Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 52-1 (2019) 52–57

Modifier-Adaptation Schemes Employing Gaussian Processes and Trust Regions for Real-Time Optimization ⋆

E.A. del Rio Chanona * **J.E. Alves Graciano** *,** **E. Bradford** *** **B. Chachuat** *

* *Centre for Process Systems Engineering (CPSE), Department of Chemical Engineering, Imperial College London, UK*
** *Universidade de São Paulo, Escola Politecnica, Departamento de Engenharia Qumica, São Paulo, Brazil*
*** *Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway*

AbstractThis paper investigates modifier-adaptation schemes based on Gaussian processes to handle plant-model mismatch in real-time optimization of uncertain processes. Building upon the recent work by Ferreira et al. [European Control Conference, 2018], we present two improved algorithms that rely on trust-region ideas in order to speed-up and robustify the approach. The first variant introduces a conventional trust region on the input variables, whose radius is adjusted based on the Gaussian process predictors' ability to capture the cost and constraint mismatch. The second variant exploits the variance estimates from the Gaussian processes to define multiple trust regions directly on the cost and constraint predictors. These algorithms are demonstrated and compared on a Williams-Otto reactor benchmark problem.

Keywords: real-time optimization; modifier adaptation; Gaussian process; trust region

1. INTRODUCTION

Real-time optimization (RTO) systems are well-accepted by industrial practitioners, with numerous successful applications reported over the last few decades (Darby et al., 2011). These systems rely on first-principles models, and in those processes where the optimization execution period is much longer than the closed-loop process dynamics, steady-state models are commonly employed to conduct the optimization (Marlin and Hrymak, 1997). Traditionally, the model is updated in real-time using the available measurements, before repeating the optimization. This two-step RTO paradigm is both intuitive and popular, yet it can hinder convergence to the actual plant optimum due to lack of integration between the model-update and optimization steps, especially in the presence of plant-model mismatch (Tatjewski, 2002; Chachuat et al., 2009). This has motivated the development of alternative adaptation paradigms in RTO, such as modifier adaptation (Marchetti et al., 2009).

Similar to two-step RTO, modifier adaptation (MA) embeds the available process model into a nonlinear optimization problem that is solved at each RTO execution. The key difference is that the process measurements are now used to update so-called modifiers that are added to the cost and constraint function in the optimization model. This methodology greatly alleviates the problem of offset from the actual plant optimum, by enforcing that the KKT conditions determined by the model match those of the plant upon convergence. However, this desirable property comes at the cost of having to estimate the cost and constraint gradients from process measurements.

Recent advances in MA schemes are reviewed in the survey paper by Marchetti et al. (2016). The nested MA scheme proposed by Navia et al. (2015) removes the need for estimating the plant gradients by embedding the modified optimization model into an outer problem that optimizes over the gradient modifiers using a derivative-free algorithm. Gao et al. (2015) combined MA with a quadratic surrogate trained with historical data in an algorithm called MAWQA. Likewise, Singhal et al. (2016) investigated data-driven approaches based on quadratic surrogates. More recently, Ferreira et al. (2018) considered the use of Gaussian processes (GP), trained from past measurement information, as the modifiers for the cost and constraint functions.

This paper further develops MA schemes based on GP. Two improved algorithms are presented, which rely on trust-region ideas in order to speed-up and robustify the original algorithm by Ferreira et al. (2018). The first variant introduces a conventional trust region on the input variables, whose radius is adjusted based on the accuracy of the GP predictors to capture the cost and constraint mismatch. The second variant exploits the variance estimates from the GPs to define trust regions directly on the cost and constraint predictors. The rest of the paper provides background on MA and GP in Sec. 2, describes the new trust-region MA-GP schemes in Sec. 3, and illustrates their performance in Sec. 4.

⋆ The first two authors contributed equally to the paper. Corresponding author: b.chachuat@imperial.ac.uk

10.1016/j.ifacol.2019.06.036

2. PRELIMINARIES

2.1 Modifier Adaptation

The problem of optimizing the steady-state performance of a given plant subject to operational or safety constraints can be formulated as:

$$\begin{aligned} \min_{\mathbf{u}\in\mathcal{U}} \; & G_0^{\mathrm{p}}(\mathbf{u}) := g_0\left(\mathbf{u}, \mathbf{y}^{\mathrm{p}}(\mathbf{u})\right) \\ \text{s.t. } & G_i^{\mathrm{p}}(\mathbf{u}) := g_i\left(\mathbf{u}, \mathbf{y}^{\mathrm{p}}(\mathbf{u})\right) \leq 0, \quad i = 1 \dots n_g \end{aligned} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ and $\mathbf{y}^{\mathrm{p}} \in \mathbb{R}^{n_y}$ are vectors of the plant input and output variables, respectively; $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 0..., n_g$, 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, $\mathbf{u}^{\mathrm{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathrm{U}}$. Notice the superscript $(\cdot)^{\mathrm{p}}$ used to indicate plant-related quantities.

The challenge is of course that an exact mapping $\mathbf{y}^{\mathrm{p}}(\cdot)$ is unknown in practice, and the output $\mathbf{y}^{\mathrm{p}}(\mathbf{u})$ can only be measured for a particular input value $\mathbf{u}$, in the manner of an oracle. However, provided that a parametric (approximate) model of the plant's input-output behavior is available, $\mathbf{y}(\mathbf{u}, \cdot)$, one may solve the following model-based optimization problem instead:

$$\begin{aligned} \min_{\mathbf{u}\in\mathcal{U}} \; & G_0(\mathbf{u}) := g_0\left(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\right) \\ \text{s.t. } & G_i(\mathbf{u}) := g_i\left(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\right) \leq 0, \quad i = 1 \dots n_g \end{aligned} \tag{2}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is a vector of adjustable model parameters.

In the presence of model mismatch and process disturbances, the optimal solution value of the problem (2) could greatly differ from that of (1). On-line optimization takes advantage of the available measurements in order to compensate for such uncertainty and adapt the model-based problem in order to get closer to the actual process optimum. In a modifier-adaptation scheme (Marchetti et al., 2009), the measurements are used to correct the cost and constraint function values at a given iterate $\mathbf{u}^k$, in order to determine the next iterate $\mathbf{u}^{k+1}$:

$$\begin{aligned} \mathbf{u}^{k+1} \in \arg\min_{\mathbf{u}\in\mathcal{U}} \; & G_0(\mathbf{u}) + (\boldsymbol{\lambda}_{G_0}^k)^{\mathsf{T}}\mathbf{u} \\ \text{s.t. } & G_i(\mathbf{u}) + \varepsilon_{G_i}^k + (\boldsymbol{\lambda}_{G_i}^k)^{\mathsf{T}}[\mathbf{u} - \mathbf{u}^k] \leq 0, \\ & \qquad\qquad i = 1 \dots n_g \end{aligned} \tag{3}$$

where $\varepsilon_{G_i}^k \in \mathbb{R}$ are zeroth-order modifiers for the constraints, and $\boldsymbol{\lambda}_{G_i}^k \in \mathbb{R}^{n_u}$ are first-order modifiers for the cost and constraints.

The use of modifiers is appealing in that a KKT point $\mathbf{u}^\infty$ for the corrected model-based problem (3) is also a KKT point for the original problem (1), provided that the modifiers satisfy (Marchetti et al., 2009):

$$\begin{aligned} \varepsilon_{G_i}^k &= G_i^{\mathrm{p}}(\mathbf{u}^\infty) - G_i(\mathbf{u}^\infty), \quad i = 1 \dots n_g \\ \boldsymbol{\lambda}_{G_i}^k &= \boldsymbol{\nabla} G_i^{\mathrm{p}}(\mathbf{u}^\infty) - \boldsymbol{\nabla} G_i(\mathbf{u}^\infty), \quad i = 0 \dots n_g \end{aligned}$$

A simple update rule for the modifiers that fulfills the foregoing conditions upon convergence is:

$$\begin{aligned} \varepsilon_{G_i}^{k+1} &= (1-\eta)\varepsilon_{G_i}^{k+1} + \eta\left[G_i^{\mathrm{p}}(\mathbf{u}^k) - G_i(\mathbf{u}^k)\right] \\ \boldsymbol{\lambda}_{G_i}^{k+1} &= (1-\eta)\boldsymbol{\lambda}_{G_i}^k + \eta\left[\boldsymbol{\nabla} G_i^{\mathrm{p}}(\mathbf{u}^k) - \boldsymbol{\nabla} G_i(\mathbf{u}^k)\right] \end{aligned}$$

where the tuning parameters $\eta \in (0, 1]$ may be reduced to help stabilize the iterations. The biggest burden with this approach is estimating the gradients $\boldsymbol{\nabla} G_i^{\mathrm{p}}(\mathbf{u}^k)$ of the cost and constraint functions at each RTO iteration. Finite difference methods and linear system identification methods may be applied, but they require perturbation and additional plant evaluations. Better approaches include the use of rank-one updates (Rodger and Chachuat, 2011), transient measurements (François and Bonvin, 2013), or directional derivatives (Singhal et al., 2016).

2.2 Gaussian Processes

GPs are an interpolation technique developed by Krige (1951) and popularized by the machine learning community (Rasmussen and Williams, 2016). GP regression aims to describe an unknown function $f : \mathbb{R}^{n_u} \to \mathbb{R}$ using noisy observations, $y = f(\mathbf{u}) + \nu$, where $\nu \sim \mathcal{N}(0, \sigma_\nu^2)$ is Gaussian distributed measurement noise with zero mean and (possibly unknown) variance σ_ν^2.

GPs consider a distribution over functions, and they can be seen as a generalization of multivariate Gaussian distributions,

$$f(\cdot) \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot))$$

where the mean function $m(\cdot)$ can be interpreted as the deterministic part of the function; and the covariance function $k(\cdot, \cdot)$ accounts for correlations between the function values at different points.

Our focus herein is on a constant mean function, $m(\mathbf{u}) := c$, alongside the squared-exponential (SE) covariance function (Rasmussen and Williams, 2016):

$$k(\mathbf{u}, \mathbf{u}') := \sigma_n^2 \exp\left(-\frac{1}{2}(\mathbf{u} - \mathbf{u}')^{\mathsf{T}}\boldsymbol{\Lambda}(\mathbf{u} - \mathbf{u}')\right)$$

where σ_n^2 is the covariance magnitude; and $\boldsymbol{\Lambda} := \mathrm{diag}(\lambda_1 \cdots \lambda_{n_u})$ is a scaling matrix. Underlying the choice of the SE covariance function is the assumption that the inferred function f is both smooth and stationary.

Maximum likelihood estimation is commonly applied to infer the unknown hyperparameters $\boldsymbol{\Psi} := [c \;\; \sigma_n \;\; \sigma_\nu \;\; \lambda_1 \;\; \dots \;\; \lambda_{n_u}]^{\mathsf{T}}$, including σ_ν in case the measurement noise variance is also unknown. Consider N (noisy) function evaluations, denoted by $\mathbf{y} := [y_1 \;\; \cdots \;\; y_N]^{\mathsf{T}} \in \mathbb{R}^N$, with corresponding inputs collected in the matrix $\mathbf{U} := [\mathbf{u}_1 \;\; \cdots \;\; \mathbf{u}_N] \in \mathbb{R}^{n_u \times N}$. The log-likelihood of the observed data, ignoring constant terms, is given by:

$$\mathcal{L}(\boldsymbol{\Psi}) := -\frac{1}{2}\ln(|\mathbf{K}(\mathbf{U})|) - \frac{1}{2}(\mathbf{y} - \mathbf{1}c)^{\mathsf{T}}\mathbf{K}(\mathbf{U})^{-1}(\mathbf{y} - \mathbf{1}c)$$

with $K_{ij}(\mathbf{U}) := k(\mathbf{u}_i, \mathbf{u}_j) + \sigma_\nu^2\delta_{ij}$ for each pair $(i, j) \in \{1 \dots N\}^2$; and the Kronecker delta function δ_{ij}.

The predicted distribution of $f(\mathbf{u})$ at an arbitrary input point $\mathbf{u}$, given the input-output data $(\mathbf{U}, \mathbf{y})$ and the maximum-likelihood estimates of $\boldsymbol{\Psi}$, follows the Gaussian distribution

$$f(\mathbf{u})|\mathbf{U}, \mathbf{y} \sim \mathcal{N}(\mu_f(\mathbf{u}), \sigma_f^2(\mathbf{u}))$$

with

$$\begin{aligned} \mu_f(\mathbf{u}) &:= \mathbf{r}(\mathbf{u}, \mathbf{U})\,\mathbf{K}(\mathbf{U})^{-1}\,\mathbf{y} + c \\ \sigma_f^2(\mathbf{u}) &:= \sigma_n^2 - \mathbf{r}(\mathbf{u}, \mathbf{U})\,\mathbf{K}(\mathbf{U})^{-1}\,\mathbf{r}(\mathbf{u}, \mathbf{U})^{\mathsf{T}} \end{aligned}$$

and $\mathbf{r}(\mathbf{u}, \mathbf{U}) := [k(\mathbf{u}, \mathbf{u}_1) \cdots k(\mathbf{u}, \mathbf{u}_N)]$. The mean μ_f in this context is the prediction made by the GP at $\mathbf{u}$, while the

variance σ_f^2 provides a measure of the uncertainty around this predictor.

3. METHODOLOGY

3.1 Basic MA scheme with GP modifiers (MA-GP)

The use of GPs to describe the plant-model mismatch in an RTO context was first proposed by Ferreira et al. (2018). The main idea entails the training and use of GPs in order to correct the cost and each constraint separately,

$$G_i^{\mathrm{p}} - G_i \sim \mathcal{GP}\left(\mu_{\delta G_i}, \sigma^2_{\delta G_i}\right), \quad i = 0 \dots n_g$$

The resulting modified optimization problem that is solved at each RTO iteration is given by

$$\begin{aligned} \mathbf{u}^{k+1} \in \arg\min_{\mathbf{u}\in\mathcal{U}} \; & [G_0 + \mu^k_{\delta G_0}](\mathbf{u}) \\ \text{s.t. } & [G_i + \mu^k_{\delta G_i}](\mathbf{u}) \le 0, \quad i = 1 \dots n_g \end{aligned} \tag{4}$$

where $\mu^k_{\delta G_i}$ denotes the predictor of the GP trained with the input-output data set $(\mathbf{U}^k, \boldsymbol{\delta}\mathbf{G}_i^k)$; and $\boldsymbol{\delta}\mathbf{G}_i^k$ comprises measurements of the mismatch $\delta G_i(\cdot) := G_i^{\mathrm{p}}(\cdot) - G_i(\cdot)$ at input points in the matrix $\mathbf{U}^k$.

Algorithm 1 Basic MA-GP scheme

Input: GP predictors $\mu^0_{\delta G_i}$, $i = 0 \dots n_g$, trained with the data sets $(\mathbf{U}^0, \boldsymbol{\delta}\mathbf{G}_i^0)$

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

(1) Solve the modified optimization problem (4)
(2) Filter the new operating point $\mathbf{u}^{k+1}$
(3) Obtain measurements of the cost and constraint functions $G_i^{\mathrm{p}}(\mathbf{u}^{k+1})$, $i = 0 \dots n_g$
(4) Update the data sets $(\mathbf{U}^{k+1}, \boldsymbol{\delta}\mathbf{G}_i^{k+1})$, $i = 0 \dots n_g$ with the measurements at $\mathbf{u}^{k+1}$, and the corresponding GP predictors $\mu^{k+1}_{\delta G_i}$

A statement of the basic MA-GP scheme is provided in Algorithm 1. A set of GPs are trained for the cost and constraint mismatch as part of the initialization phase, and there is of course considerable freedom regarding the choice of this initial training set. For instance, an initial sample could be obtained in a given control subdomain using Sobol sampling.

The training set is updated in Step 4 as the iterations progress. In order to prevent overfitting and numerical difficulties in constructing the GPs and to be able to track a moving plant optimum, Ferreira et al. (2018) keep a limited number of historical records in the input-output data set; for instance, the N nearest-neighbors to the new point $\mathbf{u}^{k+1}$ obtained in Step 1. Moreover, the current iterate $\mathbf{u}^{k+1}$ need not be included in $\mathbf{U}^{k+1}$ should it be within a given radius of an existing point in $\mathbf{U}^k$, or $\mathbf{u}^{k+1}$ could be substituted for an existing nearby point in $\mathbf{U}^{k+1}$ instead. In any case, the predictors of the cost and constraint GP need recomputing after updating $\mathbf{U}^{k+1}$, thereby reducing the plant-model mismatch.

Similar to the classical MA approach (Sec. 2.1), the solution point $\mathbf{u}^{k+1}$ of the modified problem (4) could be filtered in order to reduce the step-size and help stabilize the MA-GP scheme. One simple filter consists in taking $(1-\eta)\mathbf{u}^k + \eta\mathbf{u}^{k+1}$ instead of $\mathbf{u}^{k+1}$ as the next iterate, with $\eta \in (0, 1]$. Claimed advantages of MA-GP over the classical MA with linear correction terms are a greater simplicity since estimates of the gradients $\boldsymbol{\nabla} G_i^{\mathrm{p}}(\mathbf{u}^k)$ are no longer required at each RTO iteration, a lower number of plant evaluations to reach a plant optimum, and a superior ability to handle measurement noise. However, tuning such schemes to achieve convergence may not be straightforward in the first place.

3.2 MA-GP scheme with input trust region

Inspired by globalization techniques based on trust-region ideas developed in derivative-free optimization (Conn et al., 2000) and surrogate-based optimization (Eason and Biegler, 2016), we propose a simple extension of the basic GP-MA scheme in Sec. 3.1 that relies on an adaptive trust region in input space. The optimization problem (4) is modified as follows:

$$\begin{aligned} \mathbf{d}^{k+1} \in \arg\min_{\mathbf{d}} \; & [G_0 + \mu^k_{\delta G_0}](\mathbf{u}^k + \mathbf{d}) \\ \text{s.t. } & [G_i + \mu^k_{\delta G_i}](\mathbf{u}^k + \mathbf{d}) \le 0, \quad i = 1 \dots n_g \\ & \|\mathbf{d}\| \le \Delta^k, \quad \mathbf{u}^k + \mathbf{d} \in \mathcal{U} \end{aligned} \tag{5}$$

where $\Delta^k \ge 0$ denotes the radius of a trust region in input space; and $\mathbf{d}^{k+1} \in \mathbb{R}^{n_u}$ is the predicted step.

Algorithm 2 MA-GP scheme with input trust region

Input: GP predictors $\mu^0_{\delta G_i}$, $i = 0 \dots n_g$, trained with the data sets $(\mathbf{U}^0, \boldsymbol{\delta}\mathbf{G}_i^0)$; initial operating point $\mathbf{u}^0 \in \mathcal{U}$; initial and maximal trust region radii $0 < \Delta^0 \le \overline{\Delta}$; trust-region parameters $0 < \eta_1 \le \eta_2 < \eta_3 \le 1$ and $0 < t_1 < 1 < t_2$

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

(1) Solve the modified optimization problem (5)
(2) Obtain measurements of the cost and constraint functions $G_i^{\mathrm{p}}(\mathbf{u}^k + \mathbf{d}^{k+1})$, $i = 0 \dots n_g$
(3) Compute the ratio ρ^{k+1} as per Eq. (6)
(4) Update the trust region radius
  (a) If $G_i^{\mathrm{p}}\left(\mathbf{u}^k + \mathbf{d}^{k+1}\right) > 0$ for some $i = 1 \dots n_g$ or $\rho^{k+1} < \eta_2$:
    $\Delta^{k+1} := t_1 \Delta^k$
  (b) Else if $\rho^{k+1} > \eta_3$ and $\|\mathbf{d}^{k+1}\| = \Delta^k$:
    $\Delta^{k+1} := \min\{t_2 \Delta^k, \overline{\Delta}\}$
  (c) Else:
    $\Delta^{k+1} := \Delta^k$
(5) Update the operating point
  (a) If $G_i^{\mathrm{p}}\left(\mathbf{u}^k + \mathbf{d}^{k+1}\right) > 0$ for some $i = 1 \dots n_g$ or $\rho^{k+1} < \eta_1$:
    $\mathbf{u}^{k+1} := \mathbf{u}^k$
  (b) Else:
    $\mathbf{u}^{k+1} := \mathbf{u}^k + \mathbf{d}^{k+1}$
(6) Update the data sets $(\mathbf{U}^{k+1}, \boldsymbol{\delta}\mathbf{G}_i^{k+1})$, $i = 0 \dots n_g$ with the measurements at $\mathbf{u}^k + \mathbf{d}^{k+1}$, and the corresponding GP predictors $\mu^{k+1}_{\delta G_i}$

A statement of the MA-GP scheme with input trust region (MA-GP-ITR) is provided in Algorithm 2. The initialization of the cost and constraint GPs, as well as the data set and GP updates in step 6, are identical to the basic MA-GP scheme in Sec. 3.1. Notice that there is also considerable flexibility in the choice of the trust region parameters η_1, η_2, η_3, t_1, t_2. A common setting in trust-region methods, which is also the setting used throughout

the case study in Sec. 4, is $\eta_1 = \eta_2 = 0.1$, $\eta_3 = 0.9$, $t_1 = 0.5$ and $t_2 = 2$.

The filtering step 2 in Algorithm 1 is replaced with the update steps 4 and 5 in Algorithm 2. The decisions about moving the trust-region center (current operating point) or changing its radius are based upon:

(i) the ratio of actual cost reduction to predicted cost reduction,

$$\rho^{k+1} := \frac{G_0^{\mathrm{p}}\left(\mathbf{u}^k\right) - G_0^{\mathrm{p}}\left(\mathbf{u}^k + \mathbf{d}^{k+1}\right)}{[G_0 + \mu^k_{\delta G_0}](\mathbf{u}^k) - [G_0 + \mu^k_{\delta G_0}](\mathbf{u}^k + \mathbf{d}^{k+1})} \tag{6}$$

(ii) the violation of any inequality constraint $i = 1 \dots n_g$,

$$G_i^{\mathrm{p}}\left(\mathbf{u}^k + \mathbf{d}^{k+1}\right) > 0$$

The trust-region radius Δ^{k+1} is reduced whenever the accuracy ratio ρ^{k+1} is too small or a plant constraint is violated after implementing the predicted step $\mathbf{d}^{k+1}$. Conversely, Δ^{k+1} is increased if the optimization problem (5) takes a full step and the modified cost provides a good enough prediction of the plant cost variation around this point. Otherwise, the trust-region radius stays unchanged. As for the operating point update, the full step $\mathbf{d}^{k+1}$ is accepted when no constraint violation is triggered and the accuracy ratio ρ^{k+1} is large enough. Otherwise, the operating point remains unchanged, which would entail a back-tracking in a practical RTO setup. Notice that such a back-tracking strategy assumes that the initial point $\mathbf{u}^0$ satisfies all the plant constraints. Determination of a feasible point $\mathbf{u}^0$ could be via the solution of an auxiliary feasibility problem prior to running Algorithm 2—see, e.g. (Bajaj et al., 2018).

3.3 MA-GP scheme with multiple predictor trust regions

The basic MA-GP scheme and the MA-GP scheme with ITR detailed in Sec. 3.1 & 3.2 take advantage of the GP predictors $\mu_{\delta G_i}$ in correcting the cost and constraint functions, but neither of them exploit the associated variance terms $\sigma^2_{\delta G_i}$. We now present another extension of Algorithm 1 that relies on multiple adaptive trust regions in the predictor space. The optimization problem (4) is modified as follows:

$$\begin{aligned} \mathbf{u}^{k+1} \in \arg\min_{\mathbf{u}\in\mathcal{U}} \; & [G_0 + \mu^k_{\delta G_0}](\mathbf{u}) \\ \text{s.t. } & [G_i + \mu^k_{\delta G_i}](\mathbf{u}) \le 0, \quad i = 1 \dots n_g \\ & \sigma^k_{\delta G_i}(\mathbf{u}) \le \Delta^k_i, \quad i = 0 \dots n_g \end{aligned} \tag{7}$$

where $\sigma^k_{\delta G_i}$ denotes the standard-deviation associated with the predictor $\mu^k_{\delta G_i}$ in the GP trained with the input-output data set $(\mathbf{U}^k, \boldsymbol{\delta}\mathbf{G}_i^k)$; and $\Delta^k_i \ge 0$ denote separate trust-region radii for the predictions made by the cost and constraint GPs.

A statement of the MA-GP scheme with multiple predictor trust regions (MA-GP-MPTR) is provided in Algorithm 3. The initialization of the cost and constraint GPs, as well as the data set and GP updates in step 6, are here again identical to the basic MA-GP scheme in Sec. 3.1. The trust region parameters η_1, η_2, η_3, t_1, t_2 are moreover the same as in Algorithm 2.

Algorithm 3 MA-GP scheme with multiple predictor trust regions

Input: GP predictors $\mu^0_{\delta G_i}$ and error functions $\sigma^0_{\delta G_i}$, $i = 0 \dots n_g$, trained with the data sets $(\mathbf{U}^0, \boldsymbol{\delta}\mathbf{G}_i^0)$; initial and maximal trust region radii $0 < \Delta_i^0 \le \overline{\Delta}_i$, $i = 0 \dots n_g$; trust-region parameters $0 < \eta_1 \le \eta_2 < \eta_3 \le 1$ and $0 < t_1 < 1 < t_2$

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

(1) Solve the modified optimization problem (7)
(2) Obtain measurements of the cost and constraint functions $G_i^{\mathrm{p}}(\mathbf{u}^{k+1})$, $i = 0 \dots n_g$
(3) Compute the ratio ρ^{k+1} as per Eq. (6)
(4) Update the predictor trust region of the cost
  (a) If $\rho^{k+1} < \eta_2$:
    $\Delta_0^{k+1} := t_1 \Delta_0^k$
  (b) Else if $\rho^{k+1} > \eta_3$ and $\sigma^k_{\delta G_0}(\mathbf{u}) = \Delta_0^k$:
    $\Delta^{k+1} := \min\{t_2 \Delta^k, \overline{\Delta}\}$
  (c) Else:
    $\Delta_0^{k+1} := \Delta_0^k$
(5) Update the predictor trust region of each constraint $i = 1 \dots n_g$
  (a) If $G_i^{\mathrm{p}}(\mathbf{u}^{k+1}) > 0$:
    $\Delta_i^{k+1} := t_1 \Delta_i^k$
  (b) Else if $G_i^{\mathrm{p}}(\mathbf{u}^{k+1}) < 0$ and $\sigma^k_{\delta G_i}(\mathbf{u}) = \Delta_i^k$:
    $\Delta_i^{k+1} := \min\{t_2 \Delta_i^k, \overline{\Delta}_i\}$
  (c) Else:
    $\Delta_i^{k+1} := \Delta_i^k$
(6) Update the data sets $(\mathbf{U}^{k+1}, \boldsymbol{\delta}\mathbf{G}_i^{k+1})$, $i = 0 \dots n_g$ with the measurements at $\mathbf{u}^{k+1}$, and the corresponding GP predictors $\mu^{k+1}_{\delta G_i}$

The decisions about changing the radii of the predictor trust regions are different for the cost and the constraints. Adaptation of the trust region for the cost predictor is based on the same ratio of actual cost reduction to predicted cost reduction as in Eq. (6),

$$\rho^{k+1} := \frac{G_0^{\mathrm{p}}\left(\mathbf{u}^k\right) - G_0^{\mathrm{p}}\left(\mathbf{u}^{k+1}\right)}{[G_0 + \mu^k_{\delta G_0}](\mathbf{u}^k) - [G_0 + \mu^k_{\delta G_0}](\mathbf{u}^{k+1})}$$

Essentially, the trust region radius Δ_0^{k+1} is reduced when the accuracy of the cost predictor is insufficient, increased when the accuracy is high and the cost trust-region constraint is active, and kept the same otherwise.

By contrast, adaptation of the trust region for a constraint predictor is directly based on the violation of the corresponding plant constraint. The trust-region radius Δ_i^{k+1} is reduced when the ith plant constraint is violated, increased when that constraint is inactive and the corresponding trust-region constraint is active, and kept the same otherwise. Notice also that back-tracking is irrelevant here in the presence of constraint violation, since a predictor trust region is not centered around a particular operating point $\mathbf{u}^{k+1}$.

The consideration of multiple trust regions on the cost and constraint predictor errors presents several advantages. Distinguishing between the cost and constraints could make the approach more resilient to poor scaling in larger problems, instead of having a joint trust region for all the inputs. Imposing a trust region on the predictor error is

also more versatile, as it could account for measurements scattered across the entire feasible region in order to overcome convergence to local optima. The price to pay for this globalization however, is that the predictor trust region constraints introduce extra nonconvexity in the modified optimization subproblems.

4. CASE STUDY

We consider a Williams-Otto benchmark problem to demonstrate and compare the various MA-GP schemes presented in Sec. 3. A continuous stirred-tank reactor (CSTR) is fed with two streams of pure components A and B, with respective mass flowrates F_{A} and F_{B}. The reactor operates at steady state and under the temperature T_{r}. The chemical reactions between these reagents produce two main products P and E, through a series of chemical reactions that also produce an intermediate C and a byproduct G:

$$\begin{aligned} \mathsf{A}+\mathsf{B} &\longrightarrow \mathsf{C} \\ \mathsf{B}+\mathsf{C} &\longrightarrow \mathsf{P}+\mathsf{E} \\ \mathsf{C}+\mathsf{P} &\longrightarrow \mathsf{G} \end{aligned}$$

Structural plant-model mismatch is introduced in the problem by assuming that the approximate model only knows about the following (incorrect) two reactions, which do not consider the intermediate C:

$$\begin{aligned} \mathsf{A}+2\mathsf{B} &\longrightarrow \mathsf{P}+\mathsf{E} \\ \mathsf{A}+\mathsf{B}+\mathsf{P} &\longrightarrow \mathsf{G} \end{aligned}$$

The complete set of mass-balance equations and kinetic rate equations for both reaction systems are reported, e.g., by Mendoza et al. (2016).

The optimization problem consists in maximizing the economic profit by manipulating the feedrate F_{B} and the reactor temperature T_{r}, subject to operating constraints on the residual mass fractions of A and G at the reactor outlet:

$$\begin{aligned} \min_{F_{\mathsf{B}},T_{\mathrm{r}}} \quad & G_0 := (1043.38X_{\mathsf{P}} + 20.92X_{\mathsf{E}})\,(F_{\mathsf{A}}+F_{\mathsf{B}}) \\ & \qquad - 79.23F_{\mathsf{A}} - 118.34F_{\mathsf{B}} \qquad (8) \\ \text{s.t.} \quad & \text{CSTR model (Mendoza et al., 2016)} \\ & G_1 := X_{\mathsf{A}} - 0.12 \le 0 \\ & G_2 := X_{\mathsf{G}} - 0.08 \le 0 \\ & F_{\mathsf{B}} \in [4,7], \quad T_{\mathrm{r}} \in [70,100] \end{aligned}$$

where X_i denotes the mass fraction of species i. We assume throughout that (indirect) measurements of the cost G_0 and inequality constraints G_1, G_2 are available, which are corrupted by Gaussian noise with zero mean and standard deviation $\sigma_{G_0} = 0.5$, $\sigma_{G_1} = \sigma_{G_2} = 0.0005$.

A graphical depiction of the optimization problem (8) is presented in Fig. 1. This plot shows the contour lines of the plant cost (multicolor thin lines), as well as the plant constraint limits (blue lines). Notice the gap between the unconstrained plant and model-based unconstrained optima (blue and red pentagrams), as an illustration of the plant-model mismatch in this problem.

Also shown in Fig. 1 is a comparison between the paths followed by the three MA-GP schemes in Algorithms 1, 2 and 3 (black, red and magenta lines). We implemented these algorithms in MATLAB, using the (local) NLP solver fmincon as part of the Optimization Toolbox for solving the

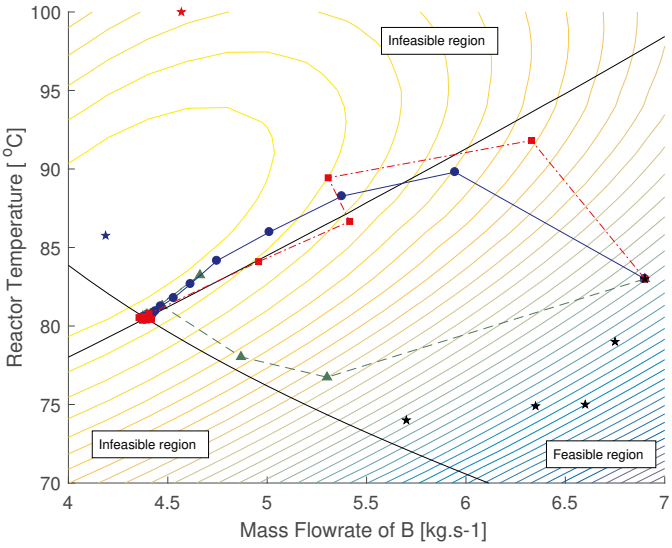

Figure 1. Graphical representation of the optimization problem (8) and comparison of the paths followed by the MA-GP schemes in Algorithms 1, 2 and 3. **Blue pentagram**: unconstrained plant optimum. **Red pentagram**: unconstrained model-based optimum. **Black pentagrams**: sample points used to initialize the cost and constraint GPs. **Blue lines**: plant constraints. **Black line**: iterates of MA-GP scheme (Algorithm 1). **Red line**: iterates of MA-GP-ITR scheme (Algorithm 2). **Magenta line**: iterates of MA-GP-MPTR scheme (Algorithm 3).

optimization subproblems, and the Statistics and Machine Learning Toolbox for training and prediction of the GPs describing the cost and constraint mismatch. Following Ferreira et al. (2018), we used 5 sampling points (black pentagrams) to initialize the cost and constraint GPs, and the last 10 measurement points (at most) in the data sets $\mathbf{U}^k$ for rebuilding these GPs at each RTO iteration. The initial operating point $\mathbf{u}^0$ corresponds to the sample point farthest to the right.

All three MA-GP schemes converge to the constrained plant optimum, where both inequality constraints are active, despite the presence of plant-model mismatch. The iterates cluster around the plant optimum upon convergence, showing that the GPs are indeed effective at describing the mismatch in the presence of measurement noise.

The iterations of the basic MA-GP scheme (Algorithm 1) – here with a filter parameter value of $\eta = 0.4$ – follow an infeasible path after the second iterate and up until they reach the plant optimum. Likewise, the second iterate of the MA-GP-ITR scheme (Algorithm 2) violates the plant constraints, but then the back-tracking implemented in Step 5a drives the iterates back to the feasible region after updating the constraint GPs and adapting the trust-region size accordingly. This constraint violation could be circumvented by reducing the size of the initial trust region Δ^0, in order to prevent the cost and constraint GPs from extrapolating too far from their respective training sets. Notice also that the MA-GP-ITR scheme requires fewer steps than the basic MA-GP scheme to reach the plant optimum in this case, which is attributed to the use of a

rather conservative filter value in the latter. It is indeed one of the main advantages of the proposed MA-GP-ITR scheme that an adaptive trust region is employed instead of setting an a priori value for the filter parameter η, resulting in RTO schemes that are both faster and easier to tune.

By contrast, the MA-GP-MPTR scheme (Algorithm 3) follows a very different path to the plant optimum. Imposing a trust-region on the cost and constraint predictors, rather than on the inputs, constrains the iterate $\mathbf{u}^{k+1}$ to remain close to points in the training data set $\mathbf{U}^k$. This explains why the first MA-GP-MPTR step deviates significantly from the direction of the unconstrained model-based optimum. The rest of the path to the plant optimum is fast, albeit taking smaller steps as limited by the trust regions of both the cost and constraint GPs. Here, trust-region constraints are active during the first 3 iterations only, corresponding to the GPs of G_1 and G_2 in the 1st iteration, to the GP of G_0 in the 2nd iteration, and to the GP of G_2 in the 3rd iteration. Overall, the main benefit of defining a trust region on the GP predictors' accuracy is making sure that extrapolation of the cost and constraint GPs away from their training remains under control, thereby improving the reliability of the RTO scheme.

5. CONCLUSIONS AND FUTURE DIRECTIONS

Building upon recent work by Ferreira et al. (2018), this paper has presented two improved RTO schemes combining modifier adaptation and GPs for handling structural plant-model mismatch, yet without the need for computing plant gradients explicitly. The first scheme (MA-GP-ITR) entails the use of similar trust-region ideas as in derivative-free or surrogate-based optimization, instead of a simple filtering of the inputs, in order to speed-up the iterations and facilitate tuning. The second scheme (MA-GP-MPTR) takes advantage of the error estimates in the GPs to define multiple trust regions directly on the GP predictors, thereby constraining the extrapolative capability of the GPs and improving reliability. The counterpart of this globalization however, is the need to handle extra nonconvexity introduced by such predictor trust regions in MA-GP-MPTR, whereas MA-GP-ITR features convex trust regions. A comparison between these two schemes and the original MA-GP scheme by Ferreira et al. (2018) has been carried out on a simple benchmark problem, showing a better performance in terms of convergence speed and mitigating plant constraint violations.

As part of future work, we will investigate the application of both improved MA-GP schemes to larger-scale RTO problems. Building on existing work connecting the modifier-adaptation and trust-region frameworks (e.g., Bunin, 2014), we shall moreover investigate the convergence properties of both schemes, including their ability to guarantee (local) optimality upon convergence and their global convergence properties.

ACKNOWLEDGEMENTS

This paper is based upon work supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grants EP/N010531/1 and EP/P016650/1. Financial support from Shell and FAPESP through the "Research Centre for Gas Innovation - RCGI" (FAPESP Proc. 2014/50279-4), ANP (Brazil National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation, and CNPq Brasil (Conselho Nacional de Desenvolvimento Cientfico e Tecnolgico - Proc. 200470/2017-5) is gratefully acknowledged. This project has also received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675215.

REFERENCES

Bajaj, I., Iyer, S.S., and Hasan, M.F. (2018). A trust region-based two phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. *Computers & Chemical Engineering*, in press.

Bunin, G.A. (2014). On the equivalence between the modifier-adaptation and trust-region frameworks. *Computers & Chemical Engineering*, 71, 154–157.

Chachuat, B., Srinivasan, B., and Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, 33(10), 1557–1567.

Conn, A.R., Gould, N.I.M., and Toint, P.L. (2000). *Trust-Region Methods*. MPS-SIAM Series on Optimization.

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

Eason, J.P. and Biegler, L.T. (2016). A trust region filter method for glass box/black box optimization. *AIChE Journal*, 62(9), 3124–3136.

Ferreira, T.A., Shukla, H.A., Faulwasser, T., Jones, C.N., and Bonvin, D. (2018). Real-time optimization of uncertain process systems via modifier adaptation and Gaussian processes. In *European Control Conference (ECC'18)*.

François, G. and Bonvin, D. (2013). Modifier-adaptation methodology for real-time optimization. *Industrial & Engineering Chemistry Research*, 53(13), 5148–5159.

Gao, W., Wenzel, S., and Engell, S. (2015). Modifier adaptation with quadratic approximation in iterative optimizing control. In *European Control Conference (ECC'15)*.

Krige, D.G. (1951). *A statistical approach to some mine valuations and allied problems at the Witwatersrand*. Ph.D. thesis, University of Witwatersrand.

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

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

Marlin, T.E. and Hrymak, A.N. (1997). Real-time operations optimization of continuous processes. In *AIChE Symposium Series - CPC-V*, volume 93, 156–164.

Mendoza, D.F., Graciano, J.E.A., Liporace, F.S., and Le Roux, G.A.C. (2016). Assessing the reliability of different real-time optimization methodologies. *The Canadian Journal of Chemical Engineering*, 94(3), 485–497.

Navia, D., Briceño, L., Gutiérrez, G., and de Prada, C. (2015). Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem. *Industrial & Engineering Chemistry Research*, 54(48), 12054–12071.

Rasmussen, C.E. and Williams, C.K.I. (2016). *Gaussian Processes for Machine Learning*. MIT Press.

Rodger, E.A. and Chachuat, B. (2011). Design methodology of modifier adaptation for on-line optimization of uncertain processes. *IFAC Proceedings Volumes*, 44(1), 4113–4118.

Singhal, M., Marchetti, A.G., Faulwasser, T., and Bonvin, D. (2016). Improved directional derivatives for modifier-adaptation schemes. *IFAC-PapersOnLine*, 50, 5718–5723.

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