition modeling. Our results demonstrate that the proposed framework can efficiently find input parameters that produce the desired outcome and minimize the process cost.

Index Terms—Process Control, Probability and Statistical Methods, Intelligent and Flexible Manufacturing, Machine Learning for Control, Bayesian Optimization

I. INTRODUCTION

IN manufacturing, the optimization of process inputs for a new production task to meet productivity and quality requirements, is a challenging task. This is true especially for processes where the direct outcome of the process inputs comprises several interconnected outputs, whose quality analysis requires time-consuming or destructive measurements. Additive manufacturing, and in general the technologies that deposit material layer by layer on a substrate, are examples of such processes [1]. A sample-efficient, data-driven approach to find optimal process parameters for a manufactured part is thus beneficial, especially when the number of possible trials is restricted by the produced pieces quality assessment costs.

The standard approach for process configuration, control and optimization uses modeling based on statistical principles [2], [3]. Candidate combinations of process parameters are proposed following a full factorial or fractional design of experiments, using optimal orthogonal designs or Taguchi arrays to limit the number of experiments [4]. Then, the best candidates are selected following regression analysis of

Manuscript received: April 29, 2022; Revised August 8, 2022; Accepted September 11, 2022.

This paper was recommended for publication by Editor Ashis Banerjee upon evaluation of the Associate Editor and Reviewers' comments.

This work has been funded by the Swiss Innovation Agency (grant №37896) and by the Swiss National Science Foundation under NCCR Automation (grant №180545).

¹ Xavier Guidetti, Alisa Rupenyan, and John Lygeros are with the Automatic Control Laboratory, ETH Zürich, Switzerland {xaguidetti, ralisa, lygeros}@control.ee.ethz.ch

² Xavier Guidetti and Alisa Rupenyan are also with Inspire AG, Zürich, Switzerland

³ Lutz Fassl and Majid Nabavi are with the Equipment Digitalization Team, Oerlikon Metco, Switzerland {lutz.fassl, majid.nabavi}@oerlikon.com

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

the experimentally determined quality parameters [5], [6]. Alternatively, data-driven modeling using neural networks [7], [8] or support vector machines [9] has been proposed to relate process inputs to quality parameters. All these approaches require a large number of samples, to either achieve good predictive capabilities, or to reliably cover all possible process variations.

Previous work on data-driven optimization of manufacturing processes [10], [11] has shown that Bayesian optimization (BO) is an information efficient and effective technique to automate the configuration of industrial processes with a limited number of experiments. BO has been applied in parallel or nested formulations for the tuning of algorithms [12], the optimization of black-box functions [13]–[15], or the tuning of cascaded controllers [16]. Safety-aware BO methods have been used to ensure that safety constraints are respected, providing probabilistic guarantees that all candidate samples remain in the constraint set [17]. They have been demonstrated for robotic applications [18] and for adaptive control in position tracking [19]. Iterative model-based learning and control methods have recently been applied to specific processes in the field of additive manufacturing [20].

In this paper, we propose a data-driven approach for the optimization of process input parameters, given desired output properties of the manufactured components. The main contributions of our work are: 1) a sample-efficient parameters selection procedure, based on a novel BO acquisition function whose aggressiveness can be tuned, 2) a detailed analysis of the novel acquisition function performance on benchmarking problems from the literature, and 3) a parallelized status-aware optimization procedure that incorporates process information in the BO procedure, making it fully applicable to any experimental scenario. We demonstrate the proposed method on atmospheric plasma spraying (APS) and fused deposition modeling (FDM, also known as 3D printing), which perfectly exemplify the challenges of advanced manufacturing – namely, hard-to-model multiple-input-multiple-output relationships, expensive deposition trials and time-consuming quality characterization of manufactured pieces.

In Section II, we detail the challenge that motivated the research and the techniques upon which we base our work. Section III presents our contributions on BO. In Section IV we conduct a thorough analysis and comparison of the proposed optimization method. Section V extends the BO method to complex manufacturing applications. Lastly, Sections VI and

VII detail the experimental work conducted on APS and FDM.

II. BACKGROUND

A. Process optimization in advanced manufacturing

Multiple advanced manufacturing processes, especially those relying on layer-wise deposition, are challenging for precise modeling and optimization, due to their inherent complexity. Data-driven optimization is impeded by limited data availability, which is common in this class of problems. The process outputs, i.e. the desired part properties, are unknown and can only be evaluated point-wise. The properties analysis is a slow and expensive procedure, requiring destructive approaches. The configuration is often accelerated by performing experiments in batches. Between experimental sessions, the manufacturing equipment often undergoes changes (wearing, maintenance, etc.) that need to be tracked to reduce output variation. The production cost is represented by a deterministic function of the input parameters (e.g. used resources, consumed energy, induced equipment wear, etc.). In this manufacturing context, there exists a peculiar relationship existing between cost reduction and constraints fulfillment. The main focus of the optimization lies in finding *feasible* samples, while cost reduction is a secondary goal to be achieved once feasibility has been reached. Furthermore, as these processes are often used to manufacture very small lots, producing as many feasible samples as possible during configuration itself is important [21]. While suboptimal in terms of cost, these samples are usable, reducing the amount of runs conducted and the total configuration cost.

B. Gaussian Processes

Given $\mathbf{x} \in \mathbb{R}^n$ where n is the number of inputs, we model each output function $c(\mathbf{x})$ using Gaussian process regression. A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. It provides a distribution over functions $c(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ that is fully defined by its mean function $\mu(\mathbf{x})$ and its covariance, given by the kernel function $k(\mathbf{x}, \mathbf{x}')$. We denote the i -th measurement corresponding to an input vector \mathbf{x}_i by $y_i = c(\mathbf{x}_i) + \varepsilon_i$, where ε_i is the measurement noise with distribution $\mathcal{N}(0, \sigma_n^2)$. Given a set of p input vectors paired with the corresponding noise corrupted measurements $\mathcal{T} = \{(\mathbf{x}_i, y_i)\}_{i=1}^p$, we can calculate the posterior distribution of $c(\cdot)$ at any query point $\bar{\mathbf{x}}$. Denoting the set of inputs $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^p$ and the set of corresponding measurements $\mathbf{y} = \{y_i\}_{i=1}^p$, we obtain $\tilde{c}(\bar{\mathbf{x}}) \sim \mathcal{N}(\mu_c(\bar{\mathbf{x}}), \sigma_c^2(\bar{\mathbf{x}}))$, where the corresponding posterior mean and variance are given as

$$\begin{aligned}\mu_c(\bar{\mathbf{x}}) &= \mu(\bar{\mathbf{x}}) + k(\bar{\mathbf{x}}, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbb{I}]^{-1}(\mathbf{y} - \mu(\mathbf{X})), \\ \sigma_c^2(\bar{\mathbf{x}}) &= k(\bar{\mathbf{x}}, \bar{\mathbf{x}}) - k(\bar{\mathbf{x}}, \mathbf{X})[k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbb{I}]^{-1}k(\mathbf{X}, \bar{\mathbf{x}}).\end{aligned}$$

After model training, the posterior mean $\mu_c(\bar{\mathbf{x}})$ and variance $\sigma_c^2(\bar{\mathbf{x}})$ can be used respectively as the model prediction and corresponding uncertainty at point $\bar{\mathbf{x}}$. To do so, a confidence interval of 95% is typically selected. A complete overview of GPs and their practical use can be found in [22].

C. Constrained Bayesian Optimization

In its simplest form, BO is a sequential strategy for the optimization of expensive-to-evaluate functions, often subject to safety or performance constraints. BO is commonly used with GP models, which use the available evaluations to produce a probabilistic distribution of the functions and can be updated when new samples are added to the known experiment set. To find the optimal inputs \mathbf{x}^* of a general constrained problem

$$\begin{aligned}\min_{\mathbf{x} \in \mathcal{X}} \quad & f(\mathbf{x}) \\ \text{s.t.} \quad & c(\mathbf{x}) \leq \lambda,\end{aligned}\tag{1}$$

where λ is a constant and \mathcal{X} a known bounded domain, the method starts by placing priors over the unknown objective and constraint functions $f(\mathbf{x})$ and $c(\mathbf{x})$ and then updates them with the collected data to form a posterior distribution of the functions. The posterior distribution is then used to select the next candidate for evaluation \mathbf{x}_{m+1} , according to

$$\mathbf{x}_{m+1} = \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha_m(\mathbf{x}),\tag{2}$$

where $\alpha_m(\mathbf{x})$ is the acquisition function built based on the m previously evaluated inputs. Well-designed acquisition functions trade off exploration and exploitation by combining the information content at the inputs and the corresponding predicted performance. They are a central ingredient of BO – and of our proposed approach – and can be tailored to specific (classes of) optimization problems. Numerous acquisition functions have been proposed in more or less recent works [23]–[25]. In constrained optimization, the acquisition function considers both the expected objective improvement and the expected feasibility of inputs, to select cost-reducing candidates that fulfill the constraints with high probability [26].

III. METHOD

We consider the class of optimization problems having a deterministic objective and one or more black-box constraints. We write our optimization problem as

$$\begin{aligned}\min_{\mathbf{x} \in \mathcal{X}} \quad & S(\mathbf{x}) \\ \text{s.t.} \quad & c_k(\mathbf{x}) \leq \lambda_k, \quad k = 1, \dots, K,\end{aligned}\tag{3}$$

where $S(\mathbf{x})$ denotes the problem objective, $c_k(\mathbf{x})$ the k^{th} constraint, and K the number of constraints. The input combinations \mathbf{x} belong to a known bounded domain \mathcal{X} . We assume that $S(\mathbf{x})$ can be freely computed as a deterministic function of inputs \mathbf{x} , whereas all $c_k(\mathbf{x})$ are unknown, can have any complexity, and are sampled at each iteration. We also assume that the optimization can be initialized with a set of previously evaluated input vectors and corresponding constraints values $\mathcal{T} = \{\mathbf{x}_i, \mathbf{c}(\mathbf{x}_i)\}_{i=1}^p$ where, for each i , $\mathbf{c}(\mathbf{x}_i) = \{c_k(\mathbf{x}_i)\}_{k=1}^K$.

A. Acquisition Procedure

We propose a custom acquisition procedure tailored to the problems at hand. We first introduce two functions: *improvement* and *feasibility probability*. The first one determines the amount of improvement, in terms of cost reduction, that a

vector with a candidate combination of inputs can produce. We define it as

$$I(\mathbf{x}) = \max \{0, S(\mathbf{x}^+) - S(\mathbf{x})\}, \quad (4)$$

where \mathbf{x}^+ is the feasible combination of inputs with the lowest cost found so far. If no feasible point is known, we set $S(\mathbf{x}^+) = \max_{\mathbf{x} \in \mathcal{X}} S(\mathbf{x}) + 1$. Candidate combinations producing a cost higher than \mathbf{x}^+ return no improvement. If the cost $S(\mathbf{x})$ was unknown and not deterministic, we would need to take the expectation of (4), corresponding to the *expected improvement* acquisition function of [26]. However, here we do not need to take an expectation of (4) as, knowing $S(\mathbf{x})$, $I(\mathbf{x})$ is also deterministic.

To include the constraints of (3) we define the feasibility probability as

$$FP(\mathbf{x}) = Pr[\tilde{c}(\mathbf{x}) \leq \lambda] = \int_{-\infty}^{\lambda} p(\tilde{c}(\mathbf{x}) | \mathbf{x}, \mathcal{T}) d\tilde{c}(\mathbf{x}). \quad (5)$$

Since $\tilde{c}(\mathbf{x})$ has Gaussian marginals, $FP(\mathbf{x})$ is a Gaussian cumulative distribution function. With multiple independent constraints, the feasibility probability is

$$FP(\mathbf{x}) = \prod_{k=1}^K Pr[\tilde{c}_k(\mathbf{x}) \leq \lambda_k]. \quad (6)$$

We now define the two novel acquisition functions that our algorithm exploits:

$$\alpha_{FIP}(\mathbf{x}) = FP(\mathbf{x}) \operatorname{sgn}\{I(\mathbf{x})\} \quad (7a)$$

$$\alpha_{HFI}(\mathbf{x}) = (FP(\mathbf{x}) - \pi)I(\mathbf{x}), \quad (7b)$$

where $\pi \in [0, 1]$ is a confidence threshold that tunes the aggressiveness of our acquisition algorithm. *Feasible Improvement Probability* (FIP) (7a) has a conservative approach: it returns the feasibility probability of the candidates that are known to produce any cost improvement. As the magnitude of the improvement is eliminated by the sign function, maximizing $\alpha_{FIP}(\mathbf{x})$ corresponds to looking for the points that have the highest chances of respecting the constraints. *High FIP Improvement* (HFI) (7b) is more aggressive: the magnitude of the cost improvement modulates the candidate selection, producing a trade-off between feasibility probability and reward. Maximizing $\alpha_{HFI}(\mathbf{x})$ returns candidates that markedly reduce the cost while maintaining a minimum feasibility probability of π .

Algorithm 1: Candidate Selection

input: $FP(\mathbf{x})$ and $I(\mathbf{x})$ of all candidates \mathbf{x} in the candidates set \mathcal{U} , previously evaluated inputs set \mathcal{T} , constraints, threshold probability π

- 1 Compute $\alpha_{FIP}(\mathbf{x})$ and $\alpha_{HFI}(\mathbf{x})$ for all candidates;
- 2 Group the elements of \mathcal{T} respecting the constraints in $\mathcal{T}_f \subset \mathcal{T}$;
- 3 **if** $\mathcal{T}_f = \emptyset$ **then**
- 4 | $\alpha(\mathbf{x}) \leftarrow \alpha_{FIP}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{U}$;
- 5 **else**
- 6 | **if** any candidate verifies $\alpha_{FIP}(\mathbf{x}) > \pi$ **then**
- 7 | | $\alpha(\mathbf{x}) \leftarrow \alpha_{HFI}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{U}$;
- 8 | **else**
- 9 | | $\alpha(\mathbf{x}) \leftarrow \alpha_{FIP}(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{U}$;
- 10 **end**
- 11 **end**
- 12 **return** selected candidate $\mathbf{x}^* = \arg \max_{\mathbf{x} \in \mathcal{U}} \alpha(\mathbf{x})$

Algorithm 1 presents the complete candidate selection procedure for a candidate set \mathcal{U} . As the results of Sections IV-C and VI-A will show, we observed that state-of-the-art methods such as [26] are too aggressive, while an approach focusing on FIP only led to excessively conservative exploration. Given that maintaining feasible solutions has higher priority than optimizing $S(\mathbf{x})$, we introduce a novel switching acquisition procedure that maintains a trade-off between the two approaches. As long as no feasible point is found, we exclusively focus on maximizing the chances of finding one. We therefore perform the optimization according to (7a) (line 4). Once we have found a feasible experiment, we take a mixed approach. We want to ensure that the aggressive exploration of (7b) is conducted only with a sufficient safety margin, given by the confidence threshold π (line 6). When the probability of finding cost reducing points is too low, we take the conservative approach of (7a) (line 9). At line 12, we select the candidate belonging to the candidates set \mathcal{U} that has been assigned the largest $\alpha(\mathbf{x})$. This procedure increases the amount of feasible experiments found during the optimization and favors a safer exploration of the constraints space over uncertain large improvements. It is important to remark that, given the deterministic nature of our objective function, both (7a) and (7b) only consider candidates that will certainly produce a cost reduction.

IV. ACQUISITION PROCEDURE ANALYSIS

In this section, we evaluate the performance of the acquisition function proposed in Section III-A on benchmark problems, using a Monte Carlo approach. Specifically, we compare the performance of the acquisition functions proposed in Alg. 1 with that of the expected constrained improvement (EIC) [26] acquisition function, which has been successfully applied to similar industrial application and represents the state of the art in the domain of manufacturing processes configuration via Bayesian optimization. We selected three 2D problems that have been repeatedly used as benchmarks in the literature:

$$\begin{aligned} \min_{\mathbf{x} \in [0,6]^2} \quad & f(\mathbf{x}) = \cos(2x_1) \cos(x_2) + \sin(x_1) \\ \text{s.t.} \quad & c(\mathbf{x}) = \cos(x_1) \cos(x_2) - \sin(x_1) \sin(x_2) \leq -0.5; \end{aligned} \quad (P1)$$

$$\begin{aligned} \min_{\mathbf{x} \in [0,6]^2} \quad & f(\mathbf{x}) = \sin(x_1) + x_2 \\ \text{s.t.} \quad & c(\mathbf{x}) = \sin(x_1) \sin(x_2) \leq -0.95; \end{aligned} \quad (P2)$$

$$\begin{aligned} \min_{\mathbf{x} \in [0,1]^2} \quad & f(\mathbf{x}) = x_1 + x_2 \\ \text{s.t.} \quad & c_1(\mathbf{x}) = \frac{3}{2} - x_1 - 2x_2 - \frac{1}{2} \sin(2\pi(x_1^2 - 2x_2)) \leq 0 \\ & c_2(\mathbf{x}) = x_1^2 + x_2^2 - \frac{3}{2} \leq 0. \end{aligned} \quad (P3)$$

Problems P1 and P2 originally appeared in [26], while Problem P3 comes from [27]. All of them have been reused in later works, such as [28]. We consider the objectives $f(\mathbf{x})$ to be known and the constraints $c(\mathbf{x})$ to be unknown, and learned by sampling during the BO, or in an offline training phase. Both problems P1 and P2 have complicated objective and constraint functions with two disjoint feasible regions. The

feasible domain of P2 has a smaller total surface than the one of P1, making it harder to find feasible samples in P2. The objective function in P3 is simple, but its two constraint functions form a complicated feasible region, which makes P3 the most interesting problem in the context of this work. We conducted the study both for noiseless and noisy scenarios.

A. Noiseless Scenario

In the noiseless case, where the constraints evaluations are not corrupted, we set $\pi = 0.6$, corresponding to a moderately aggressive search for the optimal parameters. We grid each problem's input space to generate 20000 candidates. The optimization is stopped when the best candidate is evaluated, or when 100 iterations have been conducted, and the corresponding iteration number is called *required iterations*. Evaluated inputs \mathbf{x} respecting the problem constraints are called *feasible samples*. We begin each optimization procedure by randomly selecting two initialization samples. These are provided to both Alg. 1 and EI_C . Each optimization procedure is repeated 100 times with random initialization samples. Figure 1 shows the benchmark problems together with one representative optimization trace. The mixed strategy upon which Alg. 1 is based results in a sequence of evaluations that generally avoid large constraint violations, and produces a larger fraction of *feasible samples* than EI_C . It can be seen in all cases that EI_C selects samples that would produce very large cost reductions, but turn out to be infeasible, effectively wasting evaluations. Because of this, Alg. 1 often requires a comparable or smaller number of evaluations to find the optimizer. In the situations where $\alpha_{FIP}(\mathbf{x}) > \pi$, Alg. 1 has been designed to behave very similarly to EI_C , which explains why some sections of the optimization traces look identical for both approaches.

Fig. 2 shows that both algorithms converge with similar rates. As seen in Fig. 1, EI_C aggressiveness often makes feasible samples harder to find. EI_C shows a marginally better convergence speed only for the first problem, as can be seen on Fig. 2a. This is due to the relatively simple constraint of Problem P1 that makes the cautiousness of Alg. 1 unnecessary.

The comparison of achieved averaged feasible samples and required iterations across all 100 repetitions is summarized in Table I. In all three problems, Alg. 1 outperforms EI_C by producing more *feasible samples*, without an adverse effect on the number of *required iterations*. Increasingly complex constraints tend to favor Alg. 1, that outperforms EI_C both in terms of *required iterations* and in terms of *feasible samples* for Problems P2 and P3.

TABLE I: Noiseless Scenario Comparison

	P1		P2		P3	
	Alg. 1	EI_C	Alg. 1	EI_C	Alg. 1	EI_C
Req. it.	14.6	13.0	22.2	22.9	26.1	31.8
Feas. sam.	66%	49%	35%	28%	54%	27%

B. Noisy Scenario

In this comparison, we corrupt each constraint evaluation by adding normally distributed noise $\mathcal{N}(0, \tau^2)$, using $\tau = 0.2$,

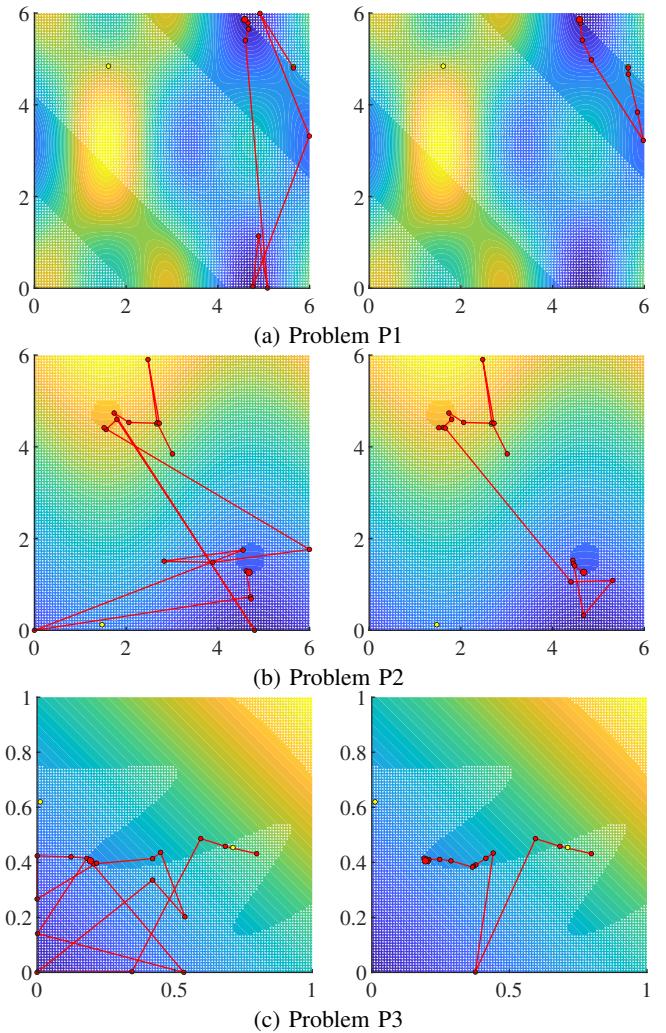


Fig. 1: Comparison of representative optimization traces for EI_C (left) and Alg. 1 (right). The problems objective is represented with a colormap, with lower values plotted in blue. White dots are used to show unfeasible regions. Initialization samples are marked in yellow, while the trace is shown in red, with a larger final marker.

following [28]. As in the previous study, we keep $\pi = 0.6$. We utilize the same list of 20000 candidates that was generated for the noiseless case. We stop the optimization when a feasible candidate in a tolerance radius (0.15 for P1 and P2, and 0.0125 for Problem P3) from the best candidate is evaluated, or after conducting 100 iterations. We consider as feasible samples the evaluated inputs \mathbf{x} whose corrupted evaluation respects the constraints. As in the noiseless case, we initialize Alg. 1 and EI_C with two identical random samples. Each algorithm is run five times with the same initialization samples, but different noise realizations. We repeat this with 20 different initializations for a total of 100 optimizations. The aggregated results, averaged across all repetitions, are shown in Table II. As expected, the performance of both algorithms deteriorates compared to the noiseless case for all problems. Despite this, Alg. 1 still outperforms EI_C in terms of feasible samples for all Problems, and the number of required iterations is again comparable. Only in the case of Problem P2, Alg. 1 requires on average 11 more iterations than EI_C to meet the stopping

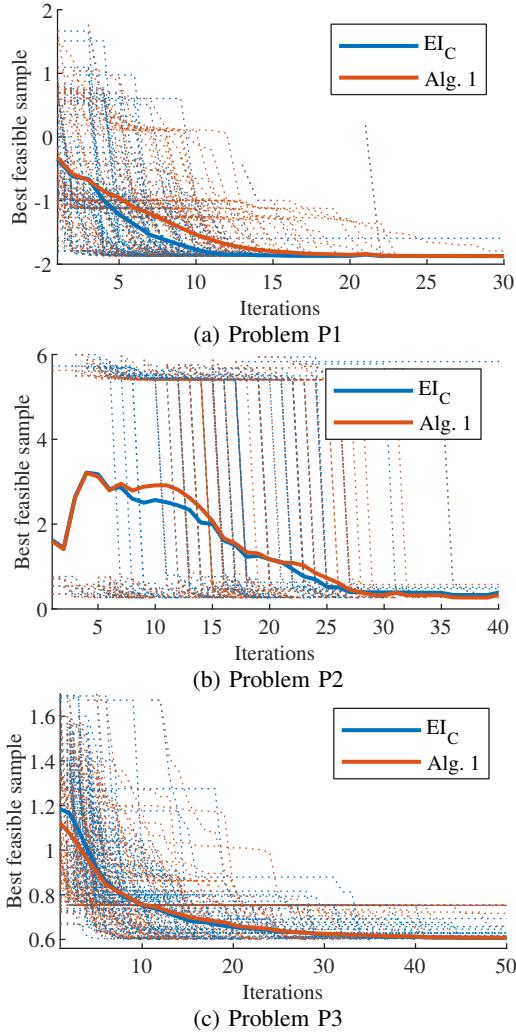


Fig. 2: Convergence speed comparison over 100 repetitions. Dashed lines represent best feasible sample found at a given iteration for each repetition. These results are averaged to produce the solid lines.

TABLE II: Noisy Scenario Comparison

	P1		P2		P3	
	Alg. 1	EI_C	Alg. 1	EI_C	Alg. 1	EI_C
Req. it.	26.0	28.5	60.9	49.9	32.5	36.0
Feas. sam.	59%	35%	25%	15%	48%	19%

condition. This is explained by the noise magnitude, which is large when compared to the feasible region size. Furthermore, when Alg. 1 finds the top-left feasible region, its cautious nature makes it less likely to explore the bottom-right feasible region where the optimizer is. The difference between the two algorithms is particularly visible in Problem P3, that has the most complex constraints. Alg. 1 finds 2.5 times more feasible samples than EI_C , and the efficient learning of the feasible region leads to faster convergence.

C. Confidence Threshold Study

We now analyze the effect of the confidence threshold π on Alg. 1 performance, focusing on Problem P3. Its simple objective and complex constraints exemplify the class of

problems for which Alg. 1 was designed. We follow the same procedure as in Sec. IV-A. We repeat the numerical study with 11 different values of π to produce the data shown in Fig. 3.

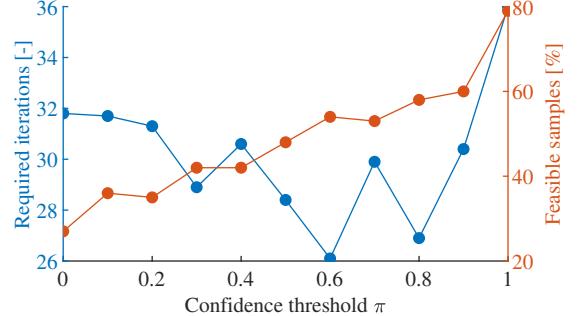


Fig. 3: Required iterations and feasible samples for different values of π , when solving Problem P3 using Alg. 1.

The confidence threshold π modulates the behavior of our algorithm. When $\pi = 0$, Alg. 1 only uses (7b), which in this case is identical to using EI_C on a known objective function. As expected, the results for $\pi = 0$ correspond to the ones in Table I for EI_C on P3. At the other extreme, when $\pi = 1$, Alg. 1 only uses (7a), which produces a very conservative and slow approach that never attempts bold cost reductions. We further observe that increasing the value of π increases the fraction of *feasible samples* in the optimization trace. Interestingly, the number of *required iterations* does not grow with π . Instead, given the nature of the problem, a cautious learning of the constraints can accelerate the optimization. All values of $\pi \in [0.1, 0.9]$ make Alg. 1 outperform EI_C on both metrics, with a minimum number of required iterations found at $\pi = 0.6$. While the confidence threshold can be fine-tuned according to the optimization goals, the algorithm is relatively robust to the changes of π and can be safely set to $\pi \in [0.3, 0.8]$ to produce satisfactory results. Tuning-dependent algorithms often have a very narrow hyper-parameter window, out of which performance deteriorates quickly. This is not the case with Alg. 1: the robust performance and its gradual change make the fine-tuning task simpler for practitioners. The complexity and computation time of Alg. 1 are independent of π and entirely similar to those of comparable methods such as EI_C , with each iteration being completed in less than 500 ms.

V. IMPLEMENTATION ON MANUFACTURING PROCESSES

In this section we describe the methods we developed to be able to efficiently utilize Alg. 1 for the parameters tuning of manufacturing processes.

A. Parallel Optimization

Parallelizing the BO evaluations accelerates the data collection procedures. Several methods to parallelize BO have been proposed [12]–[15]. We utilize a simple fixed batch size technique that is largely based on sequential selection of query points. Unlike standard BO where the candidate \mathbf{x}^* evaluation is conducted immediately, a prediction \hat{y}^* of the output produced by the candidate is made. The data

set of known evaluations is virtually expanded using the prediction, the GP is retrained, and another candidate is selected. This process of virtual evaluations is repeated until a batch $\mathcal{B} = \{\mathbf{x}_i^* | \mathbf{x}_i^* \in \mathcal{X}\}_{i=1}^n$ having the desired size n has been generated. The points belonging to the batch are evaluated simultaneously and the predictions $\{\hat{y}_i^*\}_{i=1}^n$ are replaced with the experimental results $\{y_i^*\}_{i=1}^n$ to retrain the GP. When using GPs to model the unknown functions, data set virtual augmentation can be carried on by drawing samples \hat{y}^* from the posterior distribution evaluated at the candidate site $\tilde{c}(\mathbf{x}^*) \sim \mathcal{N}(\mu_c(\mathbf{x}^*), \sigma_c^2(\mathbf{x}^*))$.

B. Status-Aware Optimization

Industrial process modeling can include uncontrollable but measurable parameters, reflecting the status of the manufacturing equipment. Output data collected in different sessions might contain offsets due to drifts or undocumented changes to the equipment between sessions. Including all measurable process parameters in the modeling and optimization makes it possible to prevent deviations during the configuration.

Let us consider a process whose model takes inputs $\mathbf{x} = (\mathbf{x}_c, \mathbf{x}_m)$ to predict the outputs $\mathbf{c}(\mathbf{x})$. The inputs in \mathbf{x}_c can be freely tuned during the BO configuration process, while \mathbf{x}_m represents measurements obtained at the end of each evaluation of \mathbf{x} . The measurements \mathbf{x}_m may depend on the controllable parameters \mathbf{x}_c as well as on the equipment status. We assume that this status remains constant during a single experimental session, but can change between sessions.

We detail a calibration procedure for the case where \mathbf{x}_m contains a single status dependent measurement V . Using the entries of the initialization data set \mathcal{T} , we train a model accepting the controllable inputs \mathbf{x}_c and predicting the measurement $\hat{V} = M_V(\mathbf{x}_c)$. At the beginning of each new experimental session, we conduct an experiment with any settings $\mathbf{x}_c^b \in \mathcal{T}$, measure the corresponding V^b and compute the offset $\delta^b = V^b - \hat{V}^b$, where $\hat{V}^b = M_V(\mathbf{x}_c^b)$ is the measurement that an unchanged equipment – with respect to \mathcal{T} – would produce. To generate the list of candidates for the BO session, we first grid the space of controllable inputs \mathcal{X}_c to produce a set of candidates \mathbf{x}_c . We then predict the status dependent measurement \hat{V}_δ corresponding to each candidate \mathbf{x}_c in the set according to $\hat{V}_\delta = M_T(\mathbf{x}_c) + \delta^b$. Each prediction is used to expand the corresponding \mathbf{x}_c , producing a candidates set \mathcal{U} of vectors $\mathbf{x} = (\mathbf{x}_c, \mathbf{x}_m)$.

Parallel BO is carried on the candidate set \mathcal{U} as detailed in Alg. 2 and Fig. 4. Following the procedure described in section V-A, we select candidates individually and expand our virtual database using the GPs posterior mean (line 13). The termination condition at line 16 interrupts the procedure when most of the candidates have a low FIP, indicating that further improvement is unlikely.

VI. ATMOSPHERIC PLASMA SPRAYING CONFIGURATION

We demonstrate an application of the proposed optimization algorithm and implementation procedure on APS. APS is a thermal spraying process where micrometer-sized powder particles are injected into a viscous enthalpy plasma jet that

Algorithm 2: Optimization Workflow

```

input: Initialization data set  $\mathcal{T}$ , candidate set  $\mathcal{U}$ , constraints  $\lambda$ , batch size  $n$ , threshold probability  $\pi$ , termination threshold  $\epsilon$ 
1 Create an empty candidate batch  $\mathcal{B} \leftarrow \emptyset$ ;
2 repeat
3   if  $\mathcal{B} \neq \emptyset$  then
4     Evaluate experimentally the candidates in  $\mathcal{B}$  and expand  $\mathcal{T}$  with  $\{\{\mathbf{x}_i^*, \mathbf{y}_i^*\}\}_{i=1}^n$ , where  $\mathbf{y}_i^*$  collects the evaluations of each  $\mathbf{x}_i^*$ ;
5     Empty  $\mathcal{B} \leftarrow \emptyset$ ;
6   end
7   Use (4) to calculate  $I(\mathbf{x})$  of all  $\mathbf{x} \in \mathcal{U}$ , make a virtual copy  $\mathcal{T}_v \leftarrow \mathcal{T}$ ;
8   for  $i = 1$  to  $n$  do
9     Using the the data in  $\mathcal{T}_v$ , model the constraints  $\mathbf{c}(\cdot)$ ;
10    Use (6) to calculate  $FP(\mathbf{x})$  of all  $\mathbf{x} \in \mathcal{U}$ ;
11    Select the candidate  $\mathbf{x}_i^*$  using Alg. 1;
12    Remove  $\mathbf{x}_i^*$  from  $\mathcal{U}$  and add it to  $\mathcal{B}$ ;
13    Expand  $\mathcal{T}_v$  with  $(\mathbf{x}_i^*, \mu_c(\mathbf{x}_i^*))$ , where  $\mu_c(\mathbf{x}_i^*)$  collects the predictive means of the constraints  $\mathbf{c}(\cdot)$ ;
14     $i \leftarrow i + 1$ ;
15  end
16 until  $\alpha_{FIP}(\mathbf{x}_i^*) < \epsilon$  for at least half of the candidates  $\mathbf{x}_i^*$  selected for a batch  $\mathcal{B}$  ;
17 return feasible stress index minimizer  $\mathbf{x}^+ = \arg \min_{\mathbf{x} \in \mathcal{T}_f} S(\mathbf{x})$ , where  $\mathcal{T}_f \subset \mathcal{T}$  contains the feasible elements of  $\mathcal{T}$ 

```

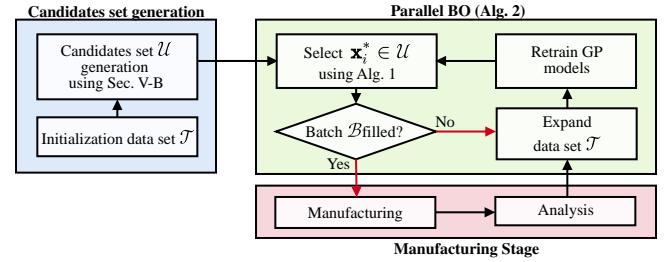


Fig. 4: Flowchart of the proposed configuration method

heats them and propels them. The particles form a protective coating that improves the mechanical properties of a substrate upon bonding with its surface. The coating properties (e.g. application rate, thickness, porosity, microhardness) depend on multiple process input parameters [29].

Details about the process data-driven modeling are available in [30]. Our goal is to select values for six controllable process inputs to regulate the coating microhardness and porosity and maximizing the equipment lifetime. In the absence of real-time measurements reflecting the system wear, the objective of maximizing the lifetime is encoded through the minimization of the stress index, an empirical relation reflecting the working conditions of the gun components during the coating process. The stress index value can be calculated explicitly from a subset the input parameters. While spraying a coating, we measure the gun voltage as it contains valuable information about the equipment status.

A. Simulated Process Optimization

To conduct simulated studies, we use the neural network model structure and data set from [30]. The neural network simulates the behavior of the APS machine and acts as an oracle during the optimization process, returning the microhardness and porosity of virtual coated samples. Following the procedure detailed in section V-B, we treat the gun voltage as a

status dependent measurement. We simulate a scenario where, during a first gun ignition, we measure a voltage offset of 2 V, indicating a change in the equipment status. The goal of the optimization is to find combinations of inputs producing coatings with microhardness ranging between 635 HV and 675 HV and porosity between 6% and 8.2% while minimizing the gun stress index. We select a batch size $n = 5$, a threshold probability $\pi = 0.4$ and a termination threshold $\epsilon = 0.05$. We initialize the optimization with $N_{\text{init}} = 86$ experiments, none of which respects the constraints.

TABLE III: Stopping Batch and Fraction of Feasible Samples for Alg. 1 and EI_C

	Alg. 1($\pi = 0.4$)	EI_C
Cost	102	102
Stopping batch	4	7
Nr. evaluations	20	35
Feasible samples	45%	20%

Table III compares the performance of the proposed acquisition procedure with that of EI_C under identical conditions. Both algorithms reach the same minimum cost when meeting the termination condition, however Alg. 1 does so using 43% less evaluations and producing 2.25 times more feasible samples. The results reaffirm those in section IV, confirming that Alg. 1 outperforms EI_C for APS configuration. A very detailed analysis of Alg. 1 behavior on this simulated problem is available in [30].

B. Process Optimization Experiments

We tested our algorithm on the APS machine to evaluate its real-world performance. As in the simulated case, the goal was to find combinations of inputs that produce coatings with microhardness ranging between 635 HV and 675 HV and porosity between 6% and 8.2%. The models were initialized using all the available data set of 86 experiments. We could coat four samples in Batch 1 (with a voltage offset of 2 V) and five samples in Batch 2 (with a voltage offset of -0.8 V). The experimental results are shown in Fig. 5. As in the case of the simulated procedure, no initialization experiment was feasible, which made the optimization algorithm begin with a cautious approach. The samples of Batch 1 respect the imposed constraints both for microhardness and porosity. The lowest found stress index corresponds to 120.3 (indicated by a star in the figure). This value is used as an upper bound in the second batch search, where the algorithm now acts aggressively to further reduce the stress index. The results are very similar to the simulated ones, with the samples of batch 2 being unfeasible because of low microhardness (cf. [30], Fig. 5). While further experimental work was impossible as APS is very expensive and sample analysis extremely time-consuming, we observe a qualitative agreement between experimental and simulated results.

VII. FUSED DEPOSITION MODELING CONFIGURATION

We further used Alg. 1 to search for optimal print parameters in FDM. We printed using a liquid-crystal polymer (LCP)

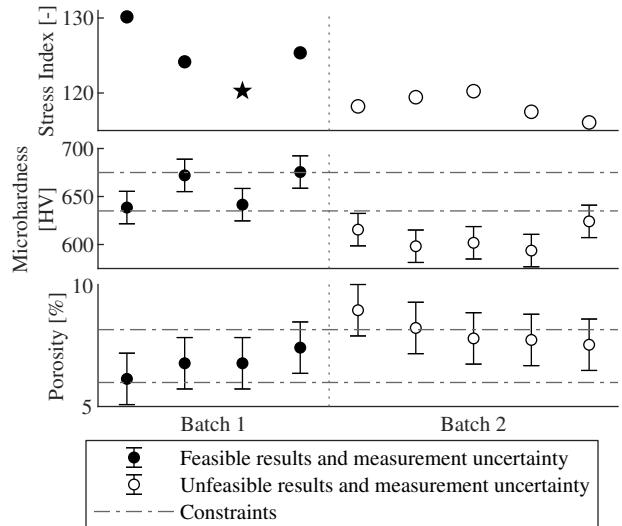


Fig. 5: Experiments for the optimization of an APS process showing the coating properties of the samples belonging to two batches. In the top panel, the star denotes the best feasible point found in the experiments.

filament, for which good print parameters are particularly hard to find [31]. The goal of the configuration consisted in finding a combination of extrusion rate and printer speed that would minimize the print time while maintaining a satisfactory print quality. The print quality constraint was enforced by setting an upper limit of $10\text{ }\mu\text{m}$ to the surface roughness R_a of the printed samples. We set the batch size to one and initialized Alg. 1 with seven available experiments.

Figure 6 shows the results of the configuration procedure. To better demonstrate the behavior of Alg. 1, we began the process with $\pi = 0.4$, making the approach relatively cautious. The algorithm steadily reduced the print time and a large fraction of the samples printed in this phase respect the roughness constraint. Then, after experiment 16, we lowered the confidence threshold to $\pi = 0.1$, making the algorithm more aggressive. Almost immediately, the print time was significantly reduced. As expected in this second phase, however, the fraction of samples respecting the constraint diminished significantly, clearly showing the trade-off induced by π .

VIII. CONCLUSION

We presented a method for the automated configuration of advanced manufacturing processes, based on GP models and parallelized constrained BO. Our method incorporates process information in the optimization procedure, to efficiently direct the search for input parameters that produce the desired output property specifications and minimize the process cost. The algorithm is based on a novel acquisition method tailored to the class of problems having known objectives and black-box constraints, to which advanced manufacturing processes belong. The acquisition method performance was compared to the state-of-the-art on benchmark problems. We also demonstrated our method on APS and FDM. The results show that the proposed method quickly finds feasible input combinations and then exploits the collected information and the problem structure to optimize the processes.

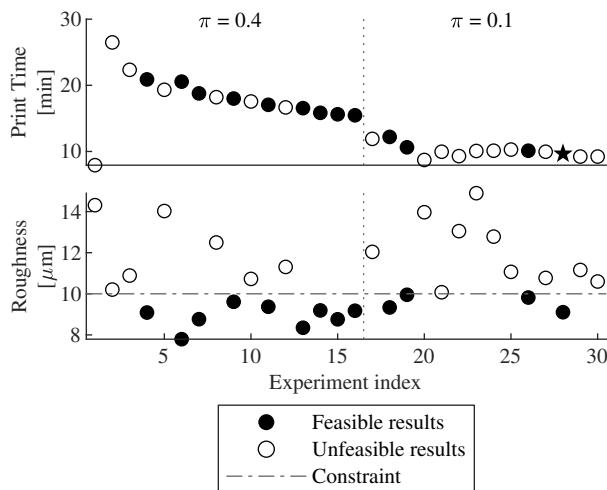


Fig. 6: Experiments for the optimization of FDM using LCP filament, conducted with two different values of π . In the top panel, the star denotes the best feasible point found in the experiments.

REFERENCES

- [1] A. Dey and N. Yodo, "A systematic survey of FDM process parameter optimization and their influence on part characteristics," *Journal of Manufacturing and Materials Processing*, vol. 3, no. 3, 2019.
- [2] C. Aguilar-Ibanez, J. Moreno-Valenzuela, O. García-Alarcón, M. Martínez-López, J. A. Acosta, and M. S. Suárez-Castañon, "Pi-type controllers and σ - δ modulation for saturated dc-dc buck power converters," *IEEE Access*, vol. 9, pp. 20346–20357, 2021.
- [3] J. D. J. Rubio, E. Orozco, D. A. Cordova, M. A. Islas, J. Pacheco, G. J. Gutierrez, A. Zácarias, L. A. Soriano, J. A. Meda-Campaña, and D. Mujica-Vargas, "Modified linear technique for the controllability and observability of robotic arms," *IEEE Access*, vol. 10, pp. 3366–3377, 2022.
- [4] F. Gao, X. Huang, R. Liu, and Q. Yang, "Optimization of plasma spray process using statistical methods," *Journal of Thermal Spray Technology*, vol. 21, pp. 176–186, 2012.
- [5] S. Datta, D. K. Pratihar, and P. P. Bandyopadhyay, "Modeling of plasma spray coating process using statistical regression analysis," *The International Journal of Advanced Manufacturing Technology*, vol. 65, no. 5-8, pp. 967–980, 2013.
- [6] Z. Wu, "Empirical modeling for processing parameters' effects on coating properties in plasma spraying process," *Journal of Manufacturing Processes*, vol. 19, pp. 1–13, 2015.
- [7] A. F. Kanta, G. Montavon, M. Vardelle, M. P. Planche, C. C. Berndt, and C. Coddet, "Artificial neural networks vs. fuzzy logic: Simple tools to predict and control complex processes - Application to plasma spray processes," *Journal of Thermal Spray Technology*, vol. 17, no. 3, pp. 365–376, 2008.
- [8] A. F. Kanta, G. Montavon, M. P. Planche, and C. Coddet, "Artificial neural networks implementation in plasma spray process: Prediction of power parameters and in-flight particle characteristics vs. desired coating structural attributes," *Surface and Coatings Technology*, vol. 203, no. 22, pp. 3361–3369, 2009.
- [9] D. Ding, F. He, L. Yuan, Z. Pan, L. Wang, and M. Ros, "The first step towards intelligent wire arc additive manufacturing: An automatic bead modelling system using machine learning through industrial information integration," *Journal of Industrial Information Integration*, vol. 23, p. 100218, 2021.
- [10] M. Maier, R. Zwicker, M. Akbari, A. Rupenyan, and K. Wegener, "Bayesian optimization for autonomous process set-up in turning," *CIRP Journal of Manufacturing Science and Technology*, vol. 26, pp. 81–87, 2019.
- [11] M. Maier, A. Rupenyan-Vasileva, C. Bobst, and K. Wegener, "Self-optimizing grinding machines using Gaussian process models and constrained Bayesian optimization," *The International Journal of Advanced Manufacturing Technology*, vol. 108, pp. 1–14, 05 2020.
- [12] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," *Advances in Neural Information Processing Systems*, vol. 4, pp. 2951–2959, 2012.
- [13] J. Wang, S. C. Clark, E. Liu, and P. I. Frazier, "Parallel Bayesian Global Optimization of Expensive Functions," *Operations Research*, pp. 1–30, 2020.
- [14] D. Ginsbourger, R. L. Riche, and L. Carraro, "A Multi-points Criterion for Deterministic Parallel Global Optimization based on Gaussian Processes," *HAL preprint hal00260579*, vol. 1, pp. 1–30, 2008.
- [15] J. Azimi, A. Jalali, and X. Z. Fern, "Hybrid batch Bayesian optimization," *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, vol. 2, pp. 1215–1222, 2012.
- [16] C. König, M. Khosravi, M. Maier, R. S. Smith, A. Rupenyan, and J. Lygeros, "Safety-aware cascade controller tuning using constrained Bayesian optimization optimization," *arXiv preprint arXiv:2010.15211*, 2020.
- [17] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *International Conference on Machine Learning (ICML)*, 2015.
- [18] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes," *CoRR*, vol. abs/1509.01066, 2015.
- [19] C. König, M. Turchetta, J. Lygeros, A. Rupenyan, and A. Krause, "Safe and efficient model-free adaptive control via Bayesian optimization optimization," *arXiv preprint arXiv:2101.07825*, 2021.
- [20] A. G. Dharmawan, Y. Xiong, S. Foong, and G. Song Soh, "A model-based reinforcement learning and correction framework for process control of robotic wire arc additive manufacturing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4030–4036, 2020.
- [21] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: A review," *Engineering*, vol. 3, no. 5, pp. 616 – 630, 2017.
- [22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- [23] G. Guinet, "Pareto-efficient Acquisition Functions for Cost-Aware Bayesian Optimization," 2020.
- [24] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani, "A general framework for constrained Bayesian optimization optimization using information-based search," *Journal of Machine Learning Research*, vol. 17, pp. 1–53, 2016.
- [25] E. C. Garrido-Merchán and D. Hernández-Lobato, "Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints," *Neurocomputing*, vol. 361, pp. 50–68, 2019.
- [26] J. R. Gardner, M. J. Kusner, Z. Xu, K. Q. Weinberger, and J. P. Cunningham, "Bayesian optimization with inequality constraints," *31st International Conference on Machine Learning, ICML 2014*, vol. 3, pp. 2581–2591, 2014.
- [27] R. B. Gramacy, G. A. Gray, S. Le Digabel, H. K. Lee, P. Ranjan, G. Wells, and S. M. Wild, "Modeling an augmented lagrangian for blackbox constrained optimization," *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016.
- [28] B. Letham, B. Karrer, G. Ottino, and E. Bakshy, "Constrained bayesian optimization with noisy experiments," *arXiv*, no. 2, pp. 495–519, 2017.
- [29] L. Leblanc and C. Moreau, "Study on the Long-Term Stability of Plasma Spraying," *Proceedings of the International Thermal Spray Conference*, vol. 11, no. September, pp. 1233–1240, 2000.
- [30] X. Guidetti, A. Rupenyan, L. Fassl, M. Nabavi, and J. Lygeros, "Plasma spray process parameters configuration using sample-efficient batch bayesian optimization," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 31–38, 2021.
- [31] S. Gantenbein, K. Masania, W. Woigk, J. P. W. Sesseg, T. A. Tervoort, and A. R. Studart, "Three-dimensional printing of hierarchical liquid-crystal-polymer structures," *Nature*, vol. 561, pp. 226–230, 2018.