2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Real-Time Optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes

Tafarel de Avila Ferreira[1], Harsh A. Shukla[1], Timm Faulwasser[2], Colin N. Jones[1] and Dominique Bonvin[1]

Abstract— **In the context of static real-time optimization, the use of measurements allows dealing with uncertainty in the form of plant-model mismatch and disturbances. Modifier adaptation (MA) is a measurement-based scheme that uses first-order corrections to the model cost and constraint functions so as to achieve plant optimality upon convergence. However, first-order corrections rely crucially on the estimation of plant gradients, which typically requires costly plant experiments.**

The present paper proposes to implement real-time optimization via MA but use recursive Gaussian processes to represent the plant-model mismatch and estimate the plant gradients. This way, one can (i) attenuate the effect of measurement noise, and (ii) avoid plant-gradient estimation by means finite-difference schemes and, often, additional plant experiments. We use steady-state optimization data to build Gaussian-process regression functions. The efficiency of the proposed scheme is illustrated via a constrained variant of the Williams-Otto reactor problem.

I. Introduction

In the light of rigorous economic and ecological requirements, real-time optimization (RTO) of process systems has received increasing scientific attention. The core objective of RTO is to ensure system operation, while meeting quality specifications and guaranteeing safe operation. Due to plant-model mismatch, purely model-based optimization is often unable to reach plant optimality. Even with accurate models, external disturbances may shift the plant optimum, which may result in infeasibility and/or suboptimality. Hence, in order to ensure optimal operation, RTO methods need to adapt the model-based optimization problem using process measurements.

The RTO method most commonly used in industry is the so-called two-step approach, which consists of repeated parameter estimation and optimization [3,16,4]. However, in the presence of structural plant-model mismatch, this approach tends not to converge to the plant optimum [20,13].

Modifier adaptation (MA) is an RTO method that uses measurements to correct the cost and constraint functions of the optimization problem [8,13]. The main advantage of MA is that, under suitable assumptions, it reaches optimality upon convergence. Its drawback is that it typically requires estimates of the plant gradients. Several MA precursors and variants are documented in the literature, including early works [19] and more recent results [1,9,13,14]. A detailed overview of the state of the art is given in [14].

While the basic MA scheme relies on first-order corrections, the use of second-order modifiers has also been proposed [6]. However, accurate Hessian estimation from noisy data is rather difficult in practice. Recently, it has been proposed to locally fit a quadratic function to plant data, with the plant gradients being obtained from this local fit [2,10]. However, the method depends heavily on the quality of the first-order modifiers, that is, it still requires gradient estimation.

In the machine learning community, Gaussian-process (GP) regression is a popular tool to estimate unknown functions [11,18]. A GP is a probabilistic, non-parametric modeling technique that can be interpreted as the extension of multivariate normal distribution to infinitely many random variables. The main strength of GP regression is its ability, using very few parameters, to capture complex unknown functions. Due to its simplicity and effectiveness, GP regression is gaining attention in the field of control, optimization and dynamical systems. For selected application examples, the reader is referred to [12] and references therein.

This paper proposes to use machine GP in the context of RTO. More specifically, process measurements are used to recursively estimate the plant-model mismatch via GP. Put differently, we propose a variant of MA, whereby the usual modifiers are replaced by high-order regression functions. Although there exists a considerable amount of recent literature on RTO and a tremendous number of papers on machine learning, this paper appears to be the first mention of using machine learning in RTO. The contributions of this paper include: (i) a simple way of introducing GP regression in the MA framework, and (ii) an illustration that high-order corrections can help reach plant optimality despite the presence of measurement noise.

The paper is organized as follows. The optimization problem, the steady-state MA formulation, and GP are presented in Section II. RTO using GP is presented in Section III. The implementation aspects are illustrated via a simulated Williams-Otto CSTR reactor in Section IV. Finally, conclusions are drawn in Section V.

Authors HAS and CNJ 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] Tafarel de Avila Ferreira, Harsh A. Shukla, Colin N. Jones, and Dominique Bonvin are with Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne Switzerland. {tafarel.deavilaferreira, harsh.shukla, colin.jones, dominque.bonvin}@epfl.ch

[2] Timm Faulwasser is with Institute for Applied Informatics, Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany. timm.faulwasser@kit.edu

II. Problem Formulation and Preliminaries

A. Problem Formulation

The optimization of plant steady state can be stated as the following NLP:

$$\min_{\mathbf{u}} \Phi_p(\mathbf{u}) := \phi\big(\mathbf{u}, \mathbf{y}_p(\mathbf{u})\big) \tag{1a}$$

subject to

$$\mathrm{G}_{p,i}(\mathbf{u}) := \mathrm{g}_i\big(\mathbf{u}, \mathbf{y}_p(\mathbf{u})\big) \leq 0 \quad i = 1, ..., n_g, \tag{1b}$$

$$\mathbf{u} \in \mathcal{U}. \tag{1c}$$

Here $\mathbf{u}^{n_u} \in \mathbb{R}$ are the decision (or input) variables; $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured output variables; ϕ: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost function to be minimized; g_i: $\mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 1, ..., n_{\mathrm{g}}$, is the set of process-dependent inequality constraint functions; and $\mathcal{U}$ is typically determined by lower and upper bounds on the input variables, $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{u}^{\mathrm{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathrm{U}}$. The subscript $(\cdot)_p$ indicates a quantity related to the plant.

Usually, since the steady-state input-output map of the plant, $\mathbf{u} \in \mathbb{R}^{n_u} \mapsto \mathbf{y}_p \in \mathbb{R}^{n_y}$, is not precisely known, one relies on an approximation given by the best available model. That is, instead of tackling Problem (1) directly, one solves the following model-based optimization problem:

$$\min_{\mathbf{u}} \Phi(\mathbf{u}) := \phi\big(\mathbf{u}, \mathbf{y}(\mathbf{u})\big) \tag{2a}$$

subject to

$$\mathrm{G}_i(\mathbf{u}) := \mathrm{g}_i\big(\mathbf{u}, \mathbf{y}(\mathbf{u})\big) \leq 0 \quad i = 1, ..., n_g, \tag{2b}$$

$$\mathbf{u} \in \mathcal{U}\,. \tag{2c}$$

Due to plant-model mismatch and disturbances, the solutions to Problems (1) and (2) are usually different. RTO aims at reaching plant optimality by iteratively updating the model using plant measurements.

B. Modifier Adaptation

MA uses first-order corrections to the cost and constraint functions in order to match the necessary conditions of optimality of the plant upon convergence [13]. Input-affine terms are added to the cost and constraint functions of Problem (2). The optimal inputs are computed by solving the following modified optimization problem [13]:

$$\mathbf{u}^{\star}_{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}} \;\; \Phi(\mathbf{u}) + (\boldsymbol{\lambda}^{\Phi}_k)^{\intercal}\mathbf{u} \tag{3a}$$

subject to

$$\mathrm{G}_i(\mathbf{u}) + \varepsilon_{i,k} + (\boldsymbol{\lambda}^{\mathrm{G}_i}_k)^{\intercal}(\mathbf{u} - \mathbf{u}_k) \leq 0 \qquad i = 1, ..., n_g, \tag{3b}$$

$$\mathbf{u} \in \mathcal{U}, \tag{3c}$$

with

$$\varepsilon_{i,k} := \mathrm{G}_{p,i}(\mathbf{u}_k) - \mathrm{G}_i(\mathbf{u}_k), \tag{3d}$$

$$(\boldsymbol{\lambda}^{\Phi}_k)^{\intercal} := \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k), \tag{3e}$$

$$(\boldsymbol{\lambda}^{\mathrm{G}_i}_k)^{\intercal} := \frac{\partial \mathrm{G}_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathrm{G}_i}{\partial \mathbf{u}}(\mathbf{u}_k). \tag{3f}$$

The RTO iteration is denoted by the subscript $(\cdot)_k$. The zeroth-order terms $\varepsilon_{i,k}$, $i = 1, ..., n_g$, represent the differences between the plant values and the predicted values of the constraints at $\mathbf{u}_k$, while the first-order modifiers $\boldsymbol{\lambda}^{\Phi}_k$ and $\boldsymbol{\lambda}^{\mathrm{G}_i}_k$ correspond to the differences between the plant gradients and the gradients predicted by the model at $\mathbf{u}_k$. The optimal input $\mathbf{u}^{\star}_{k+1}$ may be filtered to restrict the input update in the presence of noise, as proposed by [13]:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + K(\mathbf{u}^{\star}_{k+1} - \mathbf{u}_k), \tag{4}$$

where $K = \mathrm{diag}(k_1, ..., k_{n_u}) \in \mathbb{R}^{n_u}$, $k_i \in (0, 1]$, $i = 1, ..., n_u$.

The main advantage of modifier adaptation lies in its ability to reach a KKT point of Problem (1) upon convergence. However, the estimation the plant gradients $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k)$ and $\frac{\partial \mathrm{G}_{p,i}}{\partial \mathbf{u}}(\mathbf{u}_k)$ at each RTO iteration is quite challenging.

C. Gaussian Processes

We briefly introduce the concept of GP. For a more detailed introduction, we refer the reader to [18,17]. Unlike parametric methods that identify the parameters using training data during the learning phase and discard training data after learning, GP are based on kernel methods that use all the available data to describe the input-output data through mapping functions.

For example, one can be interested in approximating the unknown function $f : \mathbb{R}^n \to \mathbb{R}$ using a GP. In the RTO context, the function $f(\cdot)$ can represent the plant steady-state map, any of the constraints or the cost. In this study, given a set of plant inputs and outputs, we are interested in approximating the mismatch between the true plant map and its available model.

Consider the unknown function $f : \mathbb{R}^n \to \mathbb{R}$, $z = f(\mathbf{u}) + \nu$, where $\nu \sim \mathcal{N}\left(0, \sigma^2\right)$. Using n_p available input-output pairs, the input-output data generated by $f(.)$ are $\bar{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_p}] \in \mathbb{R}^{n \times n_p}$ and $\bar{\mathbf{z}} = [z_1, z_2, \ldots, z_{n_p}]^T \in \mathbb{R}^{n_p \times 1}$. We use GP regression to establish a relationship between $\bar{U}$ and $\bar{\mathbf{z}}$ and obtain a corresponding conditional distribution of the output z for a new query input point $\mathbf{u}$, that is,

$$z|\mathbf{u}, \bar{U}, \bar{\mathbf{z}} \sim \mathcal{N}\left(z_m, z_v\right), \tag{5}$$

where the mean and the variance of z are:

$$z_m = \mathbf{c}^T \left(\bar{C} + \sigma^2 I_{n_p}\right)^{-1} \bar{\mathbf{z}}, \tag{6}$$

$$z_v = \kappa(\mathbf{u}) - \mathbf{c}^T \left(\bar{C} + \sigma^2 I_{n_p}\right)^{-1} \mathbf{c}. \tag{7}$$

Here, $\bar{C} \in \mathbb{R}^{n_p \times n_p}$ is a covariance matrix with the elements $\bar{C}_{ij} = c(\mathbf{u}_i, \mathbf{u}_j)$, $\mathbf{c} = \left[c(\mathbf{u}, \mathbf{u}_1), c(\mathbf{u}, \mathbf{u}_2), \ldots, c(\mathbf{u}, \mathbf{u}_{n_p})\right]^T \in \mathbb{R}^{n_p \times 1}$, $\kappa(\mathbf{u}) = c(\mathbf{u}, \mathbf{u})$, where $c(\cdot, \cdot)$ is a covariance function labeled *kernel*. In this work, we use the auto relevance determination squared exponential covariance function defined as

$$c(\mathbf{u}_i, \mathbf{u}_j) = \sigma_f^2 \exp\left(-\frac{(\mathbf{u}_i - \mathbf{u}_j)^T \Lambda (\mathbf{u}_i - \mathbf{u}_j)}{2}\right), \tag{8}$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$, The parameters $\theta = [\sigma_f, \lambda_{1:n}] \in \mathbb{R}^{n+1}$ are the hyperparameters that need to be learned/estimated from the data $\{\bar{U}, \bar{\mathbf{z}}\}$ during the training phase. Since the covariance matrix $\bar{C}$ and the covariance vector $\mathbf{c}$ depend on the hyperparameters θ and the inputs $\bar{U}$, one can also write $\bar{C}$ as $\bar{C}(\theta, \bar{U})$. To this end, consider $M(\theta, \bar{U}) := \bar{C}(\theta, \bar{U}) + \sigma^2 I_{n_p}$ and the log-marginal likelihood

$$\mathcal{L}(\theta, \bar{U}, \bar{\mathbf{z}}) = -\frac{1}{2}\bar{\mathbf{z}}^T M(\theta, \bar{U})^{-1}\bar{\mathbf{z}} - \frac{1}{2}\log|M(\theta, \bar{U})| - \frac{n}{2}\log 2\pi.$$

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

$$\theta^* = \underset{\theta}{\text{argmax}}\ \mathcal{L}(\theta, \bar{U}, \bar{\mathbf{z}}). \tag{9}$$

The above maximization problem is nonlinear and nonconvex in nature, for which deterministic as well as stochastic solution methods can be used. We refer to Chapter 2.4.2 in [12] for details. In this work, we use the gpml toolbox [18] that implements the conjugate-gradient method with Wolfe-Powell stopping criterion. Despite no guarantee of reaching global optimality, GP regression can always explain the data [5]. We refer to Chapter 7 in [18] and Chapter 2 in [12] for details regarding the convergence properties.

In summary, a GP provides z that has a normal distribution for a query point $\mathbf{u}$. The mean and covariance function indicate how similar $\mathbf{u}$ is with the training data using the hyperparmeters θ^*. The GP can be used to compute the output distribution of z with more weight on the nearest inputs. This way, the predicted output z is influenced more by the nearby input-output pairs obtained from the training data set. This flexibility is the main advantage of GP compared to fixed structure input-output relationship based on parametric methods. Hence, GP models are able to capture complex nonlinear input-output relationships through the use of only a few parameters.

The main drawback of GP is that the computational complexity grows as a cubic function of the number of data points N, that is, the computational complexity is $\mathcal{O}(N^3)$. However, in the context of RTO of process systems, this is not a major challenge as most systems have a large settling time. On the positive side, GP offer an interesting trade-off between exploration and exploitation [18] and lead to good approximations of unknown functions, a very attractive feature in the context of RTO.

Subsequently, we will rely on recursively adapted GP. Hence, for the sake of compact notation, given some unknown function f, we write

$$z = (\mathcal{GP})^f\left(\mathbf{u}, \bar{U}, \bar{\mathbf{z}}\right)$$

to denote the output distribution z obtained for the input $\mathbf{u}$ using a GP regression of f based on solving Eq. (9) subject to the input-output data $\bar{U} \in \mathbb{R}^{n \times n_p}$ and $\bar{\mathbf{z}} \in \mathbb{R}^{n_p \times 1}$. Since the output data $\bar{\mathbf{z}}$ is an implicit function of the input data $\bar{U}$, that is $\bar{\mathbf{z}}(\bar{U})$, one can write $z = (\mathcal{GP})^f\left(\mathbf{u}, \bar{U}, \bar{\mathbf{z}}(\bar{U})\right)$. For simplicity of notation, we can drop the set of output data $\bar{\mathbf{z}}$ as argument of $(\mathcal{GP})^f$ and simply write:

$$z = (\mathcal{GP})^f\left(\mathbf{u}, \bar{U}\right). \tag{10}$$

While the evaluation $z = (\mathcal{GP})^f\left(\mathbf{u}, \bar{U}\right)$ is computationally cheap for fixed $\bar{U}$ and $\bar{\mathbf{z}}$, one has to repeatedly solve Problem (9) when the data $\bar{U}$ and $\bar{\mathbf{z}}$ change through acquisition of more data.

III. Using Gaussian Processes for RTO

The main limitation of standard MA stems from the need to estimate plant gradients. Typically, finite-difference approximations are used to compute plant gradients, that is, the gradients are estimated based on additional plant runs with imposed perturbations. Note that this becomes impractical when the input dimension is large. Moreover, standard MA uses only first-order corrections to update the cost and constraint functions. Here, we propose an MA scheme that uses GP to overcome these difficulties.

The main idea consists in replacing the zeroth- and first-order corrections used in Problem (3) by a description of the plant-model mismatch obtained from GP regression. To this end, and considering Eq. (10), we use

$$(\mathcal{GP})^f\left(\mathbf{u}, \bar{U}\right),\ \ f \in \{\Phi_p - \Phi, G_{p,1} - G_1, \ldots, G_{p,n_g} - G_{n_g}\}, \tag{11}$$

to denote the GP approximation to the plant-model mismatch of the cost and constraints based on the data $\bar{U}$ and $\bar{\mathbf{z}}$ (whereby the later argument is again dropped for simplicity). The superscript f serves to identify the unknown function.

A. Proposed GP-based MA Scheme

We suggest an RTO scheme based on solving the following NLP:

$$\mathbf{u}^\star_{k+1} = \underset{\mathbf{u}}{\text{argmin}}\ \ \Phi(\mathbf{u}) + (\mathcal{GP})_k^{(\Phi_p - \Phi)}(\mathbf{u}, \bar{U}_k) \tag{12a}$$

subject to

$$\mathrm{G}_i(\mathbf{u}) + (\mathcal{GP})_k^{(G_{p,i} - G_i)}(\mathbf{u}, \bar{U}_k) \leq 0, \quad i = 1, ..., n_g, \tag{12b}$$

$$\mathbf{u} \in \bar{U}, \tag{12c}$$

where the difference between the plant and the model of the cost and constraint functions are modeled by $(\mathcal{GP})_k^{(\Phi_p - \Phi)} \in \mathbb{R}$ and $(\mathcal{GP})_k^{(G_{p,i} - G_i)} \in \mathbb{R}$, $i = 1, ..., n_g$, respectively. The superscript $(\Phi_p - \Phi)$ indicates that the difference is between the plant and the model of the cost functions. Similarly, the superscript $(G_{p,i} - G_i)$ points to the difference between the plant and the model of the i^{th} constraint. $\bar{U}_k$ is the available input set at the k^{th} iteration.

We may also filter the optimal input $\mathbf{u}^\star_{k+1}$ as in Eq. (4). Fig. 1 depicts the MA scheme that uses GP to estimate the plant-model mismatch. It is important to note that, in contrast to standard MA that uses zeroth- and first-order correction terms to update the optimization problem, the optimization problem is modified here by adding GP regression functions to the cost and constraint functions such that the cost and constraint functions of the modified optimization Problem (12) locally match those of the plant. Furthermore, we use a smooth squared exponential kernel so that, if Problem (2)

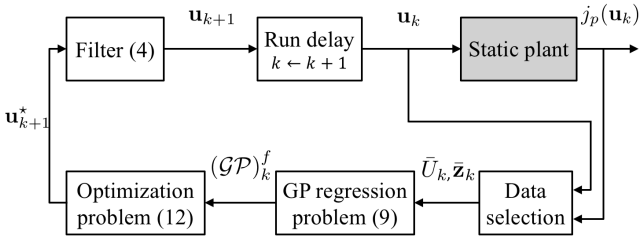

Fig. 1: MA at the k^{th} iteration using GP to estimate the plant-model mismatch, with $j \in \{\Phi, G_1, \ldots, G_{n_g}\}$ and $f \in \{\Phi_p - \Phi, G_{p,1} - G_1, \ldots, G_{p,n_g} - G_{n_g}\}$.

admits an optimal solution, so does Problem (12). Hence, upon convergence, if the GP regression functions locally approximate the plant-model mismatch well, Problem (12) will converge to a KKT point of the plant. In addition, in order to avoid overfitting, we take into account a finite number of points within a certain radius from the current operating point (see the Data selection block in Fig. 1). This way, the dimension of the covariance matrix used for building the GP functions does not grow with the size of the data and is comparatively very small as we shall see in the next section. Note that the marginal likelihood optimization problems (9) corresponding to the GP regressions for the $n_g + 1$ functions can be computed in parallel.

IV. Case Study: Williams-Otto Reactor

In this section, we apply both standard MA and GP-based MA to the Williams-Otto reactor [21]. This reactor is an ideal continuous stirred-tank reactor with the three reactions:

$$\mathrm{A} + \mathrm{B} \xrightarrow{k_1} \mathrm{C}, \tag{13}$$

$$\mathrm{C} + \mathrm{B} \xrightarrow{k_2} \mathrm{P} + \mathrm{E}, \tag{14}$$

$$\mathrm{P} + \mathrm{C} \xrightarrow{k_3} \mathrm{G}. \tag{15}$$

The reactants A and B are fed with the mass flowrates F_A and F_B, respectively. The desired products are P and E, whereas G is an undesired byproduct. The intermediate product C is also produced. Since it is assumed that the reaction scheme is not well understood, the following two reactions have been proposed for the model [7]:

$$\mathrm{A} + 2\mathrm{B} \xrightarrow{k_1^*} \mathrm{P} + \mathrm{E}, \tag{16}$$

$$\mathrm{A} + \mathrm{B} + \mathrm{P} \xrightarrow{k_2^*} \mathrm{P} + \mathrm{E}. \tag{17}$$

The material balance equations for the plant and the model are given in [22]. The objective consists in maximizing the steady-state profit, while considering constraints on the concentrations of the reactant A and of the byproduct G [15]. The optimization problem can be expressed mathematically as follows:

TABLE I: Species prices for different scenarios.

Prices	P_P	P_E	P_A	P_B
Scenario I	1043.38	20.92	79.23	118.34
Scenario II	1073.25	25.92	94.18	95

$$\begin{aligned} \max_{F_B, T_R} \quad & J = P_P X_P F + P_E X_E F - P_A F_A - P_B F_B + 200, \\ \text{s.t.} \quad & \mathrm{g}_1 = X_A - 0.12 \leq 0, \\ & \mathrm{g}_2 = X_G - 0.08 \leq 0, \\ & F_B \in [4, 7], \\ & T_R \in [70, 100], \end{aligned} \tag{18}$$

where F is sum of the reactant mass flowrates, $F = F_A + F_B$, X_i is the concentrations of Species i, F_B and T_R are the decision variables.The feed flowrate of Component A is kept constant at $F_A = 1.8275$ kg/s.

We solve the aforementioned optimization problem for the two price scenarios given in Table I. In Scenario I, both composition constraints are active at the plant optimum, whereas in Scenario II only the constraint on composition G is active at the plant optimum. We start with Scenario I and switch to Scenario II after 55 iterations. Two optimization schemes are compared :

- *Standard MA* as per Eq. (3). We consider 5 initial operating points that are used to estimate the initial values of the first-order modifiers via linear interpolation. The plant and model gradients are estimated via forward finite differences and used in Eqs. (3e) and (3f).
- *GP-based MA* as per Eq. (12). We consider the same initial operating points as with standard MA. These points are used to find the hyperparameters θ^* of the GP regression. At each RTO iteration, the newly available data are used to update the mean, the covariance, and the hyperparameters.

We assume that the plant measurements of the cost Φ_p and the concentrations X_A and X_G are subject to zero-mean noise with standard deviations σ_Φ = 0.5 and $\sigma_{X_A} = \sigma_{X_G}$ = 0.0005 as proposed by [13]. We choose the rather low filter gain of 0.4 for all diagonal matrix elements so as to easily enforce convergence. In GP-based MA, in order to avoid overfitting, we reject the data to compute the GP regression functions if more than 10 points lie in a radius of 1 kg/s for F_B and 10°C for T_R. Hence, the dimension of the computed covariance matrix ranges from 20×20 to 25×25.

Starting the RTO from the initial conservative feasible point $\mathbf{u}_0 = [6.9, 86]^\intercal$, simulations are performed for 120 iterations. Figs. 2 and 3 show the performance of standard MA, while Figs. 4 to 5 show that of GP-based MA. The first 5 points in light blue correspond to the 5 initial operating points. The remaining points in dark blue are the plant evaluations needed for both standard MA and GP-based MA. The dashed green line represents the plant optimal values for Scenarios I and II.

Upon comparing Figs. 2 to 5 for the same level of noise,

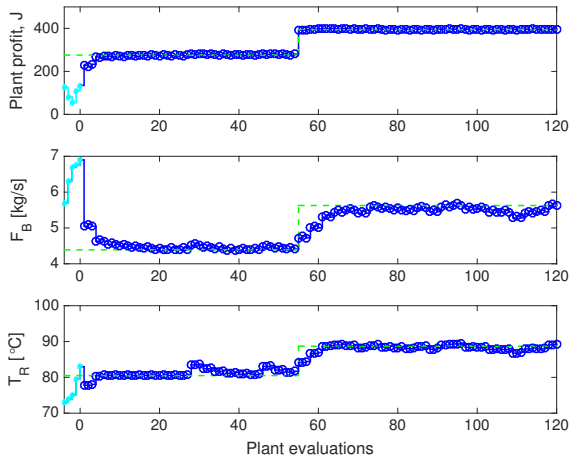

Fig. 2: Standard MA applied to the Williams-Otto reactor. Evolution of the plant cost and of the inputs F_B and T_R. Dashed green line: plant optimal values. Light blue line: initial operating points. Blue line: evolution of the cost and inputs.

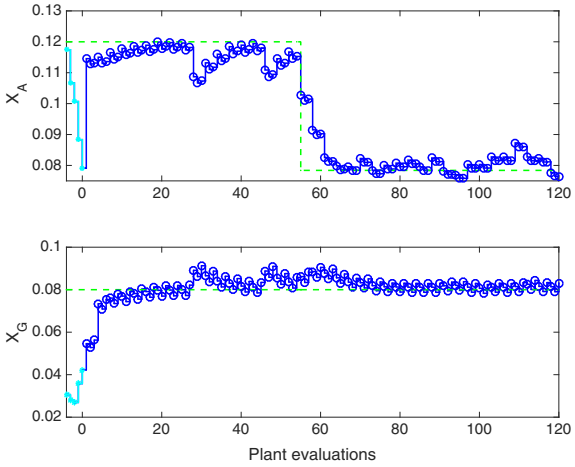

Fig. 3: Standard MA. Evolution of the constrained concentrations X_A and X_G for the plant. Dashed green line: plant optimal values. Light blue line: initial operating points. Blue line: evolution of the concentrations.

one can see that standard MA oscillates around the optimum, whereas GP-based MA converges to the plant optimum of Scenario I in 7 iterations, and in about 4 iterations from Scenario I to Scenario II. The lower plot of Fig. 3 shows that the constraint on concentration X_G is violated several times. The cost function and inputs are significantly less noisy for GP-based MA than for standard MA. This behaviour can be explained by the fact that GP-based MA deals with noisy measurements better than standard MA. In standard MA, the noise is handled by the choice of the step-length perturbation for the finite-difference approach, whereas in GP-based MA the noise is absorbed by the GP regression computed at each RTO iteration.

Fig. 6 compares the performance of standard MA and GP-based MA in the input space for both Scenarios I and II. The pink dot corresponds to the plant optimum for Scenario I, whereas the green dot indicates the plant optimum for Scenario II. The solid red lines represent the constraints X_A = 0.12 and X_G = 0.08. The dashed red lines are the contour lines of the plant profit for Scenario I, while the black lines represent the plant profit for Scenario II. Although it takes about 4 to 7 iterations to reach the plant optimum with standard MA for both scenarios, the constrained concentration X_G is constantly violated because of noisy measurements. In contrast, GP-based MA takes about 3 to 5 iterations to reach the plant optimum, with the constraints being hardly violated. Furthermore, the inputs are significantly less noisy with GP-based MA than with standard MA.

V. Conclusions

This paper has proposed a RTO scheme that combines MA and machine learning via GP. The approach, which estimates the plant-model mismatch using GP regression functions, has been illustrated by means of the real-time optimization of

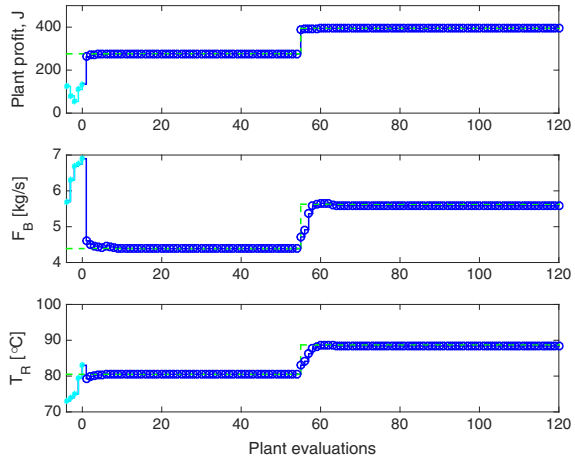

Fig. 4: GP-based MA applied to the Williams-Otto reactor. Evolution of the plant cost and of the inputs F_B and T_R. Dashed green line: plant optimal values. Light blue line: initial operating points. Blue line: evolution of the cost and inputs.

the Williams-Otto reactor. Simulations have shown that the proposed approach performs well despite the fair amount of noise added to the measurements. A comparison between standard MA and GP-based MA indicates that the latter clearly outperforms the former in terms of noise attenuation.

In future work, we will attempt to prove that the approach proposed in this paper indeed converges to plant optimality in the presence of noise.

References

[1] M. A. Brdyś and P. Tatjewski. *Iterative Algorithms for Multilayer Optimizing Control*. Imperial College Press, London UK, 2005.

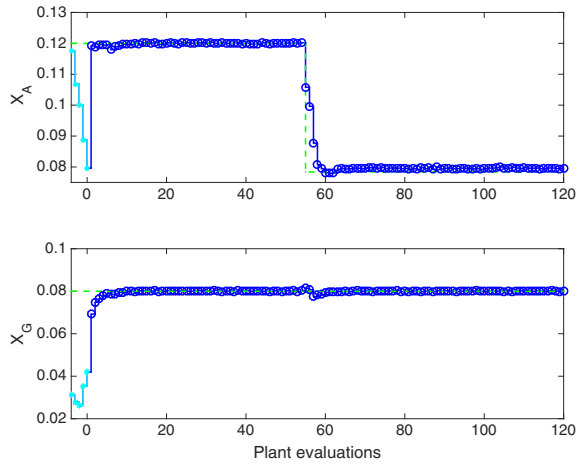

Fig. 5: GP-based MA. Evolution of the constrained concentrations X_A and X_G for the plant. Dashed green line: plant optimal values. Light blue line: initial operating points. Blue line: evolution of the concentrations.

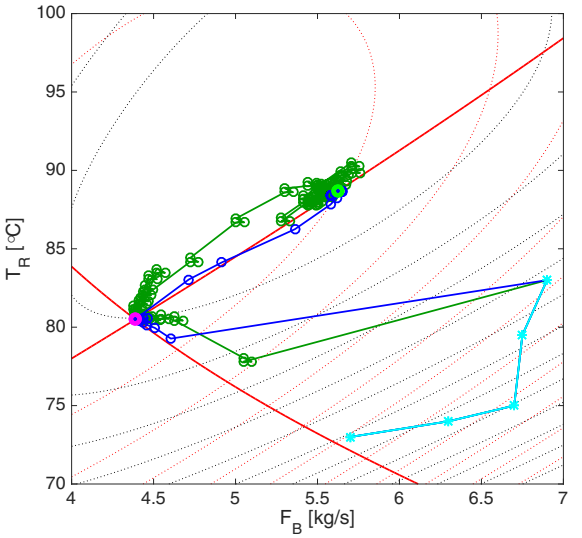

Fig. 6: Comparison of input trajectories with standard MA and GP-based MA. Pink dot: plant optimum of Scenario I. Light green dot: plant optimum of Scenario II. Green line: evolution of standard MA. Blue line: evolution of GP-based MA. Red line: constrained composition bounds. Light blue line: initial operating points.

[2] G. A. Bunin, G. François, and D. Bonvin. From discrete measurements to bounded gradient estimates: A look at some regularizing structures. *Industrial & Engineering Chemistry Research*, 52(35):12500–12513, 2013.

[3] C. Y. Chen and B. Joseph. On-line optimization using a two-phase approach: An application study. *Industrial & Engineering Chemistry Research*, 26(9), 1987.

[4] M. L. Darby, M. Nikolaou, J. Jones, and D. Nicholson. RTO: An overview and assessment of current practice. *Journal of Process Control*, 21(6):874–884, 2011.

[5] M. P. Deisenroth. *Efficient Reinforcement Learning Using Gaussian Processes*, volume 9. KIT Scientific Publishing, 2010.

[6] T. Faulwasser and D. Bonvin. On the use of second-order modifiers for real-time optimization. *IFAC Proceedings Volumes*, 47(3):7622–7628, 2014.

[7] J. F. Forbes, T. E. Marlin, and J. F. MacGregor. Model adequacy requirements for optimizing plant operations. *Computers & Chemical Engineering*, 18(6):497–510, 1994.

[8] W. Gao and S. Engell. Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering*, 29(6):1401–1409, 2005.

[9] W. Gao and S. Engell. Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering*, 29(6):1401–1409, 2005.

[10] W. Gao, R. Hernández, and S. Engell. A study of explorative moves during modifier adaptation with quadratic approximation. *Processes*, 4(4):45, 2016.

[11] D. R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 21(4):345–383, Dec 2001.

[12] J. Kocijan. *Modelling and Control of Dynamic Systems Using Gaussian Process Models*. Springer, 2016.

[13] A. G. Marchetti, B. Chachuat, and D. Bonvin. Modifier-adaptation methodology for real-time optimization. *Industrial & Engineering Chemistry Research*, 48:6022–6033, 2009.

[14] A. G. Marchetti, G. François, T. Faulwasser, and D. Bonvin. Modifier adaptation for real-time optimization: Methods and applications. *Processes*, 4(4):55, 2016.

[15] A. G. Marchetti, M. Singhal, T. Faulwasser, and D. Bonvin. Modifier adaptation with guaranteed feasibility in the presence of gradient uncertainty. *Computers & Chemical Engineering*, 98:61–69, 2017.

[16] T. E. Marlin and A. N. Hrymak. Real-time operations optimization of continuous processes. *AIChE Symposium Series*, 93(316):156–164, 1997.

[17] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[18] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*, volume 1. MIT press Cambridge, 2006.

[19] P. D. Roberts. An algorithm for steady-state system optimization and parameter estimation. *International Journal of Systems Science*, 10:719–734, 1979.

[20] P. Tatjewski. Iterative optimizing set-point control–the basic principle redesigned. *IFAC Proceedings Volumes*, 35(1):49–54, 2002.

[21] T. J. Williams and R. E. Otto. A generalized chemical processing model for the investigation of computer control. *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, 79:458–473, 1960.

[22] Y. Zhang and J. F. Forbes. Extended design cost, a performance criterion for real-time optimization systems. *Computers & Chemical Engineering*, 24(8):1829–1841, 2000.