

Convergence Certificate for Stochastic Derivative-Free Trust-Region Methods based on Gaussian Processes

Harsh A. Shukla¹, Tafarel de Avila Ferreira², Timm Faulwasser³,
Dominique Bonvin¹ and Colin N. Jones¹

Abstract—In many machine learning applications, one wants to learn the unknown objective and constraint functions of an optimization problem from available data and then apply some technique to attain a local optimizer of the learned model. This work considers Gaussian Processes (GPs) as global surrogate models and utilizes them in conjunction with derivative-free trust-region methods. It is well known that derivative-free trust-region methods converge globally—provided the surrogate model is probabilistically fully linear. We prove that GPs are indeed probabilistically fully linear, thus resulting in fast (compared to linear or quadratic local surrogate models) and global convergence. We draw upon the optimization of a chemical reactor to demonstrate the efficiency of GP-based trust-region methods.

I. INTRODUCTION

Increased computational power, ubiquitous availability of computational resources and improved algorithms have driven steady research interest in real-time optimization. However, in essentially all real-world applications, accurate plant models are not available. Hence, the issues surrounding uncertain models have been explored in different settings ranging from robust and stochastic optimization [1] via real-time optimization [2], data-driven control [3] to machine learning [4].

The set of Derivative-Free Optimization (DFO) trust-region methods comprises established tools to optimize unknown—or expensive to evaluate—objectives [5]. The pivotal idea is the use of a local surrogate model, built at each iteration by evaluating the objective at a number of sample points within the trust region. Probabilistic derivative-free trust-region methods rely on randomized surrogate models [6], [7]. The key advantage of using a probabilistic method is its ability to capture uncertainties efficiently. This is indeed useful for noisy objectives and/or inaccurate models. However, the key bottleneck of deterministic and probabilistic derivative-free trust-region methods alike is twofold: (i) ensuring the quality of the surrogate model, and (ii) guaranteeing a sufficiently large domain of validity. The former can be achieved via

complicated procedures for sample-set maintenance [5], while the latter calls for global surrogate models.

The convergence of trust-region methods relies on the accuracy of the surrogate model within the trust region. Intuitively speaking, the convergence mechanism increases the sampling of the unknown function and decreases the trust-region radius until the local surrogate model is sufficiently accurate in zeroth- and first-order compared to the unknown function. This accuracy, which is defined as “full linearity”, helps move in a decent direction. Global convergence of derivative-free trust-region methods for deterministic and stochastic version is described in [5] and [6], respectively. In this context, the main challenge is the construction of a surrogate model by performing as few plant evaluations as possible. Hence, if full-linearity can be certified, a global surrogate model is usually preferred over local ones.

At the same time, there is a recent and steadily growing interest in machine learning techniques in computer science as well as in systems and control. This spans {supervised, reinforcement} learning and data-driven function approximations by deep neural networks [8] and GPs [9], [10]. There exists a body of literature on using machine learning for optimization, e.g., [10], [11]. In some cases, it is possible to guarantee convergence to the global minimum of unknown functions.

Since GPs are excellent candidates to be used as global surrogate models, it is natural to combine them with derivative-free trust-region methods. The idea, which dates back to Conn’s book [5], was analyzed empirically in [12]. It is also used in Real-Time Optimization (RTO), where the aim is to solve a steady-state optimization problems or to optimize repeated batch operation [13], [14]. However, to the best of our knowledge, it is yet to be shown whether GPs can be certified to be fully linear, which is key for guaranteeing global convergence of derivative-free trust-region methods.

The main result of this paper is to prove that GPs can satisfy the fully-linearity property. We present the necessary procedure for this certification. Furthermore, using GP as a global surrogate model leads to fewer trust-region iterations, implying fewer plant evaluations. This has a clear advantage over local surrogate models and other empirical local model correction methods as illustrated in the numerical section.

The remainder is structured as follows: We formulate the problem, discuss a probabilistic derivative-free trust-region method, introduce GPs, and explain the need for improvement in Section II. A certification proof for GPs to be fully-linear surrogate model is provided in Section III. We illustrate

The first and fifth author acknowledge funding from the People Programme (Marie Curie Actions) of the EU Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 607957 (TEMPO).

¹ Harsh A. Shukla, Dominique Bonvin, and Colin N. Jones are with Laboratoire d’Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland. {harsh.shukla, dominique.bonvin, colin.jones}@epfl.ch

² Tafarel de Avila Ferreira is with the Group of Energy Materials, École Polytechnique Fédérale de Lausanne, CH-1951 Sion, Switzerland. tafarel.deavilaferreira@epfl.ch

³ Timm Faulwasser is with the Institute for Energy Systems, Energy Efficiency and Energy Economics, TU Dortmund University, 44227 Dortmund, Germany. timm.faulwasser@ieee.org

the effectiveness and advantage of the proposed work on a numerical case study in Section IV, while Section V concludes the paper.

II. PRELIMINARIES

A. Problem statement

We consider the Nonlinear Program (NLP)

$$\min_{x \in \mathbb{R}^{n_x}} f(x), \quad (1)$$

with the unknown objective $f : \mathbb{R}^{n_x} \rightarrow \mathbb{R}$ and the decision variables $x \in \mathbb{R}^{n_x}$. A solution to (1) can be computed using DFO by sampling the unknown function f and building a surrogate model. The samples are subject to additive noise and therefore their distribution can be written as:

$$z = f(x) + \nu \quad \text{where } \nu \sim \mathcal{N}(0, \sigma^2). \quad (2)$$

Surrogate models usually depend (implicitly or explicitly) on a—yet to be specified—number of past data points,

$$\mathbb{D}_k = \{(x_{k-l-1}, z_{k-l-1}), \dots, (x_k, z_k)\}, \quad (3)$$

where $z_{(k)}$ is a realization of the random variable z at time instant k . Hence, by building the surrogate model $m : \mathbb{R}^{n_x} \times \mathbb{R}^{(n_x+1) \times l} \rightarrow \mathbb{R}$, the solution to problem (1) becomes

$$x_{k+1} = \arg \min_{x \in \mathbb{R}^{n_x}} m_k(x), \quad (4)$$

where the shorthand $m_k(x) := m(x, \mathbb{D}_k)$ is used.

Remark 1 (*Applicability of the considered setting*):

At first glance, the problem setting outline above might look restrictive as it focuses on unconstrained optimization in real vector spaces. Although problem (4) does not explicitly incorporate constraints, one may convert constrained optimization problems into unconstrained ones using penalty functions; see [15].

Moreover, whenever one aims at optimizing the performance of a repeated (batch) process or of a periodic process, $\dot{x} = f_p(x, v)$, $x(0) = x_0$, one will typically start with an optimal control problem, which—after applying direct discretization techniques in conjunction with ideas from sequential algorithms for numerical optimal control—can be cast in a mathematically equivalent form to (1) and (4), see e.g. [16]. $\square$

B. Surrogate modeling with GPs

Unlike parametric identification techniques, where data are discarded after constructing the model, GPs are kernel-based methods that use all available data (or subset thereof) to learn a map between input and output data. We will briefly introduce GPs and refer to [10], [17] for further details.

Let $\mathbf{x} \in L^2(\mathcal{D}, \mathbb{P}; \mathbb{R}^{n_x})$ denote random variables and $x := \mathbf{x}(\omega) \in \mathbb{R}^{n_x}$ their realizations, where $L^2(\mathcal{D}, \mathbb{P}; \mathbb{R}^{n_x})$ is the underlying Hilbert space of random variables with finite variance.

Considering l available samples, the input-output data generated by f is $\mathbb{D}$ from (3) (dropping the iteration index k). Let $\mathbb{X}$ be the projection of $\mathbb{D}$ onto $\mathbb{R}^{n_x}$ and let $\bar{z} \in \mathbb{R}^l$ be the projection of $\mathbb{D}$ in direction of z . We use GP regression to establish a relationship between $\mathbb{X}$ and $\bar{z}$ and obtain a

corresponding conditional distribution of z for a new query input point x , that is,

$$z|x, \mathbb{X}, \bar{z} \sim \mathcal{N}(\mu_z, \sigma_z^2), \quad (5)$$

where the mean and the variance of z are

$$\mu_z := \mathbb{E}[z] = \bar{c}^\top (\bar{C} + \sigma^2 I_p)^{-1} \bar{z}, \quad (6a)$$

$$\sigma_z^2 := \mathbb{V}[z] = \kappa(x) - \bar{c}^\top (\bar{C} + \sigma^2 I_p)^{-1} \bar{c}. \quad (6b)$$

Here, $\bar{C} \in \mathbb{R}^{l \times l}$ is a covariance matrix with elements $\bar{C}_{ij} = c(x_i, x_j)$, $\bar{c} = [c(x, x_1), c(x, x_2), \dots, c(x, x_l)]^\top \in \mathbb{R}^{l \times 1}$, $\kappa(x) = c(x, x)$, where $c(\cdot, \cdot)$ is a covariance function denoted as *kernel*. For example, the squared exponential covariance function with automatic relevance determination is defined as

$$c(x_i, x_j) = \sigma_f^2 \exp\left(-\frac{(x_i - x_j)^\top \Lambda (x_i - x_j)}{2}\right), \quad (7)$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_x})$. The hyperparameters $\theta = [\sigma_f, \lambda_{1:n_x}] \in \mathbb{R}^{n_x+1}$ need to be learned/estimated from the data $\mathbb{D}$ during the training phase. Since the covariance matrix $\bar{C}$ and the covariance vector $\bar{c}$ depend on the hyperparameters θ and the input data $\mathbb{X}$, one can also write $\bar{C}$ as $\bar{C}(\theta, \mathbb{X})$. To this end, consider $Q(\theta, \mathbb{X}) := \bar{C}(\theta, \mathbb{X}) + \sigma^2 I_l$ and the log-marginal likelihood

$$\mathcal{L}(\theta, \mathbb{X}, \bar{z}) = -\frac{1}{2} \bar{z}^\top Q(\theta, \mathbb{X})^{-1} \bar{z} - \frac{1}{2} \log |Q(\theta, \mathbb{X})| - \frac{n}{2} \log 2\pi.$$

Given $\mathbb{X}$ and $\bar{z}$, the parameters are learned by maximizing the log-marginal likelihood,

$$\theta^* = \arg \max_{\theta} \mathcal{L}(\theta, \mathbb{X}, \bar{z}). \quad (8)$$

The above nonconvex maximization problem can be solved using deterministic as well as stochastic methods. We refer to [10, Chap. 7] and [18, Chap. 2] for further details and for insights into the convergence properties.

In summary, the optimal hyperparameters θ^* are found by training a GP. Then, for the query point x , the GP provides the output z with a normal distribution. The mean and covariance of the normal distribution is computed using the hyperparameters θ^* and finding how close x is compared to the data set. This way, the GP computes the output distribution of z with more weight on the nearest inputs.

A first advantage of using GPs compared to fixed-structure parametric models is that GP models are able to capture complex nonlinear input-output relationships through the use of only a few parameters. This happens because the predicted output is influenced more by the nearby input-output pairs obtained from the training data set. A second advantage is that, generally, they offer an interesting trade-off between exploration and exploitation [10]. A third advantage, particularly in DFO setting, is the fact GPs constitute global surrogate models. The main drawback of GPs is the computational complexity growing as a cubic function of the number of data points N , that is, the complexity is $\mathcal{O}(N^3)$. However, this can be addressed using sparse GPs [10].

In what follows, at each iteration k , we use the GP mean as a surrogate model, that is,

$$m_k(x) := \mu_z(x, \mathbb{D}_k), \quad (9)$$

where the notation $\mu_z(x, \mathbb{D}_k)$ highlights that, for fixed hyperparameters, μ_z from (6a) takes x as argument—via $\bar{c}$ —and depends on the data set $\mathbb{D}_k$ —via $\bar{c}$ and $\bar{C}$.

If the considered function samples obtained via (2) are indeed subject to additive noise, the data $\mathbb{D}_k$ will contain l samples that correspond to realizations of random variables. Hence, the uncertainty surrounding the data $\mathbb{D}_k$ induces the probabilistic nature of the surrogate model and, consequently, the model available at iteration k can be regarded as $m_k : \mathbb{R}^{n_x} \rightarrow L^2(\mathcal{D}, \mathbb{P}; \mathbb{R})$. Conceptually, its realization can be denoted as $m_k := m_k(\omega)$. This point of view leads naturally to probabilistic DFO methods.

C. Derivative-free probabilistic trust-region methods

A standard version of probabilistic derivative-free trust-region method is summarized in Algorithm 1, cf. [6], [7]. The main idea is to approximate the unknown function via $m_k(x)$ within a certain neighborhood of x_k (a.k.a. the trust region). Whenever the surrogate model fails approximating the original problem, then the trust region is shrunk and the process repeated. Next, we recall the main points of the convergence analysis given in [7].

Assumption 1 (Differentiability of f [7]): The unknown function f has bounded level sets and the gradient ∇f is Lipschitz continuous with constant L_g . $\square$

Assumption 2 (Noise with finite variance [7]): The additive noise ν observed while measuring f is drawn from a normal distribution with zero mean and finite variance. $\square$

For the remainder, we define $B(x; \Delta)$ as the ball of radius Δ centered at $x \in \mathbb{R}^n$. Furthermore, $\mathcal{C}^k$ denotes the set of functions on $\mathbb{R}^n$ with k continuous derivatives and $\mathcal{LC}^k$ denotes the set of functions in $\mathcal{C}^k$ such that the k th derivative is Lipschitz continuous.

Definition 1 (κ fully-linear model [7]): Consider f satisfying Assumption 1. Let $\kappa = (\kappa_{ef}, \kappa_{eg}, \nu_1^m)$ be a given vector of constants and let $\Delta > 0$ be given. A model $m \in \mathcal{LC}^1$ with Lipschitz constant ν_1^m is a κ fully-linear model of f on $B(x; \Delta)$ if for all $s \in B(0; \Delta)$,

$$|f(x+s) - m(x+s)| \leq \kappa_{ef} \Delta^2, \text{ and} \quad (10a)$$

$$\|\nabla f(x+s) - \nabla m(x+s)\| \leq \kappa_{eg} \Delta. \quad (10b)$$

$\square$

The above definition is key in the convergence analysis for the case of probabilistic surrogate models. The main idea is to show that these models have good accuracy with sufficiently high probability [6]. Since derivative-free trust-region algorithms sample and collect data at each iteration, let F_{k-1}^M denote the realization of events during the first $k-1$ iterations of the algorithm. Now, we are ready to define a probabilistic κ fully-linear surrogate model.

Definition 2 (κ fully-linear model with probability α [7]): A sequence of random models $\{m_k\}$ is κ fully linear with probability α on $\{B(x_k, \Delta_k)\}$ if the events

$$S_k = \{m_k \text{ is a } \kappa \text{ fully-linear model of } f \text{ on } B(x_k, \Delta_k)\}$$

satisfy the condition $\mathbb{P}(S_k | F_{k-1}^M) \geq \alpha$ for all k sufficiently large. $\square$

Algorithm 1 Derivative-Free Trust-Region Method [7]

Data: Initial model m_0 , initial point x_0 , and constants $0 < \gamma_{dec} < 1 < \gamma_{inc}$, $0 < \eta < \beta < 1$, $0 < \Delta_0$ and $\alpha \in (0, 1)$ satisfying (11). Set $k = 0$.

1) **Model building:** Build m_k , a κ fully-linear model with probability α_k on $B(x_k; \Delta_k)$, for some $\alpha_k \in (0, 1)$ such that $\alpha_k \geq \alpha$ for sufficiently large k .

2) **Step calculation:**

$$s_k := \arg \min_{s: \|s\| \leq \Delta_k} m_k(x_k + s) \quad (12)$$

3) **Compute model decrement:**

- a) If $m_k(x_k) - m_k(x_k + s_k) < \beta \min \{\Delta_k, \Delta_k^2\}$ then $x_{k+1} = x_k$; $\Delta_{k+1} = \gamma_{dec} \Delta_k$ and go to Step 6.
- b) Else go to Step 4).

4) **Estimate improvement after plant evaluation:** Evaluate

$$\rho_k = \frac{F_k^0 - F_k^{s_k}}{m_k(x_k) - m_k(x_k + s_k)}. \quad (13)$$

5) **Trust region and step update:**

- If $\rho_k \geq \eta$, then $x_{k+1} = x_k + s_k$ and $\Delta_{k+1} = \gamma_{inc} \Delta_k$.
- If $\rho_k < \eta$, then $x_{k+1} = x_k$ and $\Delta_{k+1} = \gamma_{dec} \Delta_k$.

6) **Setting index:** $k = k+1$ and go to Step 1.

Next, we introduce Algorithm 1. The main idea is to build a surrogate model within the trust-region radius and use it to compute a minimizer. As long as the objective decreases sufficiently, accept the step and increase the trust-region radius, otherwise decrease the radius and reject the step. The challenge stems from the probabilistic nature of the surrogate model, in particular from the fact that the confidence in the model is probabilistic. This hinders increasing the trust-region radius significantly. Hence, it is important to have a relationship between the probability α (confidence in the surrogate model) and $\gamma_{inc}/\gamma_{dec}$ (increment/decrement of the radius). This relationship reads [7]:

$$\alpha \geq \left\{ \frac{1}{2}, 1 - \frac{\frac{\gamma_{inc}-1}{\gamma_{inc}}}{4 \left[\frac{\gamma_{inc}-1}{2\gamma_{inc}} + \frac{1-\gamma_{dec}}{\gamma_{dec}} \right]}, 1 - \frac{1-\gamma_{dec}}{2(\gamma_{inc}^2 - \gamma_{dec})} \right\}. \quad (11)$$

Remark 2: A careful look at Step 1 of Algorithm 1 reveals that we need to build a κ fully-linear model only for sufficiently large k . This allows having a relatively inaccurate model at the beginning, thereby avoiding unnecessary sampling as long as there is sufficient improvement.

Theorem 1 (Global convergence [7]): If Assumptions 1-2 are satisfied, and α is chosen to satisfy (11), then $\{\|\nabla f(x_k)\|\}$ converges in probability to zero. That is, for all $\epsilon > 0$, $\lim_{k \rightarrow \infty} \mathbb{P}[\|\nabla f(x_k)\| > \epsilon] = 0$. $\square$

At this point a pivotal question arises: how to build a κ fully-linear surrogate model with probability α ? Details of building and certifying a probabilistic local surrogate

model at each iteration—mainly via linear and nonlinear interpolation/regression—are given in [6], [7]. Here, we aim at reducing the number of expensive plant evaluations by constructing a global instead of a local surrogate model. For that, we will use a GP as the surrogate model. We will also show how to certify a GP as a probabilistic fully-linear model. To the best of the authors' knowledge, this is still an open question, although GPs have been used in a derivative-free trust-region framework [12], [14].

III. CERTIFICATION PROOF

We certify that GPs are probabilistic fully-linear models. We remind the reader that we use the GP mean as the surrogate model, that is, $m(x) := \mu_z(x, \mathbb{D}_k)$.

Definition 3 (Reproducing kernel Hilbert Space [10]):

Let $\mathcal{H}$ be the Hilbert space of real functions f defined on the index set X . Then, $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS) endowed with an inner product $\langle \cdot, \cdot \rangle$ (and norm $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$) if there exists a function $c : X \times X \rightarrow \mathbb{R}$ with the following properties:

- 1) for every x , $c(x, x')$ as a function of x' belongs to $\mathcal{H}$, and
- 2) k has the reproducing property $\langle f(\cdot), c(\cdot, x) \rangle_{\mathcal{H}} = f(x)$.

□

The aim is to show that (10a) and (10b) hold with probability at least α when the GP mean is used as a surrogate model. For this, the following two properties are assumed.

Assumption 3 (Bounded RKHS norm [19]):

The unknown function $f(x)$ has a known bounded RKHS norm ζ under a known kernel c , that is, $\|f(x)\|_c \leq \zeta < \infty$. □

Assumption 4 (Lipschitzness of the mismatch function):

The mismatch function $h(x) := f(x) - m(x)$ has Lipschitz continuous gradient with constant γ_{lh} . Furthermore, the sequence x_k generated by applying Algorithm 1 satisfies $\|\nabla^2 h(x_k)\| \leq \kappa_{bhh} < \infty$, that is, the mismatch function has a bounded Hessian. □

Assumption 4 is not very strong and is a consequence of Assumption 1: the unknown function $f(\cdot)$ has Lipschitz continuous gradient with bounded Hessian. Note that most of the practically used kernels (e.g. Matern, squared exponential) have Lipschitz continuous gradients [10]. Before deriving the main result, we first state that the distance between an unknown function and the mean is bounded by the GP variance with some probability $1 - \delta$.

Lemma 1 (Bound on mismatch function [19]):

Let Assumption 3 holds and let $\delta \in (0, 1)$. It follows that $\mathbb{P}\left\{|m(x) - f(x)| \leq \sqrt{\beta(N, \delta)} \sigma_z(x, N)\right\} \geq 1 - \delta$. □

Here, $\sqrt{\beta(N, \delta)}$ depends on the number of samples, the probability δ and the RKHS norm $\|f\|_c$, see [19] for details. If the unknown function f is sampled from a GP, one can compute β in closed form [20]. We note that for highlighting

the dependence of β and σ_z on the number of samples and the probability δ , we simply write them as $\beta(N, \delta)$ and $\sigma_z(x, N)$.

Theorem 2 (GP is κ fully linear with probability α):

Let Assumptions 3 and 4 hold. If $0 < \Delta < \frac{6}{\gamma_{lh}}(\kappa_{eg} - 2\kappa_{ef} - \kappa_{bhm})$, then there exists a positive integer $N < \infty$ such that, after N sampling steps, a GP can be certified κ fully linear with probability α . □

Proof: Following Definition 2, the goal is to prove that equations (10a) and (10b) hold with probability at least α .

Let us start with equation (10a) and consider any point within the trust region, that is, $x \in B(x_k, \Delta_k)$. Increased sampling will validate the probability bound in Lemma 1. Upon performing N plant evaluations and applying Algorithm 1 in [19] with $\alpha \leq 1 - \delta$, the following holds with probability α for a given κ_{ef} and Δ :

$$|h(x)| = |m(x) - f(x)| \leq \sqrt{\beta(N, \delta)} \sigma_z(x, N) \leq \kappa_{ef} \Delta^2, \quad (14)$$

which certifies equation (10a) with probability α .

Next we turn to equation (10b) and take any $x, x_s \in B(x_k, \Delta_k)$ such that $x_s = x + s$. Taylor's expansions give:

$$\begin{aligned} h(x+s) &= h(x) + s^\top \nabla h(x) + s^\top \nabla^2 h(x) s + \mathcal{O}(s^3) \\ |s^\top \nabla h(x)| &= |h(x+s) - h(x) - s^\top \nabla^2 h(x) s - \mathcal{O}(s^3)| \\ &\leq |h(x+s)| + |h(x)| + |s^\top \nabla^2 h(x) s| + |\mathcal{O}(s^3)| \\ &\leq |h(x+s)| + |h(x)| + |s^\top \nabla^2 h(x) s| + \frac{\gamma_{lh}}{6} \|s\|^3, \end{aligned}$$

where the first inequality comes from norm properties and the second using Lemma 4.1.14 in [21]. Substituting $s := \frac{\nabla h(x) \Delta}{\|\nabla h(x)\|}$ by following Lemma 4.7 in [22] gives,

$$\begin{aligned} \Delta \|\nabla h(x)\| &\leq |h(x+s)| + |h(x)| + \Delta^2 \|\nabla^2 h(x)\| + \frac{\gamma_{lh}}{6} \Delta^3 \\ \Delta \|\nabla h(x)\| &\leq |h(x+s)| + |h(x)| + \kappa_{bhh} \Delta^2 + \frac{\gamma_{lh}}{6} \Delta^3. \end{aligned}$$

Here, the last inequality arises because of Assumption 4. As shown in the first part of this proof, one can guarantee that $|h(x+s)|, |h(x)| \leq \kappa_{ef} \Delta^2$ with at least probability $1 - \delta$. Hence, the following holds with probability at least $(1 - \delta)^2$:

$$\begin{aligned} \Delta \|\nabla h(x)\| &\leq 2\kappa_{ef} \Delta^2 + \Delta^2 \kappa_{bhh} + \frac{\gamma_{lh}}{6} \Delta^3 \\ \|\nabla h(x)\| &\leq 2\kappa_{ef} \Delta + \Delta \kappa_{bhh} + \frac{\gamma_{lh}}{6} \Delta^2. \end{aligned}$$

Choosing δ such that $\alpha \leq (1 - \delta)^2$ and combining the above with Definition 2 and (10b), it remains to show that, for a given κ_{eg} , the following criterion can be satisfied:

$$2\kappa_{ef} \Delta + \Delta \kappa_{bhh} + \frac{\gamma_{lh}}{6} \Delta^2 \leq \kappa_{eg} \Delta.$$

Since $0 < \Delta < \frac{6}{\gamma_{lh}}(\kappa_{eg} - 2\kappa_{ef} - \kappa_{bhm})$, the above inequality is satisfied. Hence, equation (10b) holds with probability at least α , which concludes the proof. ■

Remark 3 (Computing β and finding maximum $\sigma_z(x)$):

$\sqrt{\beta(N, \delta)}$ is not a function of x . However, we need to determine the maximum of $\sigma_z(x, N)$ over x within the trust

region. This problem has been tackled rigorously in the machine learning community, see [19] for details. However, for our application, one need not explicitly compute these quantities. Another way to look at it is that one can always choose arbitrarily large κ_{ef} and κ_{eg} such that (10a) and (10b) are satisfied with probability at least α . $\square$

Remark 4 (Condition on Δ in Theorem 2):

The condition on the trust-region radius Δ in Theorem 2 does not limit/restrict the algorithm significantly. The reason is that one can choose arbitrarily large values of κ . Moreover, the trust-region radius almost surely goes to zero [Lemma 4 [7]]. Hence, for any positive κ , the condition on the trust region is almost surely satisfied. $\square$

Using GP in the framework of Algorithm 1 yields almost surely convergence. Moreover, it has two main advantages: (i) since GPs approximate unknown functions globally, one does not need to sample after each trust-region iterations as opposed to standard trust-region approaches, where n and $\frac{(n+1)^2}{2}$ data points are required for linear interpolation and nonlinear polynomial-based regression, respectively. This saves a significant amount of plant evaluations; (ii) from an implementation point of view, there is no need to build a model at each trust-region iteration. This is due to the fact that the GP mean converges to the exact function in the limit, as per Lemma 1 and for $\alpha_k > \alpha$ for sufficiently large k . In fact, to implement the algorithm after a failed iteration, one simply needs to sample (not necessarily in the trust-region radius) a few points. Hence, the computation of $\beta(N, \delta)$ and of the maximal variance $\sigma_z(x, N)$ mentioned in the proof of Theorem 2 can be avoided. This obviously reduces the computational burden.

IV. NUMERICAL CASE STUDY

We apply the proposed method to the acetoacetylation of pyrrole with diketene [23]. It is a batch-to-batch optimization of a semi-batch reactor process with 4 reactions: $A + B \xrightarrow{k_1} C$, $2B \xrightarrow{k_2} D$, $B \xrightarrow{k_3} E$, and $B + C \xrightarrow{k_4} F$. The involved species are A: pyrrole; B: diketene; C: 2-acetoacetyl pyrrole; D: dehydroacetic acid; E: oligomers; F: undesired by-product. The material balance equations for the plant read [23]:

$$\begin{aligned}\dot{c}_A &= -k_1 c_A c_B - \frac{F}{V} c_A, \\ \dot{c}_B &= -k_1 c_A c_B - 2k_2 c_B^2 - k_3 c_B - k_4 c_B c_C + \frac{F}{V} (c_B^{\text{in}} - c_B), \\ \dot{c}_C &= k_1 c_A c_B - k_4 c_B c_C - \frac{F}{V} c_C, \\ \dot{c}_D &= k_2 c_B^2 - \frac{F}{V} c_D, \quad \dot{V} = F.\end{aligned}\tag{15}$$

It is assumed that the last two reactions are unknown (structural mismatch), thereby leading to the following plant model: $A + B \xrightarrow{k_1} C$, $2B \xrightarrow{k_2} D$.

We are interested in finding the feed profile of species B such that it maximizes the amount of C at final time, while maintaining the concentration of B and D below specified

threshold values at terminal time. The resulting problem is:

$$\begin{aligned}\max_{F(t)} J &:= c_C(t_f) V(t_f) \\ \text{subject to: model equations (15),} \\ c_B(t_f) &\leq c_B^{\text{max}}, \quad c_D(t_f) \leq c_D^{\text{max}}, \quad 0 \leq F(t) \leq F^{\text{max}}.\end{aligned}\tag{16}$$

The optimal input profile is assumed to have three parts: a first arc with $F = F^{\text{max}}$, a second arc where the feeding is between 0 and F^{max} , and a third arc with zero feeding, see [23] for details. Accordingly, we can define three decision variables, namely, $\pi := (t_m, t_s, F)$, where t_m represents the switching time between the first and second arcs, t_s the switching time between the second and third arcs, and F the assumed constant feeding rate during the second arc. We consider two different scenarios as listed in Table I. We compare the three-arc solution with 100 piecewise-constant control parametrization in Figure 1. Since the two control parameterizations offer similar performance, we use the three-arc parametrization in this work. We reformulate Problem (16) in unconstrained optimization by incorporating the constraints as a penalty term in the cost function. The GP learns the mismatch between the plant and model costs. The plant measurements are corrupted with 5% zero-mean Gaussian additive noise.

The GP is trained using 20 random points around the model optimum for Scenario I. The GP is assumed to be unaware of the change from Scenario I to Scenario II at batch/iteration 8. After each step of the trust-region algorithm, a new data point is incorporated in the GP if it is sufficiently far from the previous data to avoid overfitting. Parameters of Algorithm 1 are: $\eta = 0.5$, $\gamma_{\text{dec}} = 0.9$, $\gamma_{\text{inc}} = 3$, and $\Delta_0 = 3.5$.

Plant efficiency deteriorates significantly for Scenario II when the model-based optimized input is applied as shown in Figure 2. Furthermore, although the GP model is unaware of the change in scenarios, it quickly learns the new plant operating condition (Figure 2), and significantly outperforms the model-based approach (Figure 3).

Scenario	k_3	k_4	Batch
Scenario I	0.01	0.009	1-7
Scenario II	0.28	0.001	8-22

TABLE I: Uncertain reaction constants and batch numbers for the two scenarios.

V. CONCLUSIONS

This paper has investigated convergence a certificate for stochastic derivative-free trust-region methods based on Gaussian Processes. To the best of our knowledge, this work is the first to show that GPs are indeed probabilistic fully-linear models. This in turn allows inferring global convergence of trust-region methods in an almost surely sense. We have demonstrated the efficacy of GPs as surrogate models, drawing upon repeated open-loop optimal control of a chemical batch reaction process.

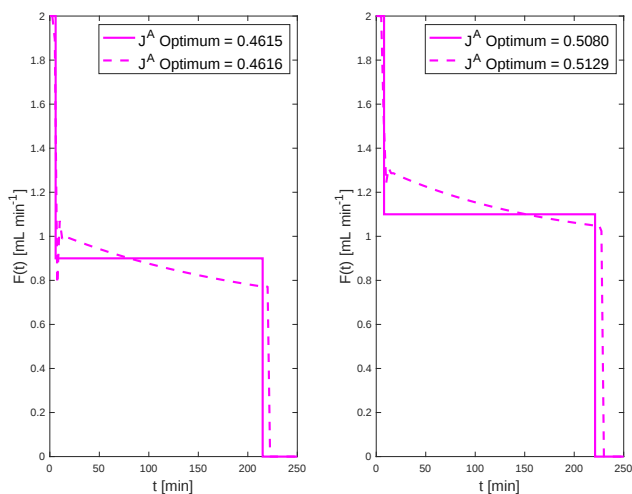


Fig. 1: Input profiles for two different control parameteriza-

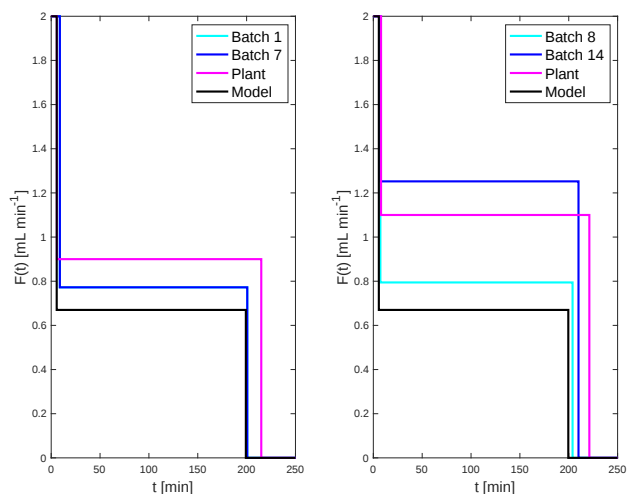


Fig. 3: Input profiles for the two scenarios

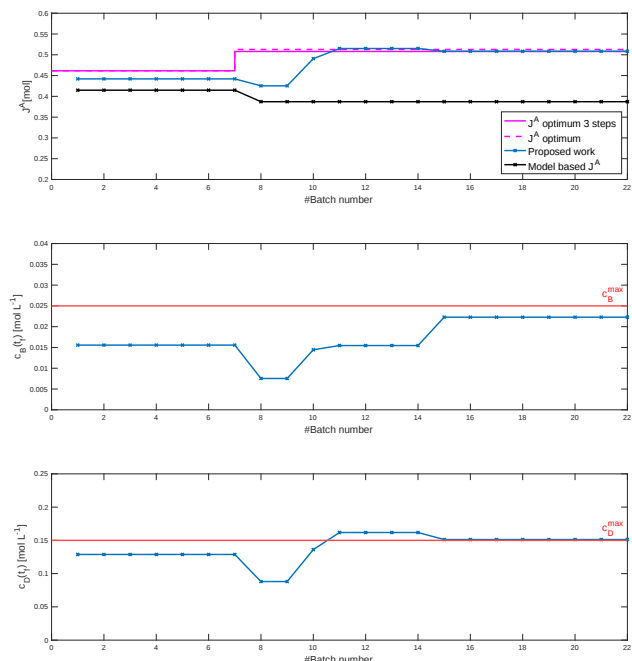


Fig. 2: Learning and optimization using GP as a surrogate model. The upper plot illustrates the cost profile, while the middle and lower plots show the constraints profiles. GP reaches near plant optimality within three iterations.

REFERENCES

- [1] N. V. Sahinidis, "Optimization under uncertainty: State-of-the-art and opportunities," *Comput. Chem. Eng.*, vol. 28, no. 6, pp. 971–983, 2004.
- [2] D. Bonvin, "Special issue on real-time optimization," *Processes*, vol. 5, p. 27, 2017.
- [3] M. Ellis, H. Durand, and P. D. Christofides, "A tutorial review of economic model predictive control methods," *J. Process Control*, vol. 24, pp. 1156–1178, 2014.
- [4] C. Ning and F. You, "Optimization under uncertainty in the era of big data and deep learning: When machine learning meets mathematical programming," *Comput. Chem. Eng.*, vol. 125, pp. 434–448, 2019.
- [5] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*. Philadelphia, PA, USA: SIAM, 2009.
- [6] A. Bandeira, K. Scheinberg, and L. Vicente, "Convergence of trust-

- region methods based on probabilistic models," *SIAM J. Optimization*, vol. 24, no. 3, pp. 1238–1264, 2014.
- [7] J. Larson and S. C. Billups, "Stochastic derivative-free optimization using a trust region framework," *Comput. Optim. Appl.*, vol. 64, no. 3, pp. 619–645, Jul. 2016.
- [8] S. Lucia and B. Karg, "A deep learning-based approach to robust nonlinear model predictive control," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 511–516, 2018.
- [9] L. Hewing, A. Liniger, and M. N. Zeilinger, "Cautious NMPC with gaussian process dynamics for autonomous miniature race cars," *IEEE*, pp. 1341–1348, 2018.
- [10] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.
- [11] S. Shalev-Shwartz and N. Srebro, "SVM optimization: Inverse dependence on training set size," in *Proc. 25th Int. Conf. on Machine Learning*. New York, NY, USA: ACM, 2008, pp. 928–935.
- [12] F. Augustin and Y. M. Marzouk, "A trust-region method for derivative-free nonlinear constrained stochastic optimization," *ArXiv e-prints*, Mar. 2017.
- [13] T. de Avila Ferreira, H. Shukla, T. Faulwasser, C. Jones, and D. Bonvin, "Real-time optimization of uncertain process systems via modifier adaptation and gaussian processes," *2018 European Control Conference*, pp. 465–470, 2018.
- [14] E. A. del Rio Chanona, J. E. Alves Graciano, E. Bradford, and B. Chachuat, "Modier-adaptation schemes employing gaussian processes and trust regions for real-time optimization," Florianópolis - SC, Brazil, pp. 52–57, Apr. 2019.
- [15] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.
- [16] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.
- [17] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.
- [18] J. Kocijan, *Modelling and Control of Dynamic Systems using Gaussian Process Models*. Springer, 2016.
- [19] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-theoretic regret bounds for gaussian process optimization in the bandit setting," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012.
- [20] A. Lederer, J. Umlauft, and S. Hirche, "Uniform error bounds for gaussian process regression with application to safe control," *arXiv e-prints*, p. arXiv:1906.01376, Jun 2019.
- [21] J. Dennis and R. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, 1996.
- [22] J. Thomann and G. Eichfelder, "A trust-region algorithm for heterogeneous multiobjective optimization," *SIAM J. Optimization*, vol. 29, no. 2, pp. 1017–1047, 2019.
- [23] B. Chachuat, B. Srinivasan, and D. Bonvin, "Adaptation strategies for real-time optimization," *Comput. Chem. Eng.*, vol. 33, no. 10, pp. 1557–1567, 2009.