

# Safe Real-Time Optimization using Multi-Fidelity Gaussian Processes

Panagiotis Petsagkourakis<sup>1</sup>, Benoit Chachuat<sup>2</sup> and Ehecatl Antonio del Rio-Chanona<sup>2</sup>

**Abstract**—This paper proposes a new class of real-time optimization schemes to overcome system-model mismatch of uncertain processes. This work’s novelty lies on integrating derivative-free optimization schemes and multi-fidelity Gaussian processes within a Bayesian optimization framework. The proposed scheme uses two Gaussian processes for the stochastic system, one emulates the (known) process model, and another, the true system through measurements. In this way, low fidelity samples can be obtained via a model, while high fidelity samples are obtained through measurements of the system. This framework captures the system’s behavior in a non-parametric fashion, while driving exploration through acquisition functions. The benefit of using a Gaussian process to represent the system is the ability to perform uncertainty quantification in real-time and allow for chance constraints to be satisfied with high confidence. This results in a practical approach that is illustrated in numerical case studies, including a semi-batch photobioreactor optimization problem.

## I. INTRODUCTION

The advantages of real-time optimization (RTO) in industrial processes are well known [1], [2], however, currently, most implementations rely on heuristics and trial and error to reach optimality [2]. The fact that complex phenomena cannot be modelled exactly is the biggest challenge in real-time optimization, in this context we refer to the ‘plant’ as the ‘true’ system, and therefore refer to ‘plant-model mismatch’ as the error of our model with respect to the real system (the plant). Real-time optimization relies on the development of models that are utilized to conduct optimization [3], these (imperfect) models are usually updated in real-time when measurements are available. With the updated models, the optimization is repeated, and an iterative procedure between model refinement and optimization ensues. This two-step procedure in RTO schemes is called the model-adaptation strategy. Despite being the most widely used method in industry, this model-adaptation strategy does not converge to the systems’s optimal operating conditions. This has led research into different methods in RTO [4], [5], [6], [7]. One approach that has shown particular promise in the literature is that of modifier adaptation [6]. Modifier adaptation relies on adding linear functions to the optimization model’s cost and constraint functions while keeping a nominal process model. This approach has shown to be very efficient in

some scenarios, nevertheless, function gradients from process measurements are needed, which are difficult to estimate in practice. Variants of modifier adaptation to resolve this issue include recursive update schemes [8], [9], directional derivatives [10], transient process measurements [11].

The use of derivative-free approaches can lift the issues with the estimation of gradients. The use of quadratic surrogates is proposed in [12] fitted on available plant data. Similarly, data-driven models are investigated in [13] based on quadratic surrogates as modifiers for the predicted cost and constraint functions. The natural extension of such surrogates is Gaussian processes (GPs) that were first used in RTO [14]. Subsequently, GPs were employed in RTO using Bayesian optimization ideas [15]. Recently, the convergence certificates for such schemes were verified [16]. The idea of correcting the mismatch of a knowledge-driven model with a data-driven model is akin to hybrid semi-parametric modelling [17], specifically a parallel hybrid model structure. The consideration of non-parametric models, whereby the nature and number of parameter areas not determined by a priori knowledge but tailored to the data at hand, makes perfect sense to capture the structural plant-model mismatch in RTO applications. In principle, this approach is even amenable to a completely model-free RTO scheme by simply discarding the first-principles model component.

The utilization of non-parametric models such as GPs, where the number of parameters is not determined a priori but rely on the available data, is the most natural approach to capture the structural plant-model mismatch of a system. In fact, a completely model-free RTO scheme could be used by discarding the mechanistic model component; however, relying on knowing information about the system in the form of a model increases performance [15], [18], [19].

### A. Multi-fidelity modelling via Gaussian processes

A process model with low predictive capabilities combined with the real system’s samples (or a high fidelity model) has gained a lot of attention in multi-fidelity modelling. Notice that most industrially relevant models are not analytical. They are often complex black-box simulators and legacy code, e.g. systems of ordinary or partial differential equations or a set of ‘if/else if’ rules. We can evaluate the models, but their derivatives with respect to the optimization variables cannot be trivially found. For this reason, a multi-fidelity Gaussian process is proposed to account for plant-model mismatch and the low fidelity black-box models. Popular techniques to perform such a multi-fidelity approach include classical Auto-Regressive (AR) schemes [20]. The linear autoregressive information scheme introduced by Kennedy

\* This project has also received funding from the EPSRC project EP/R032807/1.

<sup>1</sup>Panagiotis Petsagkourakis is with Centre for Process Systems Engineering, University College London, London, United Kingdom p.petsagkourakis@ucl.ac.uk

<sup>2</sup>Benoit Chachuat and Ehecatl Antonio del Rio-Chanona are with Centre for Process Systems Engineering, Imperial College London, London, United Kingdom a.del-rio-chanona@imperial.ac.uk, b.chachuat@imperial.ac.uk

and O’Hagan [21] has been widely used due to its ease in implementation. Recently, non-linear autoregressive techniques [22], and deep GPs [23] have been proposed to improve multi-fidelity predictive capabilities. Ideas of Multi-fidelity GP have also been used for Bayesian optimization [24].

In the high model mismatch context, three main frameworks have been proposed in model predictive control (MPC) [25]; Learning the dynamics of the system, learning the controller, and safe learning-based MPC. Most techniques cannot easily ensure that safety constraints are met, particularly during learning iterations. In the context of only parametric uncertainty, it is common to use Polynomial chaos expansions and reformulate the chance constraints using a deterministic surrogate [26], [27]. Under the presence of structural mismatch, it is common to use non-parametric models like GPs and incorporates the chance constraint satisfaction in a similar manner [28]. Similarly, safe Bayesian optimization techniques have been proposed in [29], where the GP’s variance is employed to help the safe exploration. Such approaches are sensible. However, there is no free lunch, in safe exploration; the GP’s confidence intervals may not include the true system, especially under the presence of limited data, resulting in the violation of constraints.

In this paper, we propose the use of multi-fidelity GPs to combine available models (low-fidelity) and noisy measurements from the true plant (high-fidelity). Additionally, chance constraints are reformulated using the Chebyshev-Cantelli theorem [30] to guarantee constraint satisfaction with high confidence. A trust-region is implemented to restrict the design space and avoid aggressive extrapolations on the predictions and the risky implementation of such an approach. The trust-regions have an added benefit in terms of online computational costs during the multi-fidelity GP training, as there is no need for many samples from the low fidelity model to be computed due to a restricted design space.

The rest of the paper provides background on real-time optimization and GPs in Section II, and presents convenient background in Section III. In Section IV, the proposed framework is presented. Then, this algorithm is illustrated with practical case studies in Section V, before drawing final remarks in Section VI.

## II. PROBLEM STATEMENT

The optimization of a given plant subject to operational or safety constraints under a stochastic environment can be formulated as:

$$\begin{aligned} \min_{\mathbf{u} \in \mathcal{U}} G_0^p(\mathbf{u}) &:= \mathbb{E}(g_0(\mathbf{u}, \mathbf{y}^p(\mathbf{u}))) \\ \text{s.t. } G_c^p(\mathbf{u}) &:= \mathbb{P}(g_i(\mathbf{u}, \mathbf{y}^p(\mathbf{u})) \leq 0, \quad i = 1 \dots n_g) \geq 1 - \alpha \end{aligned} \quad (1)$$

where  $\mathbf{u} \in \mathbb{R}^{n_u}$  and  $\mathbf{y}^p \in \mathbb{R}^{n_y}$  are vectors of the plant input (design) and output (measured) variables, respectively. Notice that the objective is minimized in expectation and the constraints are satisfied with a pre-defined probability, as the plant is assumed to be affected by Gaussian disturbance and noise. Additionally,  $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ ,  $i = 0 \dots, n_g$ , denote

the cost and inequality constraint functions; and  $\mathcal{U} \subseteq \mathbb{R}^{n_u}$  is the control domain, e.g. lower and upper bounds on the input variables,  $\mathbf{u}^L \leq \mathbf{u} \leq \mathbf{u}^U$ . The superscript  $(\cdot)^p$  is used to indicate plant-related quantities. Additionally, here  $(1 - \alpha) \in (0, 1)$  is a user-defined parameter representing probability of constraint satisfaction with values close to 1.

The challenge in RTO is that the exact mapping  $\mathbf{y}^p(\cdot)$  is unknown, and the output  $\mathbf{y}^p(\mathbf{u})$  can only be measured for a given input variable  $\mathbf{u}$ , under the presence of noise and disturbance. It is common to have an available model (low fidelity) of the plant’s input-output behavior  $\mathbf{y}(\mathbf{u}, \cdot)$ . Nevertheless, the available model may not have a closed-form expression, as they may be complex black-box simulators or a legacy code.

In the presence of plant-model mismatch and process disturbances, the optimal solution of Problem (1) could be significantly different from the optimization that utilizes only the model.

## III. BACKGROUND

### A. Gaussian Process

In this section, Gaussian processes (GPs) are introduced, as one of the key components of the methodology that follows. The Gaussian process generalizes the multivariate Gaussian distribution to a distribution over an infinite-dimensional vector of functions, such that every finite sample of function values are jointly Gaussian distributed. Due to the Bayesian nature, GPs can consider both epistemic (limited data) and aleatoric (stochasticity of the true model) uncertainty. A GP is fully specified with the prior mean function  $\mu(\cdot)$  and the positive semi-definite kernel function  $k(\cdot, \cdot)$ . The GP regression aims to model an unknown set of functions  $\mathbf{g} : \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$  given some noisy observations  $\mathbf{y} = \mathbf{g}(\mathbf{u}) + \varepsilon$ . For  $\mathbf{g} = [g_1, \dots, g_{n_y}]$ , this could be expressed as:

$$g_i \sim \mathcal{GP}(\mu_i(\cdot), k_i(\cdot, \cdot)), \quad (2)$$

where the prior mean  $\mu_i$  function provides knowledge of the mean of a test point prior to observing data, the kernel function  $k_i$  expresses the covariance between points. Let a number of data points be available, the posterior distribution at a test point  $\mathbf{u}^*$  is then found by the conditional distribution given the available  $N$  noisy data for each output  $i$ :  $\{\mathbf{U}, \mathbf{y}_i\}$ , with  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_N] \in \mathbb{R}^{n_u \times N}$  and  $\mathbf{y}_i = [y_{i1}, \dots, y_{iN}]^T \in \mathbb{R}^N$ . Then, the posterior mean  $m_i(\mathbf{u}^*)$  and variance  $\Sigma_i(\mathbf{u}^*)$  of the test point  $\mathbf{u}^*$  are

$$\begin{aligned} m_i(\mathbf{u}^*) &= \mathbf{k}_i^* [\mathbf{K}_i + \sigma_i^2 I]^{-1} (\mathbf{y}_i - \mathbf{1} \mu_i(\mathbf{u}^*)) + \mu_i(\mathbf{u}^*) \\ \Sigma_i(\mathbf{u}^*) &= k_i(\mathbf{u}^*, \mathbf{u}^*) - \mathbf{k}_i^* [\mathbf{K}_i + \sigma_i^2 I]^{-1} \mathbf{k}_i^{*T}, \end{aligned} \quad (3)$$

where the posterior mean is  $m_i(\mathbf{u}^*) = \mathbb{E}(f_i^* | \mathbf{u}^*, \mathbf{U}, \mathbf{y}_i)$ , the posterior variance is  $\Sigma_i(\mathbf{u}^*) = \mathbb{V}(f_i^* | \mathbf{u}^*, \mathbf{U}, \mathbf{y}_i)$ ,  $\mathbf{k}_i^* = \mathbf{k}_i(\mathbf{u}^*, \mathbf{U})$  is the covariance between training and test cases vector,  $\mathbf{K}_i = \mathbf{k}_i(\mathbf{U}, \mathbf{U})$  is covariance (or Gram) matrix. Notice that  $\Sigma_i$  is the variance of the noise-free prediction, if the noise is taken into consideration then variance should be  $\Sigma_i + \sigma_i^2$ , and  $\sigma_i^2$  is many times treated as a hyperparameter for the GP. The

predicted distribution of  $\mathbf{g}(\mathbf{u}^*)$  follows a normal distribution:

$$\mathbf{g}(\mathbf{u}^*) \sim \mathcal{N}(\mathbf{m}(\mathbf{u}^*), \mathbf{\Sigma}(\mathbf{u}^*)), \quad (4)$$

with  $\mathbf{m}(\mathbf{u}^*) = [m_1, \dots, m_{n_y}]$  and  $\mathbf{\Sigma}(\mathbf{u}^*) = \text{diag}([\Sigma_1(\mathbf{u}^*), \dots, \Sigma_{n_y}(\mathbf{u}^*)])$ . Notice, that multi-output formulation of the GP could be use instead to compute the non-diagonal terms of the covariance [31]. Common choices for the kernel  $k_i(\cdot, \cdot)$  is the squared-exponential (SE) covariance function and Matérn class of covariance functions, with special cases the Matérn 3/2 and Matérn 5/2 [32] with the hyperparameters and the noise level being estimated using maximum likelihood estimation.

### B. Multi-fidelity Gaussian process

The GP regression framework can be systematically extended to formulate probabilistic models that enable different fidelity information sources (see [22], [20]), i.e. a low-fidelity model for which many data points can be generated and a high-fidelity model/system for which only a few data-points can be obtained. Generally, numerous levels of fidelity models can be included, in this work, 2 levels of fidelity are considered, i.e. the low fidelity (known model) and the high fidelity (plant measurements).

Suppose that we have data points for the low  $y^m$  and high fidelity  $y^p$  system at locations  $\mathbf{u}^m \in \mathcal{D}^m$  and  $\mathbf{u}^p \in \mathcal{D}^p$ , respectively. The collected data can be organized as pairs by increasing fidelity as  $\mathcal{D}_t = \{\mathbf{u}^t\}$ ,  $Y^t = \{y^t\}$ ,  $t = \{m, p\}$ . The prediction model can be written as an auto-regressive scheme:

$$g^p(\mathbf{u}) = \varepsilon g^m(\mathbf{u}) + \delta(\mathbf{u}) \quad (5)$$

where  $g^m$  and  $g^p$  are GPs modelling the data at low and high fidelity respectively. Additionally,  $\varepsilon$  is a scaling constant that can be estimated and  $\delta(\cdot)$  is a GP with mean  $\mu_\delta(\cdot)$  and covariance  $k_\delta(\cdot, \cdot)$  functions, i.e.  $\delta \sim \mathcal{GP}(\mu_\delta(\cdot), k_\delta(\cdot, \cdot))$ .

A numerically efficient inference and hyper-parameter learning scheme can be employed by assuming that the experimental design sets  $\mathcal{D}_m$  and  $\mathcal{D}_p$  have a nested structure, i.e.  $\mathcal{D}_p \subseteq \mathcal{D}_m$  [33]. This assumption means that the lower fidelity's training inputs should include the data of the higher fidelity level. The inference problem is now decoupled into 2 standard GP regression problems, resulting in the multi-fidelity posterior distribution with predictive mean and variance at each level for a given  $\mathbf{u}^*$  given by

$$\begin{aligned} m_{g^p}(\mathbf{u}^*) &= \mathbf{k}_\delta^* [\mathbf{K}_\delta + \sigma_\delta^2 I]^{-1} (\mathbf{y}^p - \mathbf{1}(\varepsilon m_{g^m}(\mathbf{u}^*) \\ &\quad + \mu_\delta(\mathbf{u}^*))) + \varepsilon m_{g^m}(\mathbf{u}^*) + \mu_\delta(\mathbf{u}^*) \\ \Sigma^p(\mathbf{u}^*) &= \varepsilon^2 \Sigma^m(\mathbf{u}^*) + k_\delta(\mathbf{u}^*, \mathbf{u}^*) - \mathbf{k}_\delta^* [\mathbf{K}_\delta + \sigma_\delta^2 I]^{-1} \mathbf{k}_\delta^{*T}. \end{aligned} \quad (6)$$

The training of the lower fidelity model includes the training data points of the plant (higher fidelity)  $\mathbf{u}^{p*}$ , as the result the posterior prediction of the low fidelity at  $\mathbf{u}^{p*}$  is by construction a deterministic quantity, the model training points are assumed to be noiseless. Hence, due to the nested training sets (i.e.  $\mathcal{D}_p \subseteq \mathcal{D}_m$ ), the training of  $g^p$  given the available data reduces to a straightforward maximum-likelihood estimation problem.

### C. Chance Constraints

Given the problem definition (1), we wish to satisfy the constraints with a given probability. These chance constraints can be reformulated using the multi-fidelity GPs. To satisfy the chance constraints, a distributionally robust reformulation is employed via the Chebyshev-Cantelli theorem [30], (Theorem 3.1) using the mean and variance of the constraints.

*Theorem 3.1:* ([30]) Consider a chance constraint of the form.

$$\mathbb{P}(q \leq 0) \geq 1 - \gamma, \quad \gamma \in (0, 1) \quad (7)$$

where  $q \in \mathbb{R}^{n_q}$  is some random variable. Let  $\mathcal{Q}$  be a family of distributions with mean  $\mu_q$  and variance  $\Sigma_q$ . Then for any  $\varepsilon \in (0, 1)$ , the distributionally robust probabilistic constraint

$$\inf_{q \sim \mathcal{Q}} \mathbb{P}(q \leq 0) \geq 1 - \gamma \quad (8)$$

is equivalent to the following constraint:

$$\mu_q + r \sqrt{\Sigma_q} \leq 0, \quad (9)$$

$$\text{with } r = \sqrt{\frac{1 - \gamma}{\gamma}}.$$

This theorem provides a conservative estimate for the constraints and it holds for the whole family distribution  $\mathcal{Q}$ .

*Remark 1:* The parameter  $r$  is often used as a design parameter to reduce the inherent conservatism; however, this makes any guarantees void.

This reformulation highly relies on the accuracy of the mean and variance computed via a GP, hence the initial small amount of data points may significantly affect the result. Notice that even though the result in Theorem 3.1 is conservative for all the family of distributions with mean  $\mu_q$  and  $\Sigma_q$ , the wrong estimation of mean and variance may lead to infeasible designs.

To overcome this limitation, trust-regions are employed, where the design space of the optimization variables  $\mathbf{u}$  is further restricted.

## IV. PROPOSED FRAMEWORK

### A. Trust Regions

The use of GPs has gained a lot of attention due to their ability to provide a close form expression for both the mean and variance; however, their use far from collected data (global use of the GP) may be naive. To avoid this, trust regions are utilized to update the GPs for the model-mismatch locally and not globally. This is particularly important in safety-critical scenarios, such as in real-time optimization and many engineering applications.

To motivate the reader on the use of trust regions, we introduce the assumptions that are usually utilized for GPs as global surrogates; then, we show that even for a simple example, such an approach may not be reliable for a global prediction.

It is common to achieve bounds for the unknown function, assuming that the unknown function  $g_i$  has bounded norm in reproducing kernel Hilbert space (RKHS) [34]. This assumption[35] can lead to reliable confidence intervals.

Due to the generic nature of the systems, the assumption and confidence interval from [35] may not globally hold and invalidate the use of GPs globally. To further indicate the importance of the trust-region, a motivating example is presented next.

1) *Importance of Trust Regions: A motivating example :*

Let a simple function  $f(x) = x \sin(x)$  be assumed as the ‘true’ system, with  $x \in \mathbb{R}$  and the collected data being corrupted by a Gaussian noise  $\varepsilon_f$  with standard deviation 0.01,  $y = f(x) + \varepsilon_f$ . A GP is used to approximate the data; a squared-exponential (SE) covariance function is used and its hyperparameters are fitted using maximum likelihood estimation (with multi-starts to avoid high laying local optima). The mean  $m$  and variance  $\Sigma$  of the GP for each  $x$  are illustrated in Fig 1.

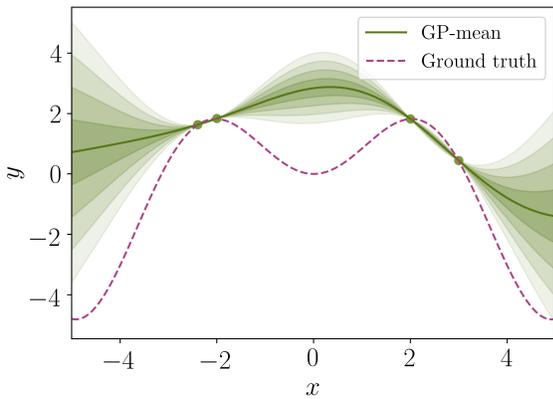


Fig. 1: The ground truth (red dashed line) is the function that the GP approximates given the available data. The GP predictions for the mean ( $\mu$ ) is the green line with variance  $\Sigma$ , and the shaded areas represents the area between  $m \pm \sqrt{\Sigma}, m \pm 2\sqrt{\Sigma}, m \pm 3\sqrt{\Sigma}, m \pm 4\sqrt{\Sigma}$  and  $m \pm 10\sqrt{\Sigma}$ .

The shaded areas show the  $m + r\sqrt{\Sigma}$  for  $r = 1, 2, 3, 4$  and 10. If we were to use Theorem 3.1, we would expect the ‘true’ model (red dashed line) to lie within these shaded areas. The Chebyshev-Cantelli theorem states that for  $r = 9.94$ ,  $x$  should lie within the shaded area with a probability of 99%. However, due to the shape of the ‘true’ model and the collected data, the predictions are poor. This example motivates the restriction of the proposed optimization problem. Although this is a well know fact for practitioners, it not generally taken into account, and it is a particular pitfall of any approach that wishes to satisfy constraints with high probability.

*Remark 2 (Trust-regions for multi-fidelity GPs):* The advantages of the trust-region are not limited to the ability of the GP to accurately represent the system and the reformulation of the chance constraints. The construction of the GP for the low fidelity model requires the acquisition of data from a simulator, which is generally expensive in large and complex processes. Without trust regions, the whole design space must be explored, which means that either i) the number of data

points is not enough to provide a good approximation, and the posterior variance is high, significantly affecting both the acquisition function and the reformulated chance constraints or ii) a large amount of data points is drawn, incurring in high computational costs for the sampling, training of the hyperparameters and prediction, which is intractable in many instances. Variational methods [36] could be used to mitigate this issue; however, the problem remains as the number of design variables increases.

This issue is accommodated with the use of the trust-region, as only data in the trust-region’s neighbourhood is needed.

B. *Proposed framework*

Herein, we introduce the concept of trust-regions from the field of derivative-free optimization together with acquisition functions from Bayesian optimization and chance constraints for safe exploration. Let  $g_i^p$  be the plant’s cost ( $i = 0$ ) and inequality constraints ( $i = 1, \dots, n_g$ ),  $g_i^m$  be the model’s cost and inequality constraints, and  $m_{g_i^p}, \Sigma_{g_i^p}, m_{g_i^m}, \Sigma_{g_i^m}$  their corresponding GP’s posterior mean and variance. Then, the modified optimization problem that is solved at each RTO iteration becomes:

$$\begin{aligned} \mathbf{d}^{k+1} \in \arg \min_{\mathbf{d}} \mathcal{A}[m_{g_0^p}, \Sigma_{g_0^p}](\mathbf{u}^k + \mathbf{d}) \\ \text{s.t. } [m_{g_i^p} + r\sqrt{\Sigma_{g_i^p}}](\mathbf{u}^k + \mathbf{d}) \leq 0, \\ \|\mathbf{d}\| \leq \Delta^k, \quad \mathbf{u}^k + \mathbf{d} \in \mathcal{U} \\ r = \sqrt{\frac{1-\alpha}{\alpha}}, \quad i = 1 \dots n_g \end{aligned} \quad (10)$$

where  $\Delta^k > 0$  is the trust-region radius for the predicted step  $\mathbf{d}^{k+1} \in \mathbb{R}^{n_u}$ ; and  $\mathcal{A}$  is an acquisition function for the cost associated with the its posterior mean and variance ( $m_{g_i^p}, \Sigma_{g_i^p}$ ), leading to, for example, the Lower Confidence Bound (LCB) or Expected Improvement (EI) function[37], [15].

The solution of Problem (10) coincides with a constrained Bayesian optimization within a trust-region. The trust-region is adapted when a new sample becomes available according to an adaptation mechanism (see next section IV-B.1 and Algorithm 1).

1) *Adaptation Mechanisms:* As stated in section IV-A, the trust region’s role is to restrict the step size of the next design, so that cost and constraints are accurate predictions of the true system. The trust region is updated according to classical update rules in trust-region algorithms [38], which relies on the ratio of the actual cost reduction versus the predicted cost reduction:

$$\rho^{k+1} := \frac{g_0^p(\mathbf{u}^k) - g_0^p(\mathbf{u}^k + \mathbf{d}^{k+1})}{m_{g_0^p}(\mathbf{u}^k) - m_{g_0^p}(\mathbf{u}^k + \mathbf{d}^{k+1})} \quad (11)$$

The trust-region radius  $\Delta^{k+1}$  is reduced whenever the accuracy ratio  $\rho^{k+1}$  is too low. Conversely,  $\Delta^{k+1}$  is increased if the solution of (10) takes a full step and the prediction of the plant cost is good around this point. In a different scenario, the trust-region radius stays unchanged. Similarly, the new design point  $\mathbf{u}^k + \mathbf{d}^{k+1}$  is accepted if the step

$\mathbf{d}^{k+1}$  produces an accuracy ratio  $\rho^{k+1}$  which is sufficiently large. Otherwise, the operating point remains unchanged. In practical situations this would trigger a back-tracking from  $\mathbf{u}^k + \mathbf{d}^{k+1}$  to  $\mathbf{u}^k$  during the RTO cycle. Any infeasibility trigger a rejection of the current step  $\mathbf{d}^{k+1}$  as well and, result in backtracking to point  $\mathbf{u}^k$ . Notice that our approach considers a chance constraint formulation which means that the constraint violation does not occur often.

---

**Algorithm 1** RTO using multifidelity GPs

---

**Input:** initial data sets  $(\mathbf{U}^0, \mathbf{g}_i^{p0})$ ,  $i = 0 \dots n_g$ ; initial operating point  $\mathbf{u}^0 \in \mathcal{U}$ ; initial and maximal trust-region radii  $0 < \Delta^0 < \Delta_{\max}$ ; trust-region parameters  $0 < \eta_1 < \eta_2 < 1$ ,  $0 < \gamma_{\text{red}} < 1 < \gamma_{\text{inc}}$ ,  $\alpha > 0$ , the number of  $N_m$  extra simulations from the black-box model.

I) Generate the nested set (i.e.  $\mathcal{D}_p \subseteq \mathcal{D}_m$ ), with  $N_m$  additional simulations in the neighborhood of the trust-region.

II) Construct the GP posterior for the low and high fidelity GP,  $\triangleright m_{g_i^p}, \Sigma_i^p, m_{g_i^m}, \Sigma_i^m$

**Repeat:** for  $k = 0, 1, \dots$

- 1) Solve modified optimization problem (Problem 10)  $\triangleright \mathbf{d}^{k+1}$
  - 2) Get process cost and constraint measurements  $\triangleright G_i^p(\mathbf{u}^k + \mathbf{d}^{k+1})$ ,  $i = 0 \dots n_g$
  - 3) Check infeasibility  
If either Problem (10) is infeasible, or if  $g_i^p(\mathbf{u}^k + \mathbf{d}^{k+1}) > 0$  for any  $i > 0$ :  
 $\Delta^{k+1} \leftarrow [\gamma_{\text{red}}, 1]\Delta^k$ ,  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k$  (reject), and go to Step 7
  - 4) Compute merit function (Equation 11)  $\triangleright \rho^{k+1}$
  - 5) Update trust region  
If  $\rho^{k+1} > \eta_2 \wedge \|\mathbf{d}^{k+1}\| = \Delta^k$ :  
 $\Delta^{k+1} \leftarrow \gamma_{\text{inc}}\Delta^k$ ,  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k + \mathbf{d}^{k+1}$  (accept)  
Else If  $\rho^{k+1} < \eta_1$ :  
 $\Delta^{k+1} \leftarrow \gamma_{\text{red}}\Delta^k$ ,  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k$  (reject)  
Else:  $\Delta^{k+1} \leftarrow \Delta^k$ ,  $\mathbf{u}^{k+1} \leftarrow \mathbf{u}^k + \mathbf{d}^{k+1}$  (accept)
  - 6) Update data set for the plant  $\triangleright \mathcal{D}_p = \{\mathcal{D}_p \cup \mathbf{u}^{k+1}\}$ ,  $\mathbf{Y}_i^p = \{\mathbf{Y}_i^p \cup g_i^{p^{k+1}}\}$ ,  $i = 0 \dots n_g$
  - 7) Generate the nested nested training set (i.e.  $\mathcal{D}_p \subseteq \mathcal{D}_m$ ), with  $N_m$  additional simulations in the neighborhood of the trust-region.
  - 8) Update low and high fidelity GP,  $\triangleright m_{g_i^p}, \Sigma_i^p, m_{g_i^m}, \Sigma_i^m$
- 

2) *Algorithm:* In **steps I** and **II**) the training set from the model and the low-fidelity GPs are constructed. The RTO iterations begin, where the first optimization is solved in **step 1**. Then the new optimized variables are applied to the true system and the corresponding measurements are sampled in **step 2**. Then, **steps 3-5** follow the adaptation procedure described in section IV-B.1. Apart from updating the trust region, both the data sets and the GPs are updated in **Steps 6, 7** and **8** respectively, this is irrespective of whether the step  $\mathbf{d}^{k+1}$  is accepted or not. Specifically, at **step 6**, the data points of the plant are generated, i.e. measurements

from the real system are collected. Then in **step 7** the nested set is constructed, where  $N_m$  random simulations are performed using the black-box model in the neighborhood of the trust-region. The black-box is also simulated at the new operational point  $\mathbf{u}^{k+1}$ . Hence, the input training set of the model  $\mathcal{D}_m$  consists of the  $N_m$  as well as the set  $\mathcal{D}_p$ . The additional  $N_m$  samples are randomly generated inside the trust-region, as it is important for the surrogate to be accurate inside the trust region. The computational burden of reconstructing the GPs at each iteration could be eased upon updating the covariance matrix at certain iterations only [32].

## V. CASE STUDY: BATCH-TO-BATCH OPTIMIZATION

This case study investigates the performance of the proposed methodology in a high-dimensional RTO problem. We consider the batch-to-batch optimization of a photobioreactor for the production of phycocyanin (P) by the blue-green cyanobacterium *Arthrospira platensis* (X) growing on nitrates (N). A dynamic model describing the concentrations  $C_X$  [gL<sup>-1</sup>],  $C_N$  [mgL<sup>-1</sup>] and  $C_P$  [mgL<sup>-1</sup>] in the photobioreactor is given by [28]:

$$\dot{C}_X = u_m \frac{I(t)}{I(t) + k_s + I(t)^2/k_i} \frac{C_N(t)}{C_N(t) + K_N} C_X(t) - u_d C_X(t) \quad (12)$$

$$\dot{C}_N = -Y_{N/X} u_m \frac{I(t)}{I(t) + k_s + I(t)^2/k_i} \frac{C_N(t)}{C_N(t) + K_N} C_X + F_N(t) \quad (13)$$

$$\dot{C}_P = k_m \frac{I(t)}{I(t) + k_{sq} + I(t)^2/k_{iq}} C_X(t) - k_d \frac{C_P(t)}{C_N(t) + K_{Np}} \quad (14)$$

where the light intensity  $I(t)$  [ $\mu\text{E m}^2 \text{s}^{-1}$ ] and the nitrate inflow rate  $F_N(t)$  [mgL<sup>-1</sup>h<sup>-1</sup>] are manipulated inputs; and the values of the model parameters  $k_d$ ,  $k_m$ ,  $k_s$ ,  $k_i$ ,  $k_{sq}$ ,  $k_{iq}$ ,  $K_N$ ,  $K_{Np}$ ,  $u_d$ ,  $u_m$ ,  $Y_{N/X}$  are the same as those reported by [28]. For simplicity, the mass-balance equations (12)–(14) neglect the change in volume due to the nitrate addition kinetic model assumes nutrient-replete growth conditions.

The optimization problem seeks to maximize the end-batch concentration of phycocyanin after 240 hours of operation. Regarding constraints, the phycocyanin-to-cyanobacterial-biomass ratio must be kept under 1.1 wt% at all times; the nitrate concentration must be kept under 800 mgL<sup>-1</sup> at all times and below 150 mgL<sup>-1</sup> at the end of the batch; and both manipulated inputs are bounded. A mathematical formulation of this (dynamic) optimization problem is as follows:

$$\min_{I(t), F_N(t)} \mathbb{E}(C_P(240)) \quad (15)$$

s.t. PBR model (12)–(14)

$$C_X(0) = 1, C_N(0) = 150, C_P(0) = 0$$

$$\mathbb{P}(C_P(t) \leq 0.011 C_X(t)) \geq 0.9, \forall t$$

$$\mathbb{P}(C_N(t) \leq 800) \geq 0.9, \forall t$$

$$120 \leq I(t) \leq 400, \forall t$$

$$0 \leq F_N(t) \leq 40, \forall t$$

To recast it as a finite-dimensional optimization problem, both control trajectories are discretized using a piecewise-constant parameterization over 6 equidistant stages (of 60 hours each). The batch-to-batch optimization, therefore, comprises a total of 12 degrees of freedom. The state path constraints are also discretized and enforced at the end of each control stage.

The case study assumes that the concentrations  $C_X$ ,  $C_N$  and  $C_P$  can all be measured during or at the end of the batch as necessary. Process noise is simulated in this virtual environment by adding white noise with zero mean and standard deviation  $\sigma_{C_X} = 0.02$  [g L<sup>-1</sup>],  $\sigma_{C_N} = 0.316$  [mg L<sup>-1</sup>], and  $\sigma_{C_P} = 0.001$  [mg L<sup>-1</sup>]. However, no prior knowledge of this measurement noise is assumed during the construction of the GP surrogates for the cost and constraints.

Next, we implement several strategies to Problem (15), including Algorithm 1 with the use of chance constraints and satisfaction in expectation. The following dynamic model is used for the latter, which presents a structural mismatch with the plant model (12)–(14) regarding the light inhibition:

$$\dot{C}_X = u_m \frac{I(t)}{I(t) + k_s} \frac{C_N(t)}{C_N(t) + K_N} C_X(t) - u_d C_X(t) \quad (16)$$

$$\dot{C}_N = -Y_{N/X} u_m \frac{I(t)}{I(t) + k_s} \frac{C_N(t)}{C_N(t) + K_N} C_X(t) + F_N(t) \quad (17)$$

$$\dot{C}_P = k_m \frac{I(t)}{I(t) + k_{sq}} \frac{c_N}{C_N(t) + K_N} C_X(t) - k_d \frac{C_P(t)}{C_N(t) + K_{Np}} \quad (18)$$

The Matérn kernel (with parameter  $\nu = \frac{3}{2}$ ) is used for both the model and physical system (low and high fidelity, respectively) to construct the multi-fidelity model. Specifically, for the model, 30 data points are used to tune the respective GP's hyperparameters. Notice that now there is no need to involve discretized equations in the NLP problem, as a GP emulates the ode's solution. A multi-start initialization (20 random starting points) is applied to overcome the NLP solver's numerical failures and reduce the likelihood of converging to high laying local optima. The initial GPs are trained with 13 feasible data points, and the initial trust region encloses all of these points.

The performance of Algorithm 1 using the multifidelity GPs and EI acquisition function is compared with three alternative frameworks:

- a) The use of constraint satisfaction in expectation *using* trust-regions.
- b) The use of chance constraints *without using* trust-regions.
- c) The use of chance constraints *using* trust-regions and *without using* a prior model (model-free RTO).

The results for the convergence of the objective are presented in Figure 2 for multiple realizations of the process noise. The circles represent an occurrence of a constraint violation. The proposed framework appears to have the best trade-off between convergence and satisfaction of constraints.

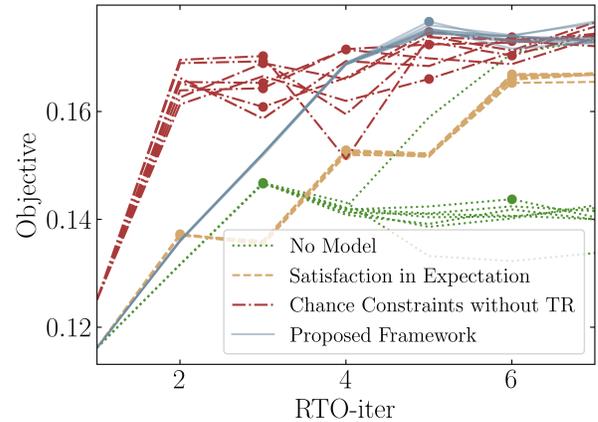


Fig. 2: Evolution of process cost with the RTO iterations. Circles represents the violation of a constraint.

The use of constraint satisfaction in expectation (a), as anticipated, results in increasing constraint violations (compare to the proposed framework); this has a side-effect, it slows down the algorithm as it cannot converge fast to an optimum point as it backtracks often. On the other hand, the proposed framework converges very fast as the chance constraint and trust-region formulation allow the 'safe' exploration with high probability. Scenario (b) explores the performance of the algorithm with the removal of the trust-region. Since the trust region is not restricting the design space, the optimization approaches high values very early in the RTO iterations. However, the GPs are overly confident, and violation of constraints occur often. The last scenario, where an a priori model is ignored (c), violates much less constraints compared to (a) and (b), as the optimization problem considers chance constraints and the trust region. However, the absence of prior model in (c) affects the convergence of the objective function significantly. Notice that the proposed framework managed to violate the least constraints compare to (a- c).

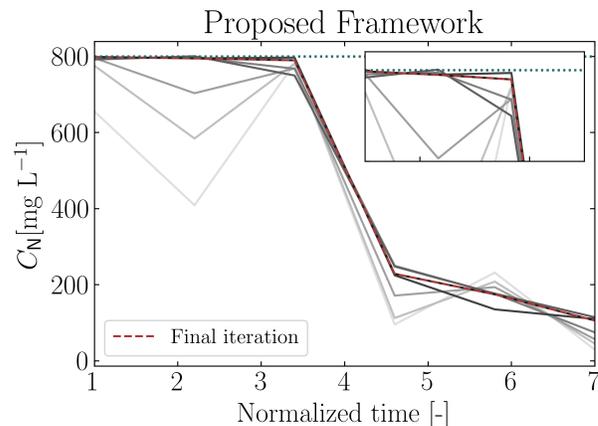
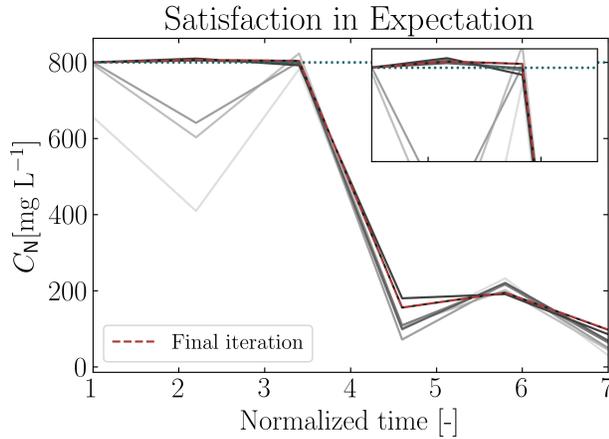
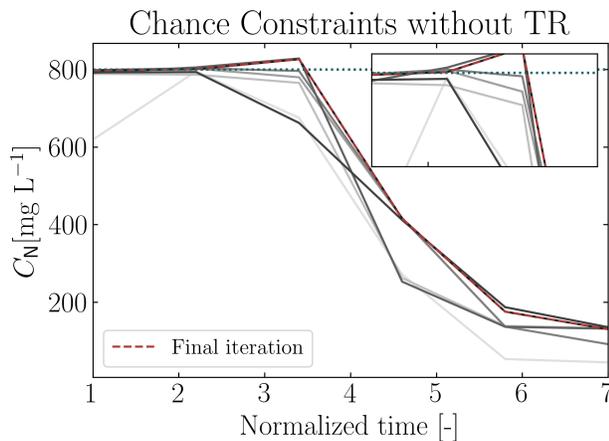


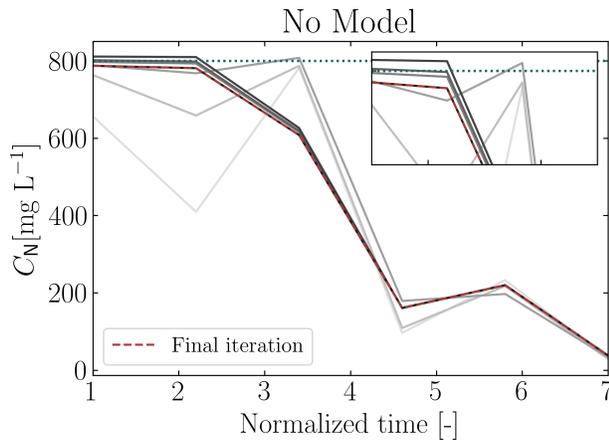
Fig. 3: RTO results for the case study (Problem 15) using our proposed framework. The lines are plotted over the RTO iterations, which are faded out towards earlier iterations



(a) Constraint satisfaction in expectation using trust-regions



(b) Chance constraint satisfaction without using trust-regions



(c) Chance constraint satisfaction without prior model using trust-regions

Fig. 4: RTO results for the photobioreactor case study (Problem 15) using two alternative approaches: **(a)** The probabilistic constraints in (10) are replaced with the satisfaction in expectation. **(b)** The trust-regions are removed from (10). **(c)** The prior model is removed from (10).

The constraint 1 ( $C_N \leq 800$ ) for all the RTO iterations for

a single MC of our proposed framework is depicted in Fig. 3, where the lines are faded out towards earlier iterations. As expected the plant converges fast to a solution that satisfies the constraints and in the beginning the constraints are not close to the bound of the constraint.

The corresponding figures for the constraint using the expected value of the constraints (but with trust-regions), chance constraints without trust regions and the absence of prior model are depicted in Figure 4a - 4c respectively. Notice that the constraint violation occur often in all cases, where in Figure 4b, violates the constraints significantly for the case of the trust-region's absence. Such violation can explain the fast increase in the objective function, as the objective may obtain better values at infeasible regions.

As depicted in Fig. 3 and Fig. 4a - 4c, our proposed frameworks is superior in terms of constraints satisfaction. Therefore it is clear that the proposed framework can significant beneficial real-time optimization, when stochastic environments are present and safe exploration is needed.

## VI. CONCLUSIONS

A new approach is proposed for real-time optimization under plant-model mismatch using multi-fidelity GPs and trust-regions to efficiently incorporate chance constraints even when only low fidelity models are available. The low fidelity model assumed to be a black-box is emulated using a GP so that the optimization of the problem is solved more efficiently. The trust region restricts the design space which results in meaningful predictions for the mean and variance of the GPs, and allows us to reformulate the chance constraints with confidence.

The benefits are analyzed and illustrated with a numerical case study: a challenging batch-to-batch optimization problem with a dozen inputs and constraints. Here we have shown that our proposed framework is a good trade-off between optimality and safe exploration. In practical applications, this added reliability could outweigh the benefits of model-free RTO, for instance, ease of design and maintainability. These results reflect that using a nominal model in the manner of a prior constitutes an effective de-risking strategy in higher-dimensional RTO problems.

## ACKNOWLEDGMENT

This project has received funding from the EPSRC grant project (EP/R032807/1) is gratefully acknowledged.

## REFERENCES

- [1] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson, "RTO: An overview and assessment of current practice," *Journal of Process Control*, vol. 21, no. 6, pp. 874–884, 2011.
- [2] M. M. Câmara, A. D. Quelhas, and J. C. Pinto, "Performance evaluation of real industrial RTO systems," *Processes*, vol. 4, no. 4, pp. 1–20, 2016.
- [3] T. E. Marlin and A. N. Hrymak, "Real-time operations optimization of continuous processes," in *AIChE Symposium Series - CPC-V*, vol. 93, 1997, pp. 156–164.
- [4] C. De Prada, D. Sarabia, G. Gutierrez, E. Gomez, S. Marmol, M. Sola, C. Pascual, and R. Gonzalez, "Integration of rto and mpc in the hydrogen network of a petrol refinery," *Processes*, vol. 5, no. 1, 2017. [Online]. Available: <https://www.mdpi.com/2227-9717/5/1/3>

- [5] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Computers & Chemical Engineering*, vol. 33, no. 10, pp. 1557–1567, 2009.
- [6] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," *Industrial & Engineering Chemistry Research*, vol. 48, no. 13, pp. 6022–6033, 2009.
- [7] C. Tsay, Y. Cao, Y. Wang, J. Flores-Cerrillo, and M. Baldea, "Identification and online updating of dynamic models for demand response of an industrial air separation unit," 2020.
- [8] W. Gao and S. Engell, "Iterative set-point optimization of batch chromatography," *Computers & Chemical Engineering*, vol. 29, no. 6, pp. 1401–1409, 2005.
- [9] A. Marchetti, B. Chachuat, and D. Bonvin, "A dual modifier-adaptation approach for real-time optimization," *Journal of Process Control*, vol. 20, no. 9, pp. 1027–1037, 2010.
- [10] S. Costello, G. François, and D. Bonvin, "A directional modifier-adaptation algorithm for real-time optimization," *Journal of Process Control*, vol. 39, pp. 64–76, 2016.
- [11] J. Speakman and G. François, "Real-time optimization via modifier adaptation of closed-loop processes using transient measurements," *Computers & Chemical Engineering*, vol. 140, p. 106969, 2020.
- [12] W. Gao, S. Wenzel, and S. Engell, "A reliable modifier-adaptation strategy for real-time optimization," *Computers & Chemical Engineering*, vol. 91, pp. 318–328, 2016.
- [13] M. Singhal, A. G. Marchetti, T. Faulwasser, and D. Bonvin, "Real-time optimization based on adaptation of surrogate models," *IFAC-PapersOnLine*, vol. 49, no. 7, pp. 412–417, 2016.
- [14] T. d. A. Ferreira, H. A. Shukla, T. Faulwasser, C. N. Jones, and D. Bonvin, "Real-time optimization of uncertain process systems via modifier adaptation and Gaussian processes," in *Proceedings of European Control Conference*, 2018, pp. 465–470.
- [15] E. A. del Rio Chanona, P. Petsagkourakis, E. Bradford, J. E. A. Graciano, and B. Chachuat, "Real-time optimization meets bayesian optimization and derivative-free optimization: A tale of modifier adaptation," *Computers & Chemical Engineering*, vol. 147, p. 107249, 2021.
- [16] H. A. Shukla, T. d. A. Ferreira, T. Faulwasser, D. Bonvin, and C. N. Jones, "Convergence certificate for stochastic derivative-free trust-region methods based on Gaussian processes," 2020.
- [17] M. L. Thompson and M. A. Kramer, "Modeling chemical processes using prior knowledge and neural networks," *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994.
- [18] P. Petsagkourakis, I. Sandoval, E. Bradford, D. Zhang, and E. del Rio-Chanona, "Reinforcement learning for batch bioprocess optimization," *Computers & Chemical Engineering*, vol. 133, p. 106649, 2020.
- [19] P. Petsagkourakis, I. O. Sandoval, E. Bradford, F. Galvanin, D. Zhang, and E. A. del Rio-Chanona, "Chance constrained policy optimization for process control and optimization," *arXiv preprint arXiv:2008.00030*, 2020.
- [20] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on multi-output gaussian process regression," *Knowledge-Based Systems*, vol. 144, pp. 102–121, 2018.
- [21] M. C. Kennedy and A. O'Hagan, "Bayesian calibration of computer models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 3, pp. 425–464, 2001. [Online]. Available: <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00294>
- [22] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, and G. E. Karniadakis, "Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2198, 2017.
- [23] K. Cutajar, M. Pullin, A. Damianou, N. Lawrence, and J. González, "Deep gaussian processes for multi-fidelity modeling," 2019.
- [24] J. Wu, S. Toscano-Palmerin, P. I. Frazier, and A. G. Wilson, "Practical multi-fidelity bayesian optimization for hyperparameter tuning," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, ser. Proceedings of Machine Learning Research, R. P. Adams and V. Gogate, Eds., vol. 115. Tel Aviv, Israel: PMLR, 22–25 Jul 2020, pp. 788–798. [Online]. Available: <http://proceedings.mlr.press/v115/wu20a.html>
- [25] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, "Learning-based model predictive control: Toward safe learning in control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 269–296, 2020.
- [26] E. Bradford and L. Imsland, "Output feedback stochastic nonlinear model predictive control for batch processes," *Computers & Chemical Engineering*, vol. 126, pp. 434–450, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098135419300286>
- [27] L. Fagiano and M. Khammash, "Nonlinear stochastic model predictive control via regularized polynomial chaos expansions," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, 2012, pp. 142–147.
- [28] E. Bradford, L. Imsland, D. Zhang, and E. A. del Rio-Chanona, "Stochastic data-driven model predictive control using Gaussian processes," *Computers & Chemical Engineering*, vol. 139, p. 106844, 2020.
- [29] F. Berkenkamp, A. Krause, and A. P. Schoellig, "Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics," 2016. [Online]. Available: <http://arxiv.org/abs/1602.04450>
- [30] A. W. Marshall, I. Olkin, and B. C. Arnold, *Inequalities: Theory of Majorization and its Applications*, 2nd ed. Springer, 2011, vol. 143.
- [31] M. A. Alvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: a review," 2012.
- [32] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2016.
- [33] L. L. Gratiet and J. Garnier, "Recursive co-kriging model for design of computer experiments with multiple levels of fidelity," *International Journal for Uncertainty Quantification*, vol. 4, no. 5, pp. 365–386, 2014.
- [34] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," *32nd International Conference on Machine Learning, ICML 2015*, vol. 2, pp. 997–1005, 2015.
- [35] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, "Gaussian process optimization in the bandit setting: No regret and experimental design," in *Proceedings of 27th International Conference on International Conference on Machine Learning*. Madison (WI): Omnipress, 2010, pp. 1015–1022.
- [36] M. K. Titsias, "Variational learning of inducing variables in sparse Gaussian processes," in *Journal of Machine Learning Research*, ser. Proceedings of Machine Learning Research, D. van Dyk and M. Welling, Eds., vol. 5. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, 2009, pp. 567–574. [Online]. Available: <http://proceedings.mlr.press/v5/titsias09a.html>
- [37] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *Journal of Global optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [38] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. MOS-SIAM Series on Optimization, 2009.