

# Use of Transient Measurements for the Optimization of Steady-State Performance via Modifier Adaptation

Grégory François and Dominique Bonvin\*

*Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne*

*CH-1015 Lausanne, Switzerland*

E-mail: dominique.bonvin@epfl.ch

## Abstract

Real-time optimization (RTO) methods use measurements to offset the effect of uncertainty and drive the plant to optimality. RTO schemes differ in the way measurements are incorporated in the optimization framework. Explicit RTO schemes solve a static optimization problem repeatedly, with each iteration requiring transient operation of the plant to steady state. In contrast, implicit RTO methods use transient measurements to bring the plant to steady-state optimality in a single iteration, provided the set of active constraints is known. This paper considers the explicit RTO scheme “modifier adaptation” (MA) and proposes a framework that allows using transient measurements for the purpose of steady-state optimization. It is shown that convergence to the plant optimum can be achieved in a single transient operation provided the plant gradients can be estimated accurately. The approach is illustrated through the simulated example of a continuous stirred-tank reactor. The time needed for convergence is of the order of the plant settling time, while more than five iterations to steady state are required with conventional (static) MA. In other words, MA using transient information is able to compete in performance with RTO schemes based on gradient control, with the additional ability to

---

\*To whom correspondence should be addressed

handle plant constraints.

**Keywords :** Real-time optimization, Modifier adaptation, Plant-model mismatch, Gradient estimation, Gradient control.

## Introduction

Optimal operation of chemical processes is key for meeting productivity, quality, safety and environmental objectives. Both model-based and data-driven schemes are used to compute optimal operating conditions. The model-based techniques are intuitive and widespread, but they suffer from the presence of plant-model mismatch. Furthermore, even with an accurate plant model, the presence of disturbances generally leads to a drift of the optimal operating conditions, and adaptation based on measurements is needed to maintain plant optimality. On the other hand, data-driven optimization techniques rely exclusively on measurements to adjust the optimal inputs in real time. Consequently, real-time measurements are typically used to help achieve plant optimality. This field, which is labeled real-time optimization (RTO), has received growing attention in recent years.

*Explicit RTO schemes* solve the optimization problem repeatedly. For example, the two-step approach uses (i) measurements to update the model parameters (that is, to refine the plant model), and (ii) the updated model to perform the optimization<sup>1</sup>. It has also been proposed to update the model differently. Instead of adjusting the model parameters, one updates input-affine correction terms that are added to the cost and constraint functions of the optimization problem. The technique, labeled modifier adaptation (MA), forces the modeled cost and constraints to match the plant values<sup>2-6</sup>. The main advantage of MA lies in its proven ability to converge to the plant optimum, even in the presence of structural plant-model mismatch, a case where the two-step approach will generally fail. Hence, MA is also capable of detecting the correct set of active plant constraints without additional assumptions. MA is a static optimization method, which means that its application to a continuous plant requires waiting for steady state before taking measurements,

updating the correction terms and repeating the numerical optimization. Thus, several iterations are generally required to achieve convergence.

In contrast, *implicit RTO schemes*, such as self-optimizing control<sup>7</sup> and NCO tracking<sup>8</sup>, propose to adjust the inputs on-line in a control-inspired manner. In the absence of constraints, or when assumptions can be made regarding the set of plant constraints that are active at the optimum, implicit RTO methods reduce to gradient control, as the degrees of freedom are adjusted in real time to drive the plant cost gradient to zero. The main difficulty lies in the estimation of the steady-state plant gradient, a task that can be achieved using either measurements or, better, measurements together with a plant model<sup>9</sup>. Implicit RTO is much more challenging when the set of active constraints is unknown, as not only the cost gradient has to be inferred from the measurements but also the set of active constraints and the constraint gradients.

This paper proposes a framework for using MA during the transient phase toward steady state, thereby attempting to reach optimality in a single iteration to steady state. For this, two features are required: (i) the model-based optimization problem needs to be solved online in real time, which is made easier by the use of convex approximations<sup>10</sup>, and (ii) the modifiers are computed using transient measurements. Since there is no conceptual difference between estimating the cost gradient and the constraint gradients, we propose to take inspiration from gradient control and extend some of the related unconstrained methods<sup>9</sup> to make them fit the MA framework. In particular, both the linearization-based gradient-estimation method associated with neighboring extremals (NE) and the multiple-unit method (MU) are extended to fit the proposed MA scheme using transient information.

The paper is organized as follows. The problem formulation and the static MA formulation are presented in Section 2. Section 3 introduces both the framework for using MA during transient operation and a way of estimating the modifiers. The application of the proposed methodology is illustrated through a simulated 2-input 6-constraint CSTR in Section 5, and Section 6 concludes the paper.

## Problem Formulation

The problem of optimizing the plant performance at steady state, in the presence of constraints, can be formulated mathematically as a nonlinear program (NLP):

$$\begin{aligned} \mathbf{u}_p^* &:= \arg \min_{\mathbf{u}} \phi_p(\mathbf{u}) \\ \text{s.t. } & \mathbf{G}_p(\mathbf{u}) \leq \mathbf{0}, \end{aligned} \quad (1)$$

where  $\mathbf{u}$  is the  $n_u$ -dimensional vector of inputs,  $\mathbf{G}_p$  is the  $n_G$ -dimensional vector of plant constraints and  $\phi_p(\mathbf{u})$  is the scalar cost function. Here, the subscript  $(\cdot)_p$  indicates a quantity related to the plant.

The necessary conditions of optimality for the plant are:

$$\begin{aligned} \mathbf{G}_p(\mathbf{u}_p^*) \leq \mathbf{0}, \quad v_p^* \geq \mathbf{0}, \quad v_p^{*T} \mathbf{G}_p(\mathbf{u}_p^*) &= \mathbf{0} \\ \nabla_{\mathbf{u}} \phi_p(\mathbf{u}_p^*) + v_p^{*T} \nabla_{\mathbf{u}} \mathbf{G}_p(\mathbf{u}_p^*) &= \mathbf{0}, \end{aligned} \quad (2)$$

where  $v_p$  is the  $n_G$ -dimensional vector of Lagrange multipliers.

In practice, the functions  $\phi_p$  and  $\mathbf{G}_p$  are unknown. A steady-state plant model is used to construct the following model-based NLP:

$$\begin{aligned} \mathbf{u}^* &:= \arg \min_{\mathbf{u}} \varphi(\mathbf{u}, \bar{\mathbf{y}}, \theta) \\ \text{s.t. } & \dot{\bar{\mathbf{x}}} = \mathbf{F}(\mathbf{u}, \bar{\mathbf{x}}, \theta) = \mathbf{0} \\ & \bar{\mathbf{y}} = \mathbf{h}(\mathbf{u}, \bar{\mathbf{x}}, \theta) \\ & \mathbf{g}(\mathbf{u}, \bar{\mathbf{y}}, \theta) \leq \mathbf{0}, \end{aligned} \quad (3)$$

where  $\varphi$  is the model cost function,  $\mathbf{F}$  the  $n$ -dimensional vector function representing the dynamic model,  $\mathbf{g}$  the  $n_G$ -dimensional vector of constraint functions,  $\bar{\mathbf{x}}$  the  $n$ -dimensional state vector at steady state,  $\bar{\mathbf{y}}$  the  $p$ -dimensional vector of outputs at steady state,  $\theta$  the  $q$ -dimensional vector

of uncertain model parameters. For the sake of simplicity, we assume that there exist explicit functions  $\bar{\mathbf{y}} = \mathbf{H}(\mathbf{u}, \theta)$ , which allows reformulating Problem (3) as follows:

$$\begin{aligned} \mathbf{u}^* &= \arg \min_{\mathbf{u}} \phi(\mathbf{u}, \theta) \\ \text{s.t. } &\mathbf{G}(\mathbf{u}, \theta) \leq \mathbf{0}, \end{aligned} \quad (4)$$

where  $\phi$  and  $\mathbf{G}$  represent the models of the cost and constraint functions, respectively. These models require the identification of the model parameters  $\theta$ . We will assume in this paper that  $\phi$  and  $\mathbf{G}$  are twice differentiable. The model NCO read:

$$\begin{aligned} \mathbf{G}(\mathbf{u}^*, \theta) \leq \mathbf{0}, \quad \mathbf{v}^* \geq \mathbf{0}, \quad \mathbf{v}^{*T} \mathbf{G}(\mathbf{u}^*, \theta) &= \mathbf{0} \\ \nabla_{\mathbf{u}} \phi(\mathbf{u}^*, \theta) + \mathbf{v}^{*T} \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}^*, \theta) &= \mathbf{0}. \end{aligned} \quad (5)$$

## Measurement-Based Real-Time Optimization

If the model matches the plant perfectly, solving the model-based optimization problem solves Problem (1). Unfortunately, this is rarely the case, and usually  $\mathbf{u}^* \neq \mathbf{u}_p^*$ . Real-time optimization encompasses a family of methods for which plant measurements are used to update  $\mathbf{u}^*$  in order to approach  $\mathbf{u}_p^*$ . We will review next two such methods, namely, explicit RTO via modifier adaptation and implicit RTO using gradient control.

### Explicit RTO via Modifier Adaptation

With modifier adaptation, plant measurements are used to iteratively modify the model-based optimization problem (4) in such a way that, upon convergence, the NCO of the *modified* problem match those of the plant. This is made possible by using modifiers that, at each iteration, correspond to the differences between the predicted and measured values of the constraints and between the predicted and measured cost and constraint gradients. These modifiers are used to add input-

affine corrections to the model cost and constraint functions, according to the intuitive observation that first-order corrections are required to achieve matched first-order optimality conditions. At the  $k^{\text{th}}$  iteration, the optimal inputs computed using the modified problem are applied to the plant, and the resulting values of the plant constraints and of the plant cost and constraint gradients are compared to the model-based predictions. Then, the following optimization problem is solved to determine the next inputs:

$$\mathbf{u}_{k+1}^* := \arg \min_{\mathbf{u}} \phi_m(\mathbf{u}, \theta) := \phi(\mathbf{u}, \theta) + \varepsilon_k^\phi + \Lambda_k^{\phi^T} (\mathbf{u} - \mathbf{u}_k^*) \quad (6)$$

$$\text{s.t. } \mathbf{G}_m(\mathbf{u}, \theta) := \mathbf{G}(\mathbf{u}, \theta) + \varepsilon_k^G + \Lambda_k^{G^T} (\mathbf{u} - \mathbf{u}_k^*) \leq \mathbf{0} \quad (7)$$

$$\text{with } \varepsilon_k^\phi := \phi_p(\mathbf{u}_k^*) - \phi(\mathbf{u}_k^*, \theta) \quad (8)$$

$$\varepsilon_k^G := \mathbf{G}_p(\mathbf{u}_k^*) - \mathbf{G}(\mathbf{u}_k^*, \theta) \quad (9)$$

$$\Lambda_k^{\phi^T} := \nabla_{\mathbf{u}} \phi_p(\mathbf{u}_k^*) - \nabla_{\mathbf{u}} \phi(\mathbf{u}_k^*, \theta) \quad (10)$$

$$\Lambda_k^{G^T} := \nabla_{\mathbf{u}} \mathbf{G}_p(\mathbf{u}_k^*) - \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}_k^*, \theta), \quad (11)$$

where the scalar  $\varepsilon_k^\phi$  and the  $n_G$ -dimensional vector  $\varepsilon_k^G$  are the zeroth-order modifiers, and the  $n_u$ -dimensional row vector  $\Lambda_k^{\phi^T}$  and the  $(n_G \times n_u)$  matrix  $\Lambda_k^{G^T}$  represent the first-order modifiers.

As seen from Eqns (6)-(11), MA is a static optimization method that, at iteration  $k$ , requires application of the constant input values  $\mathbf{u}_k^*$  until the plant reaches steady state. Once this occurs, the modifiers are updated and used for the subsequent iteration, as illustrated in Figure 1. In practice MA is implemented with exponential filtering of the modifiers, that is, the filtered modifiers are obtained as  $\Lambda_{f,k} = \mathbf{K} \Lambda_k + (\mathbf{I} - \mathbf{K}) \Lambda_{f,k-1}$ , with the  $(n_u + 1)(n_G + 1)$ -dimensional modifier vector

$$\Lambda_k^T = \left[ \varepsilon_k^\phi, \varepsilon_k^{G_1}, \dots, \varepsilon_k^{G_{n_G}}, \Lambda_k^{\phi^T}, \Lambda_k^{G_1^T}, \dots, \Lambda_k^{G_{n_G}^T} \right]. \quad (12)$$

Exponential filtering both prevents abrupt modifications of the optimization problem between two consecutive iterations and reduces the impact of measurement noise. Also, exponential filtering preserves the properties of RTO-MA upon convergence and provides degrees of freedom (by means

of the filter gain matrix  $\mathbf{K}$ ) that can be used for enforcing convergence<sup>6</sup>. For the sake of simplicity, we will implicitly assume the presence of this filter and not distinguish between raw and filtered modifiers hereafter.

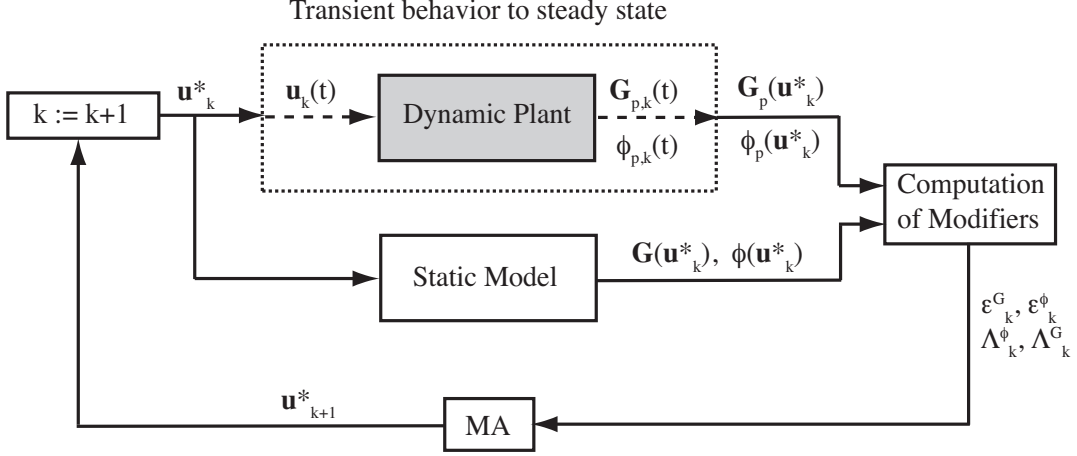


Figure 1: Basic scheme for steady-state plant optimization via MA.

The advantage of the modifier-adaptation scheme (6)-(11) lies in its ability to converge to a KKT point of the plant<sup>6</sup>. Eqns (7) and (9) show that, upon convergence at  $\mathbf{u}_\infty^*$ , one has:

$$\mathbf{G}_m(\mathbf{u}_\infty^*, \theta) = \mathbf{G}(\mathbf{u}_\infty^*, \theta) + \boldsymbol{\varepsilon}_\infty^G = \mathbf{G}_p(\mathbf{u}_\infty^*) \leq \mathbf{0}. \quad (13)$$

Hence, the zeroth-order modifiers  $\boldsymbol{\varepsilon}_k^G$  allow enforcing the feasibility conditions  $\mathbf{G}_p(\mathbf{u}_\infty^*) \leq \mathbf{0}$ . Note that the correction term  $\boldsymbol{\varepsilon}_k^\phi$  simply shifts the cost function up or down, without changing the location of its minimizer, and thus is generally discarded.

Similarly, upon differentiating Eqns (6) and (7) with respect to  $\mathbf{u}$  and using Eqns (10) and (11), one obtains upon convergence:

$$\nabla_{\mathbf{u}} \phi_m(\mathbf{u}_\infty^*, \theta) = \nabla_{\mathbf{u}} \phi(\mathbf{u}_\infty^*, \theta) + \Lambda_\infty^{\phi T} = \nabla_{\mathbf{u}} \phi_p(\mathbf{u}_\infty^*) \quad (14)$$

$$\nabla_{\mathbf{u}} \mathbf{G}_m(\mathbf{u}_\infty^*, \theta) = \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}_\infty^*, \theta) + \Lambda_\infty^{G T} = \nabla_{\mathbf{u}} \mathbf{G}_p(\mathbf{u}_\infty^*). \quad (15)$$

Hence, the first-order correction terms in the cost and constraint functions (with slopes  $\Lambda_k^{\phi T}$  and

$\Lambda_k^{G^T}$ ) modify the model gradients to force them to match the corresponding plant gradients. Note that the zeroth- and first-order corrections also lead to matched Lagrange multipliers upon convergence, thus ensuring the correct set of active constraints. In other words, modifier adaptation forces the NCO of the model-based optimization problem (4) to match those of the plant optimization problem (1).

## Implicit RTO using Gradient Control

Implicit RTO schemes recast Problem (1) as a control problem, whose controlled variables are the NCO (2), with the associated setpoints being zero. In the absence of constraints, gradient control can be implemented straightforwardly since the NCO (2) reduce to  $\nabla_{\mathbf{u}}\phi_p(\mathbf{u}_p^*) = \mathbf{0}$ . For example, the following control law can drive the plant gradient to zero in real time:

$$\dot{\mathbf{u}}(t) = -\kappa \mathbf{P}^{-1} \Gamma_\phi(t), \quad \mathbf{u}(0) = \mathbf{u}_0, \quad (16)$$

where  $\kappa$  is the controller gain matrix,  $\mathbf{P}$  an estimate of the Hessian of the plant cost,  $\Gamma_\phi(t)$  a time-dependent signal that estimates the plant cost gradient *at steady state*. Several methods exist for implementing the control law (16), which mainly differ in the way  $\Gamma_\phi(t)$  is obtained. Three model-based and three data-driven methods have been discussed and compared<sup>9</sup>. This article focuses on two of these methods, one data-driven (multiple units) and one model-based (neighboring extremals), which are presented next.

Gradient from multiple units (MU): This data-driven method assumes the availability of multiple similar units, as in the case of fuel cell stacks or large array of microreactors. The inputs to the various units differ by an offset, and the gradient is estimated delay-free from the difference in the measured costs<sup>11</sup>. The input offset represents the excitation needed to estimate the plant gradient.

For the single-input case, two units are needed, which are labeled 'a' and 'b'. It follows:

$$u_a(t) = u(t), \quad u_b(t) = u(t) + \Delta, \quad \Gamma_\phi(t) = \frac{\phi_{p,b}(t) - \phi_{p,a}(t)}{\Delta}, \quad (17)$$



where  $\Delta$  represents the input offset. For the multivariable case, the typical configuration is to work with  $n_u + 1$  units. Only one input direction is perturbed for each of the first  $n_u$  units, while the last unit has no input offset. The gradient is computed in the  $n_u$  directions by comparing the cost of each of the first  $n_u$  units to the cost of the last one, as in the one-dimensional case. Note that only the cost needs to be measured online and that the gradient can be computed easily without delay.

Gradient from neighboring extremals (NE): Uncertainty causes the optimal inputs and outputs as well as the gradient to deviate from their nominal values. For the unconstrained case, NE proposes to estimate the gradient on the basis of a variational analysis around the *nominal* operating point  $\mathbf{u}_0^*$  obtained by solving Problem (4) – without constraints – for the nominal parameter values  $\theta_0$ , for which the nominal gradient  $\nabla_{\mathbf{u}}\phi(\mathbf{u}_0^*, \theta_0) = \mathbf{0}$ .<sup>12</sup>

From  $\bar{\mathbf{y}} = \mathbf{H}(\mathbf{u}, \theta)$ , the variation of the steady-state outputs can be expressed as:

$$\delta\bar{\mathbf{y}} = \nabla_{\mathbf{u}}\mathbf{H} \delta\mathbf{u} + \nabla_{\theta}\mathbf{H} \delta\theta, \quad (18)$$

with  $\delta\bar{\mathbf{y}} := \bar{\mathbf{y}} - \bar{\mathbf{y}}_0^*$ ,  $\delta\mathbf{u} := \mathbf{u} - \mathbf{u}_0^*$  and  $\delta\theta := \theta - \theta_0$ , where  $\bar{\mathbf{y}}_0^* = \mathbf{H}(\mathbf{u}_0^*, \theta)$ . For  $p \geq q$ , the variation of the uncertain parameters can be estimated from Eq. (18) using input and output measurements:

$$\delta\theta = (\nabla_{\theta}\mathbf{H})^+ [\delta\bar{\mathbf{y}} - \nabla_{\mathbf{u}}\mathbf{H}\delta\mathbf{u}]. \quad (19)$$

Similarly, the gradient can be written as:

$$\nabla_{\mathbf{u}}\phi = \nabla_{\mathbf{u}\mathbf{u}}^2\phi \delta\mathbf{u} + \nabla_{\mathbf{u}\theta}^2\phi \delta\theta, \quad (20)$$

which, with Eq. (19), allows writing the gradient in terms of  $\delta\bar{\mathbf{y}}$  and  $\delta\mathbf{u}$ :

$$\nabla_{\mathbf{u}}\phi = \nabla_{\mathbf{u}\theta}^2\phi (\nabla_{\theta}\mathbf{H})^+ \delta\bar{\mathbf{y}} + \left( \nabla_{\mathbf{u}\mathbf{u}}^2\phi - \nabla_{\mathbf{u}\theta}^2\phi (\nabla_{\theta}\mathbf{H})^+ \nabla_{\mathbf{u}}\mathbf{H} \right) \delta\mathbf{u}. \quad (21)$$

It follows that upon:

- assuming that the  $p$  model outputs can be measured online for the plant, that is,  $\mathbf{y}_p(t)$  is available,
- defining  $\delta \mathbf{y}_p(t) = \mathbf{y}_p(t) - \bar{\mathbf{y}}_0^*$  and  $\delta \mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}_0^*$ ,

the following expression for the gradient estimate  $\Gamma_\phi(t)$  and the control law can be obtained:<sup>9,12</sup>

$$\Gamma_\phi(t) = \nabla_{\mathbf{u}\theta}^2 \phi (\nabla_{\theta} \mathbf{H})^+ \delta \mathbf{y}_p(t) + \left( \nabla_{\mathbf{u}\mathbf{u}}^2 \phi - \nabla_{\mathbf{u}\theta}^2 \phi (\nabla_{\theta} \mathbf{H})^+ \nabla_{\mathbf{u}} \mathbf{H} \right) \delta \mathbf{u}(t), \quad (22)$$

$$\delta \dot{\mathbf{u}}(t) = -\kappa \mathbf{P}^{-1} \Gamma_\phi(t), \quad \delta \mathbf{u}(0) = 0. \quad (23)$$

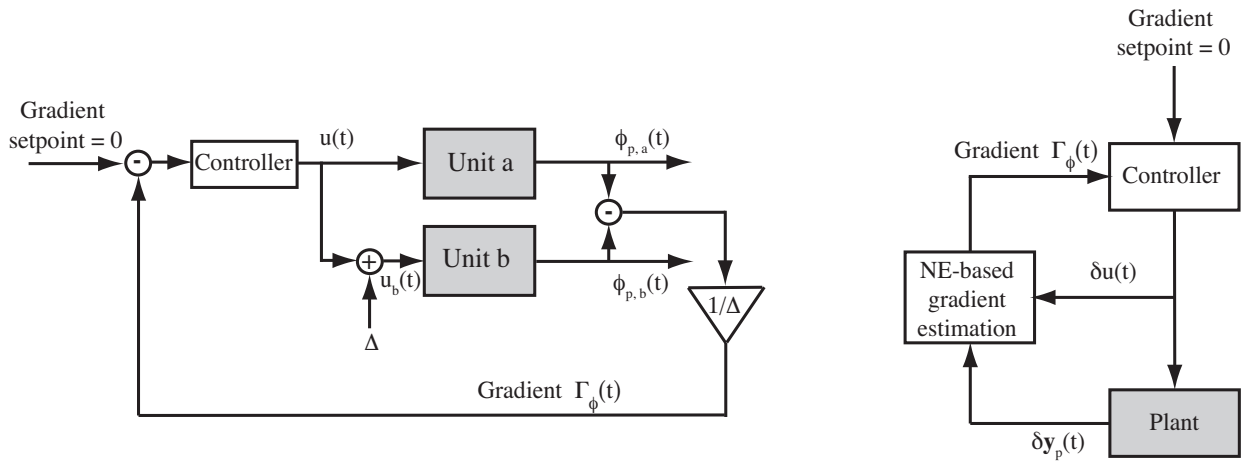


Figure 2: Gradient control for the single-input case. Use of MU (right-hand side) and NE (left-hand side) to compute the gradient.

### Remarks

1. The two gradient-control methods discussed above are depicted in Figure 2 for the single-input case. These two methods were originally introduced as RTO methods<sup>11,12</sup>, although their originality lies more in the way the gradient is estimated than in the way control is performed. For instance, it is obvious that the novelty of the MU method is in gradient computation and not in the use of integral control. Regarding the NE method, although the approach was proposed to compute the input update that is required to offset the effect of  $\delta \theta$  on optimality, this input update was shown to correspond to the deadbeat control of a first-order approximation of the gradient<sup>12</sup>.

2. NE relies on the assumption that the uncertainty is of parametric nature, while MU does not. As shown by Eq. (21), NE can be used to estimate the steady-state cost gradient using steady-state measurements<sup>12</sup>. In contrast, Eq. (22) proposes to use *transient* measurements to compute online an estimate of the steady-state cost gradient. This approach has been shown to be similar to self-optimizing control based on the null-space method<sup>9,13</sup>. Schematically, Eq. (22) uses the inputs and the measured outputs as if the plant were at steady state. It turns out that, if the plant reaches steady state, the steady-state gradient is estimated accurately.
3. With MU, the excitation required to estimate the gradient is obtained by perturbing the inputs to the various units, while no excitation is necessary with NE since additional information is available in the form of output measurements and a nominal model.
4. In the presence of constraints, assumptions have to be made regarding the constraints that are active at the plant optimum<sup>8</sup>. As a consequence, the control law (16) pushes the *reduced gradients* to zero, with the gradient terms estimated using for example the two aforementioned techniques.
5. Direct use of the dual feasibility condition (the 2nd row of Eq. 5) as a control law has also been considered<sup>14,15</sup>. However, these approaches have only been investigated for the case of perfect modeling or for the numerical optimization of analytical functions, that is, with no model error.
6. Extremum-seeking control techniques can be used to drive a dynamic plant to steady-state optimality using transient measurements<sup>16–18</sup>. However, these techniques require multiple time-scale separations, which strongly penalizes the convergence time<sup>9</sup>. This is even more acute when the number of inputs increases. Hence, in the presence of uncertainty and constraints, there are no implicit RTO techniques capable of driving a plant to steady-state optimality with a convergence time of the order of the plant settling time.

# Modifier Adaptation using Transient Measurements

## Basic Idea

Modifier adaptation has two main features, namely, convergence to the plant optimum even in the presence of structural plant-model mismatch and the possibility of handling constraints explicitly. In this subsection, we propose a MA framework that uses *transient measurements* to estimate the *steady-state* values of the modifiers, thus allowing convergence to the steady-state plant optimum within a single iteration. For this purpose, measurements at each re-optimization instant (during the transient) are used to estimate the modifiers of the optimization problem, the solution of which provides the new set of constant inputs to be applied to the plant until the next re-optimization instant.

The philosophy behind this framework is inspired from gradient-control techniques, which use transient information for steady-state optimization – despite the fact that, strictly speaking, plant gradients are only defined at steady state. For the unconstrained case, the estimated signal  $\Gamma_\phi(t)$  is controlled to zero using integral control as given by Eq. (16). This is justified since, with this control law, the plant reaches steady state when  $\dot{\mathbf{u}} = \mathbf{0}$ , i.e. when  $\Gamma_\phi = \mathbf{0}$ . If the estimated signal  $\Gamma_\phi(t)$  represents the true gradient, the plant will reach a steady state that satisfies the NCO. This paper proposes to implement MA like gradient control, that is, online use of an estimated value of the steady-state gradient. The main difference is that the control update is not obtained by computing a control law, but rather by solving a (modified) optimization problem.

At each re-optimization instant during transient operation, denoted here by the index  $j$ , the scheme determines the constant inputs  $\mathbf{u}_{j+1}^*$  that are applied until the next re-optimization instant.

The optimization problem for computing  $\mathbf{u}_{j+1}^*$  reads:

$$\mathbf{u}_{j+1}^* := \arg \min_{\mathbf{u}} \phi_m(\mathbf{u}, \theta) := \phi(\mathbf{u}, \theta) + \hat{\varepsilon}_j^\phi + \hat{\Lambda}_j^{\phi^T} (\mathbf{u} - \mathbf{u}_j^*) \quad (24)$$

$$\text{s.t. } \mathbf{G}_m(\mathbf{u}, \theta) := \mathbf{G}(\mathbf{u}, \theta) + \hat{\varepsilon}_j^G + \hat{\Lambda}_j^{G^T} (\mathbf{u} - \mathbf{u}_j^*) \leq \mathbf{0} \quad (25)$$

$$\text{with } \hat{\varepsilon}_j^\phi := \hat{\phi}_p(\mathbf{u}_j^*) - \phi(\mathbf{u}_j^*, \theta) \quad (26)$$

$$\hat{\varepsilon}_j^G := \hat{\mathbf{G}}_p(\mathbf{u}_j^*) - \mathbf{G}(\mathbf{u}_j^*, \theta) \quad (27)$$

$$\hat{\Lambda}_j^{\phi^T} := \widehat{\nabla_{\mathbf{u}} \phi_p}(\mathbf{u}_j^*) - \nabla_{\mathbf{u}} \phi(\mathbf{u}_j^*, \theta) \quad (28)$$

$$\hat{\Lambda}_j^{G^T} := \widehat{\nabla_{\mathbf{u}} \mathbf{G}_p}(\mathbf{u}_j^*) - \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}_j^*, \theta), \quad (29)$$

where the notation  $\widehat{(\cdot)}$  indicates an estimated *steady-state value* corresponding to the current inputs  $\mathbf{u}_j^*$ . Again, it is necessary here to estimate the steady-state values of the modifiers since, strictly speaking, the modifiers are only defined at steady state. The conditions ensuring that the point reached upon convergence is optimal for the plant are given in the following theorem.

### Theorem 1

*Consider the MA problem that uses transient measurements with the inputs computed iteratively as the solution to the optimization problem (24)-(29). If the controlled plant reaches steady state and the estimates  $\hat{\phi}_p$ ,  $\hat{\mathbf{G}}_p$ ,  $\widehat{\nabla_{\mathbf{u}} \phi_p}$  and  $\widehat{\nabla_{\mathbf{u}} \mathbf{G}_p}$  converge to their true values, then the plant will satisfy the NCO (2).*

**Proof:** Let start by making the obvious remark that the plant reaching steady state implies convergence of the iterative scheme (24)-(29) since, otherwise, the inputs will change, thereby preventing the plant to reach steady state. The conditions  $\hat{\phi}_p(\mathbf{u}_\infty^*) = \phi_p(\mathbf{u}_\infty^*)$ ,  $\hat{\mathbf{G}}_p(\mathbf{u}_\infty^*) = \mathbf{G}_p(\mathbf{u}_\infty^*)$ ,  $\widehat{\nabla_{\mathbf{u}} \phi_p}(\mathbf{u}_\infty^*) = \nabla_{\mathbf{u}} \phi_p(\mathbf{u}_\infty^*)$  and  $\widehat{\nabla_{\mathbf{u}} \mathbf{G}_p}(\mathbf{u}_\infty^*) = \nabla_{\mathbf{u}} \mathbf{G}_p(\mathbf{u}_\infty^*)$  imply  $\hat{\varepsilon}_\infty^\phi = \varepsilon_\infty^\phi$ ,  $\hat{\varepsilon}_\infty^G = \varepsilon_\infty^G$ ,  $\hat{\Lambda}_\infty^{\phi^T} = \Lambda_\infty^{\phi^T}$  and  $\hat{\Lambda}_\infty^{G^T} = \Lambda_\infty^{G^T}$ , and thus the plant satisfies the NCO (2).

### Remarks

1. The conditions of Theorem 1 are very similar to those for applying gradient control. In the unconstrained case, for the gradient control law to converge to a KKT point of the plant, it

is required that (i) the plant reaches steady state, and (ii) the gradient estimates tend to the steady-state plant gradients.

2. The choice of a very low re-optimization frequency corresponds to static MA. In other words, if one waits long enough before re-optimization, steady-state will eventually be reached, and the proposed scheme corresponds formally to static MA.
3. Similarly to static MA, it is advantageous to exponentially filter the estimated modifiers, with the filtered modifiers being obtained as  $\hat{\Lambda}_{f,j} = \mathbf{K}\hat{\Lambda}_j + (\mathbf{I} - \mathbf{K})\hat{\Lambda}_{f,j-1}$ . Filtering helps in the presence of measurement noise, but also when the plant dynamics are particularly tricky as in the case of unstable internal dynamics.

## Estimation of Static Modifier Terms

The key requirement in Theorem 1 is the ability to estimate, during transient, the values that the modifiers would have if the plant would stabilize at the steady state corresponding to the current inputs  $\mathbf{u}_j^*$ . This section proposes several schemes for performing this estimation. The schemes rely on the validity of some assumptions:

- A1: All process variables needed to calculate the constraints are available on-line or, alternatively, the plant constraints are measured online, that is, the *signals*  $\mathbf{G}_p(t)$  are available.
- A2: All process variables needed to calculate the plant cost are available on-line or, alternatively, the plant cost is measured online, that is, the *signal*  $\phi_p(t)$  is available.
- A3: The plant outputs are measured online, that is, the *signals*  $\mathbf{y}_p(t)$  are available. There are  $p$  independent output measurements, with  $p \geq q$ .

Assumption A1 will be required for computing the zeroth-order modifiers  $\hat{\epsilon}_j^G$  associated with the constraints, whereas either A1 and A2 or A3 will be used for estimating the first-order modifiers.

These assumptions are often met in practice when the plant is composed of a single or a few units. However, for large plants encompassing several units, certain constraints cannot be measured online. In fact, since the cost and constraints functions are inherited from the steady-state problem formulation, their running values might not even be available during transient operation, or might only be available at low frequency. The latter case implies that the re-optimization frequency is limited. If some of these measurements are only available at steady state, an alternative is to estimate the missing running cost and constraint values from available output measurements.

Another important remark concerns the computational burden of the proposed framework for large plants. In the presence of numerous constraints and manipulated variables, the online solution of the modified optimization problem can be difficult. Fortunately, the plant model can be replaced by a convex response-surface approximation<sup>10</sup>, thus allowing the use of fast and reliable convex optimization methods while preserving the ability of RTO-MA to converge to the plant optimum.

### Estimation of the zeroth-order modifiers

If A1 holds, the following scheme can be used for estimating  $\hat{\boldsymbol{\varepsilon}}_j^G$ :

$$\hat{\boldsymbol{\varepsilon}}_j^G = \mathbf{G}_p(t_j) - \mathbf{G}(\mathbf{u}_j^*, \boldsymbol{\theta}). \quad (30)$$

In other words,  $\hat{\boldsymbol{\varepsilon}}_j^G$  is estimated at the time instant  $t_j$  as the difference between the *measured* (or estimated from measurements) constraints  $\mathbf{G}_p(t_j)$  and the values that the *modeled* constraints would have at the steady state corresponding to  $\mathbf{u}_j^*$ . This choice is motivated by the fact that, when  $t_j \rightarrow t_\infty$ ,  $\mathbf{u}_j^* \rightarrow \mathbf{u}_\infty^*$  and  $\mathbf{G}_p(t_\infty)$  will be the plant constraints  $\mathbf{G}_p(\mathbf{u}_\infty^*)$  associated with the converged inputs  $\mathbf{u}_\infty^*$ .

Similarly, if needed, the zeroth-order modifiers associated with the cost can be estimated as:

$$\hat{\boldsymbol{\varepsilon}}_j^\phi = \phi_p(t_j) - \phi(\mathbf{u}_j^*, \boldsymbol{\theta}), \quad (31)$$

provided Assumption A2 holds.

## Estimation of the first-order modifiers

The computation of the first-order modifiers requires the estimation of plant gradients, for which several methods are available<sup>9</sup>. Two methods are presented next to illustrate (i) the use of only cost and constraint measurements (via multiple units), and (ii) the use of input and output measurements<sup>1</sup> (via the neighboring-extremal approach).

Gradients from multiple units (MU): When Assumptions A1 and A2 hold, it is possible to estimate the first-order modifiers associated with the cost and the constraints similarly to the way  $\Gamma_\phi(t)$  is estimated in Eq. (17) for the gradient-control case. We assume the existence of  $n_u + 1$  identical units, for which the cost and the constraints are measured. One of these units is labeled the *main unit*, indexed with the subscript  $a$ , while the  $n_u$  remaining units are indexed with  $i \in [1; n_u]$ .

At the time instant  $t_j$ ,  $\mathbf{u}_j^*$  is applied to the main unit, while the  $i^{th}$  unit of the  $n_u$  remaining units is presented with

$\mathbf{u}_{j,i}^* = \left[ u_{j,1}^* \dots u_{j,i-1}^* \quad u_{j,i}^* + \Delta \quad u_{j,i+1}^* \dots u_{j,m}^* \right]^T$ , that is, with the  $i^{th}$  coordinate of  $\mathbf{u}_j^*$  offset by  $\Delta$ .

Denoting by  $\phi_{p,a}(t_j)$  and  $\mathbf{G}_{p,a}(t_j)$  the measured values of the cost and constraints for the main unit at the time instant  $t_j$ , and by  $\phi_{p,i}(t_j)$  and  $\mathbf{G}_{p,i}(t_j)$  the corresponding quantities for each unit  $i \in [1; n_u]$ , the elements of the first-order modifiers are estimated as the finite differences *between units*:

$$\Gamma_{\phi_i}(t_j) = \widehat{\nabla_{u_i} \phi_p}(\mathbf{u}_j^*) = \frac{\phi_{p,i}(t_j) - \phi_{p,a}(t_j)}{\Delta} \quad (32)$$

$$\Gamma_{G_i^h}(t_j) = \widehat{\nabla_{u_i} G_p^h}(\mathbf{u}_j^*) = \frac{G_{p,i}^h(t_j) - G_{p,a}^h(t_j)}{\Delta}, \quad (33)$$

where  $\widehat{\nabla_{u_i} \phi_p}(\mathbf{u}_j^*)$  is the  $i^{th}$  coordinate of the vector  $\widehat{\nabla_{\mathbf{u}} \phi_p}(\mathbf{u}_j^*)$ ,  $h \in [1; n_G]$  is the counter for the constraints, and  $\widehat{\nabla_{u_i} G_p^h}(\mathbf{u}_j^*)$  is the  $(h, i)^{th}$  element of the matrix  $\widehat{\nabla_{\mathbf{u}} \mathbf{G}_p}(\mathbf{u}_j^*)$ .

Eqns (32) and (33) describe the estimation of the cost and constraint gradients with the MU method in the multi-dimensional case<sup>11</sup>. Note that the same excitation can be used to estimate both the cost and the constraint gradients, which confirms the observation that the complexity of

<sup>1</sup>If available, the cost and constraint measurements can be considered as elements of the output vector.



the MU method grows only with the number of inputs<sup>9</sup>, also in the presence of constraints.

Gradients from neighboring extremals (NE): When Assumption A3 holds, additional information is available through the outputs, which allows estimating the gradient terms without input excitation<sup>9</sup>. We exploit here the way  $\Gamma_\phi(t)$  is estimated in Eq. (22). A variational analysis at steady state gives:

$$\begin{aligned}\nabla_{\mathbf{u}}\phi(\mathbf{u}, \theta) &= \nabla_{\mathbf{u}}\phi(\mathbf{u}_0^*, \theta) + \nabla_{\mathbf{u}\theta}^2\phi(\nabla_{\theta}\mathbf{H})^+ \delta\bar{\mathbf{y}} \\ &\quad + \left(\nabla_{\mathbf{u}\mathbf{u}}^2\phi - \nabla_{\mathbf{u}\theta}^2\phi(\nabla_{\theta}\mathbf{H})^+ \nabla_{\mathbf{u}}\mathbf{H}\right)\delta\mathbf{u}\end{aligned}\quad (34)$$

$$\begin{aligned}\nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}, \theta) &= \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_0^*, \theta) + \nabla_{\mathbf{u}\theta}^2\mathbf{G}(\nabla_{\theta}\mathbf{H})^+ \delta\bar{\mathbf{y}} \\ &\quad + \left(\nabla_{\mathbf{u}\mathbf{u}}^2\mathbf{G} - \nabla_{\mathbf{u}\theta}^2\mathbf{G}(\nabla_{\theta}\mathbf{H})^+ \nabla_{\mathbf{u}}\mathbf{H}\right)\delta\mathbf{u},\end{aligned}\quad (35)$$

with  $\delta\bar{\mathbf{y}} = \bar{\mathbf{y}} - \bar{\mathbf{y}}_0^*$  and  $\delta\mathbf{u} = \mathbf{u} - \mathbf{u}_0^*$  the output and input deviations from the nominal operating point. Note that the cost gradient  $\nabla_{\mathbf{u}}\phi(\mathbf{u}_0^*, \theta)$  and constraint gradients  $\nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_0^*, \theta)$  do not always vanish in the constrained case.

We propose to use the output measurements  $\delta\mathbf{y}_p(t_j) = \mathbf{y}_p(t_j) - \bar{\mathbf{y}}_0^*$  and the current values of the inputs  $\delta\mathbf{u}^*(t_j) = \mathbf{u}_j^* - \mathbf{u}_0^*$  to estimate the gradients:

$$\begin{aligned}\Gamma_\phi(t_j) &= \widehat{\nabla_{\mathbf{u}}\phi_p}(\mathbf{u}_j^*) = \nabla_{\mathbf{u}}\phi(\mathbf{u}_0^*, \theta) + \nabla_{\mathbf{u}\theta}^2\phi(\nabla_{\theta}\mathbf{H})^+ \delta\mathbf{y}_p(t_j) \\ &\quad + \left(\nabla_{\mathbf{u}\mathbf{u}}^2\phi - \nabla_{\mathbf{u}\theta}^2\phi(\nabla_{\theta}\mathbf{H})^+ \nabla_{\mathbf{u}}\mathbf{H}\right)\delta\mathbf{u}^*(t_j)\end{aligned}\quad (36)$$

$$\begin{aligned}\Gamma_G(t_j) &= \widehat{\nabla_{\mathbf{u}}\mathbf{G}_p}(\mathbf{u}_j^*) = \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_0^*, \theta) + \nabla_{\mathbf{u}\theta}^2\mathbf{G}(\nabla_{\theta}\mathbf{H})^+ \delta\mathbf{y}_p(t_j) \\ &\quad + \left(\nabla_{\mathbf{u}\mathbf{u}}^2\mathbf{G} - \nabla_{\mathbf{u}\theta}^2\mathbf{G}(\nabla_{\theta}\mathbf{H})^+ \nabla_{\mathbf{u}}\mathbf{H}\right)\delta\mathbf{u}^*(t_j).\end{aligned}\quad (37)$$

Hence, the way gradients are estimated online with NE is similar to the way they would be estimated at steady state, but for the fact that we propose to use running instead of steady-state values of the measured outputs. This is consistent with standard NE in that, when the plant reaches steady state, the increments  $\delta\mathbf{y}_p(t_j)$  and  $\delta\mathbf{u}^*(t_j)$  correspond to the standard values  $\delta\bar{\mathbf{y}}$  and  $\delta\bar{\mathbf{u}}$ . This has already been proposed and successfully tested in simulation in the unconstrained case<sup>9</sup>. Note that

the model used for estimating the gradients with NE can differ from the model used at the optimization layer: one can use a detailed model for estimating the gradients and a simplified model for optimization purposes. This is indeed how RTO-MA will be implemented in the illustrative example, i.e. using (i) a convex approximation of the model in the optimization layer (for fast solution and enforcement of the model adequacy conditions that are necessary for convergence<sup>6,10</sup>), and (ii) the detailed model for NE.

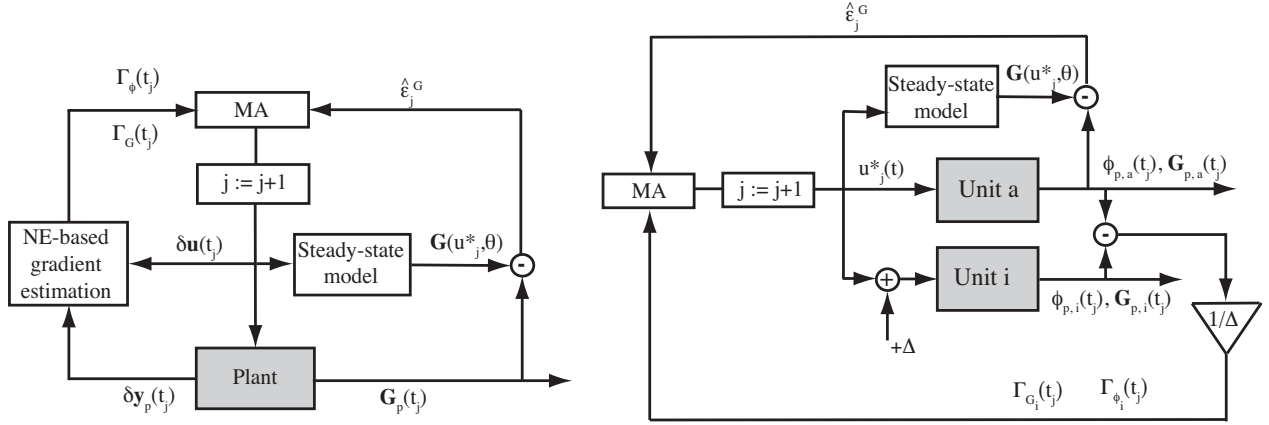


Figure 3: MA using transient information and either MU (right-hand side) or NE (left-hand side) to estimate the plant gradients.

### Remarks

1. Figure 3 illustrates the way MA using transient information is combined with the two proposed gradient-estimation schemes. For each method, only the gradient-estimation step is outlined in the figure, as RTO is performed using MA as given by Eqns (24)-(29). This reinforces the idea that, in most RTO schemes, gradient estimation can be kept separate from the control/optimization step, as virtually any controller can be used to perform gradient control given a gradient estimate. Conversely, any gradient-estimation method can in principle be used by any measurement-based optimization technique that uses gradients.
2. The estimated modifiers are consistent in the sense that they tend to their static counterparts when the plant reaches steady state. For example, with the MU method, the gradient estimates tend to the values estimated using finite differences. With NE, the gradient esti-

mates tend to the values obtained when static MA is applied using static NE to estimate the gradients<sup>12</sup>.

3. The estimated first-order modifiers inherit the accuracy of the method used to estimate the plant gradients. For example, the MU-based gradient-estimation scheme has the (low) accuracy of a finite-difference scheme, while the accuracy of the NE-based estimation scheme is limited by the accuracy of linearization, which decreases with larger  $\delta\theta$ . With NE, it is possible to reduce the error due to linearization by repeating the linearization around the converged operating conditions, i.e. updating the matrices in Eqns (34)-(35).
4. The presence of measurement noise is clearly detrimental to MU since MU relies on a finite-difference scheme between the units and identical units are required, which is rarely the case. Note however that some progress has been made to reduce the need for exactly identical units<sup>19</sup>. NE is less affected by measurement noise since it does not rely on numerical differentiation. Still measurement noise affects the quality of the gradient estimates via  $\delta\mathbf{y}_p(t_j)$ , but modifiers are typically filtered.
5. Since no assumption is made regarding the structure of the uncertainty, the MU-based estimation scheme is able to perform well even when the source of uncertainty is either unknown or of non-parametric nature as in the case of plant-model mismatch. In contrast, NE assumes that the identity of the uncertain parameters is known, but only the *nominal* parameter values are used to compute the matrices in Eqns (36)-(37), and neither the values of the model parameters nor these matrices are updated. The NE-based estimation scheme has been shown to perform well also when the source of uncertainty is unknown or in the presence of structural plant-model mismatch<sup>9</sup>.
6. The MU method is clearly not suited for large industrial plants, where the curse of dimensionality makes the task of estimating plant gradients particularly difficult.

# Real-Time Optimization of a Continuous Stirred-Tank Reactor

Let us consider the continuous stirred-tank reactor described elsewhere<sup>9,20</sup>.

## Reactor Model

The reactions  $A + B \rightarrow C$  and  $2B \rightarrow D$  take place in an isothermal CSTR, with  $C$  being the desired product. The two manipulated variables are the feed rates of  $A$  and  $B$ . The reactor mass and heat balances lead to the following dynamic model:

$$\dot{c}_A(t) = -k_1 c_A(t) c_B(t) + \frac{u_A(t)}{V} c_{Ain} - \left( \frac{u_A(t) + u_B(t)}{V} \right) c_A(t) \quad (38)$$

$$\dot{c}_B(t) = -k_1 c_A(t) c_B(t) - 2k_2 c_B^2(t) + \frac{u_B(t)}{V} c_{Bin} - \left( \frac{u_A(t) + u_B(t)}{V} \right) c_B(t) \quad (39)$$

$$\dot{c}_C(t) = k_1 c_A(t) c_B(t) - \left( \frac{u_A(t) + u_B(t)}{V} \right) c_C(t) \quad (40)$$

$$\dot{c}_D(t) = k_2 c_B^2(t) - \left( \frac{u_A(t) + u_B(t)}{V} \right) c_D(t) \quad (41)$$

$$Q(t) = V k_1 c_A(t) c_B(t) (-\Delta H_{r,1}) + V k_2 c_B^2(t) (-\Delta H_{r,2}), \quad (42)$$

where  $c_X$  denotes the concentration of species  $X$ ,  $V$  is the reactor volume,  $u_A$  and  $u_B$  are the feed rates of  $A$  and  $B$ ,  $Q$  is the total heat generated,  $\Delta H_{r,1}$  and  $\Delta H_{r,2}$  are the reaction enthalpies of the two reactions,  $c_{Ain}$  and  $c_{Bin}$  are the inlet concentrations, and  $k_1$  and  $k_2$  are the rate constants of the two chemical reactions. The numerical values of the model parameters are given in Table 1.

## Optimization Problem

The objective is to optimize the steady-state performance of the reactor by determining the optimal feed rates of  $A$  and  $B$ .

## Problem formulation

The optimization of steady-state performance is formulated mathematically as follows:

$$\max_{\bar{u}_A, \bar{u}_B} J := \frac{\bar{c}_C^2 (\bar{u}_A + \bar{u}_B)^2}{\bar{u}_A c_{Ain}} - w (\bar{u}_A^2 + \bar{u}_B^2) \quad (43)$$

$$\text{s.t.} \quad -k_1 \bar{c}_A \bar{c}_B + \frac{\bar{u}_A}{V} c_{Ain} - \left( \frac{\bar{u}_A + \bar{u}_B}{V} \right) \bar{c}_A = 0$$

$$-k_1 \bar{c}_A \bar{c}_B - 2 k_2 \bar{c}_B^2 + \frac{\bar{u}_B}{V} c_{Bin} - \left( \frac{\bar{u}_A + \bar{u}_B}{V} \right) \bar{c}_B = 0$$

$$k_1 \bar{c}_A \bar{c}_B - \left( \frac{\bar{u}_A + \bar{u}_B}{V} \right) \bar{c}_C = 0$$

$$k_2 \bar{c}_B^2 - \left( \frac{\bar{u}_A + \bar{u}_B}{V} \right) \bar{c}_D = 0$$

$$\bar{Q} = V k_1 \bar{c}_A \bar{c}_B (-\Delta H_{r,1}) + V k_2 \bar{c}_B^2 (-\Delta H_{r,2})$$

$$G_1 := \frac{\bar{Q}}{Q_{max}} - 1 \leq 0 \quad (44)$$

$$G_2 := \frac{\bar{D}}{D_{max}} - 1 \leq 0 \quad (45)$$

$$0 \leq \bar{u}_A \leq u_{max} \quad (46)$$

$$0 \leq \bar{u}_B \leq u_{max}, \quad (47)$$

where  $\bar{(\cdot)}$  indicates a steady-state value and  $\bar{D} := \frac{\bar{c}_D}{\bar{c}_A + \bar{c}_B + \bar{c}_C + \bar{c}_D}$  is the steady-state molar fraction of the by-product  $D$ . Note that this problem differs from the optimization problems in<sup>9,20</sup> as constraints on the maximal heat generation and the molar fraction  $\bar{D}$  are introduced through the inequalities (44) and (45). The objective function  $J$  represents the productivity of  $C$ , with a penalty for control action by means of  $w(\bar{u}_A^2 + \bar{u}_B^2)$ ,  $w$  being a weighting parameter. The numerical values of the weighting parameter and the bounds are given in Table 2.

Optimization problem (43)-(47) can be reformulated as a minimization problem to fit the form commonly found in the literature, with  $\phi = -J$ . Furthermore, it may be convenient to perform MA with a convex approximation to Problem (43)-(47) to enforce the *adequacy condition*, which is a necessary condition for MA to converge to the plant optimum<sup>6,10</sup>. Hence, Problem (43)-(47)

Table 1: Nominal model parameters

$k_1$	0.75	$\frac{1}{\text{mol min}}$
$k_2$	1.5	$\frac{1}{\text{mol min}}$
$c_{Ain}$	2	$\frac{\text{mol}}{l}$
$c_{Bin}$	1.5	$\frac{\text{mol}}{l}$
$V$	500	l
$(-\Delta H_{r,1})$	3.5	$\frac{\text{kcal}}{\text{mol}}$
$(-\Delta H_{r,2})$	1.5	$\frac{\text{kcal}}{\text{mol}}$

Table 2: Parameters of the optimization problem

$w$	0.004	$\frac{\text{mol min}}{l^2}$
$Q_{max}$	110	kcal
$\bar{D}_{max}$	0.1	—
$u_{max}$	50	$\frac{1}{\text{min}}$

is approximated by the following convex optimization problem:

$$\min_{\mathbf{u}} \quad \phi_c(\mathbf{u}) := \phi^* + [a_\phi \ b_\phi] (\mathbf{u} - \mathbf{u}^*) + \frac{1}{2} (\mathbf{u} - \mathbf{u}^*)^T \mathbf{Q}_\phi (\mathbf{u} - \mathbf{u}^*) \quad (48)$$

$$\text{s.t.} \quad G_{1,c}(\mathbf{u}) := G_1(\mathbf{u}^*) + [a_{G_1} \ b_{G_1}] (\mathbf{u} - \mathbf{u}^*) \leq 0 \quad (49)$$

$$G_{2,c}(\mathbf{u}) := G_2(\mathbf{u}^*) + [a_{G_2} \ b_{G_2}] (\mathbf{u} - \mathbf{u}^*) \leq 0 \quad (50)$$

$$0 \leq \bar{u}_A \leq u_{max} \quad (51)$$

$$0 \leq \bar{u}_B \leq u_{max}, \quad (52)$$

with  $\mathbf{u} = [\bar{u}_A \ \bar{u}_B]^T$  and where  $\phi_c$ ,  $G_{1,c}$  and  $G_{2,c}$  are constructed as convex response-surface approximations to the *model* cost and constraint functions around the nominal steady state corresponding to  $\mathbf{u}_0^*$  by solving three least-squares regression problems for  $\phi_c$ ,  $G_{1,c}$  and  $G_{2,c}$  with the scalars  $a_\phi$ ,  $b_\phi$ ,  $a_{G_1}$ ,  $b_{G_1}$ ,  $a_{G_2}$ ,  $b_{G_2}$  and the  $(2 \times 2)$  matrix  $\mathbf{Q}_\phi$  as degrees of freedom. Note that this model is used for both performing MA and computing the modifiers in Eqns (26)-(29).

Eqns (49) and (50) show that the constraints are approximated by linear functions. This is motivated by the fact that  $G_1$  tends to exhibit a concave behavior in the region of interest, while  $G_2$

is globally neither concave nor convex. Hence, it is simpler to model them as linear functions and enforce strict convexity for the cost function  $\phi_c$ <sup>10</sup>. The diagonal elements of  $\mathbf{Q}_\phi$  are determined (together with  $a_\phi, b_\phi$ ) to force  $\phi_c$  to be strictly convex, with the additional constraints that  $\mathbf{Q}_\phi$  be symmetric and the eigenvalues of  $\mathbf{Q}_\phi$  be greater than user-specified strictly positive values, here chosen both equal to 0.08. Conceptually, only the signs of these eigenvalues matter since strictly positive values guarantee the positive definiteness of  $\mathbf{Q}_\phi$ . In practice, however, these values may affect the convergence rate and should be chosen with care. Note that since  $\phi_c$  is designed to fit the *modeled* cost function  $\phi$ , no experiment is required. The results of the aforementioned constrained least-squares regression are given in Table 3.

Table 3: Parameters of the convex approximations

$a_\phi$	-0.8305	$b_\phi$	-0.9121
$a_{G_1}$	0.0051	$b_{G_1}$	0.0126
$a_{G_2}$	-0.0643	$b_{G_2}$	0.0857
$\mathbf{Q}_\phi =$	0.08	0	
	0	0.08	

### Nominal vs. plant optimum

Both the plant and the model are described by Eqns (38)-(42). Uncertainty in some of the parameters is considered, with the model parameters given in Table 1 and the plant parameters being the same except for  $c_{Ain,p} = 2.5 \frac{\text{mol}}{\text{T}}$ ,  $k_{1,p} = 1.4 \frac{1}{\text{mol min}}$  and  $k_{2,p} = 0.4 \frac{1}{\text{mol min}}$ . Structural uncertainty is not introduced as it will automatically occur upon constructing the convex approximations to  $\phi$ ,  $G_1$  and  $G_2$ .

The optimal solutions for the plant (which is assumed to be unknown and thus will not be used thereafter) and the model are given in Table 4. It is seen that, not only does the parametric uncertainty lead to different optimal input values, but also to a different set of active constraints, with  $G_1$  active and  $G_2$  inactive in the plant, whereas the model predicts the opposite.

Table 4: Solutions to the model and plant optimization problems

Model optimal solution			Plant optimal solution		
$u_A^*$	14.52	$\frac{1}{\text{min}}$	$u_{A,p}^*$	17.20	$\frac{1}{\text{min}}$
$u_B^*$	14.90	$\frac{1}{\text{min}}$	$u_{B,p}^*$	30.30	$\frac{1}{\text{min}}$
$J(\mathbf{u}^*) = -\phi^*$	4.51	$\frac{\text{mol}}{\text{min}}$	$J_p(\mathbf{u}_p^*) = -\phi_p^*$	15.42	$\frac{\text{mol}}{\text{min}}$
$G_1(\mathbf{u}^*) = \frac{Q^*}{Q_{max}} - 1$	-0.48	-	$G_{1,p}(\mathbf{u}_p^*) = \frac{Q_p^*}{Q_{max}} - 1$	0	-
$G_2(\mathbf{u}^*) = \frac{D^*}{D_{max}} - 1$	0	-	$G_{2,p}(\mathbf{u}_p^*) = \frac{D_p^*}{D_{max}} - 1$	-0.19	-

## RTO Using Steady-State Information

Static MA given by Eqns (6)-(11) is implemented with exponential filtering of the modifiers. In the ideal noise-free case with true plant gradients, convergence to the plant optimum is achieved as follows (Figure 4):

- after 4 steady-state iterations when the convex approximation to the plant model is used, with a diagonal gain matrix  $\mathbf{K} = k\mathbf{I}$  tuned to the values where damped oscillations around  $\mathbf{u}_p^*$  start to occur ( $k = 0.8$ ),
- after 6 steady-state iterations when the plant model is used, with a diagonal gain matrix  $\mathbf{K} = k\mathbf{I}$  tuned to the values where damped oscillations around  $\mathbf{u}_p^*$  start to occur ( $k = 0.4$ ).

## RTO Using Transient Information

### MA using transient information and true plant gradients

We start this subsection by illustrating the effect of the choice of the RTO period, denoted  $\tau_{RTO}$ , in the ideal case of true plant gradients. The plant is initialized at the steady state corresponding to the nominal operating point, and MA is performed using a convex approximation to the plant model.

Performing MA every  $\tau_{RTO} = 60$  min, which is a good approximation of the plant settling time, leads to results that are similar to those obtained using static MA (Figure 5). The scheme



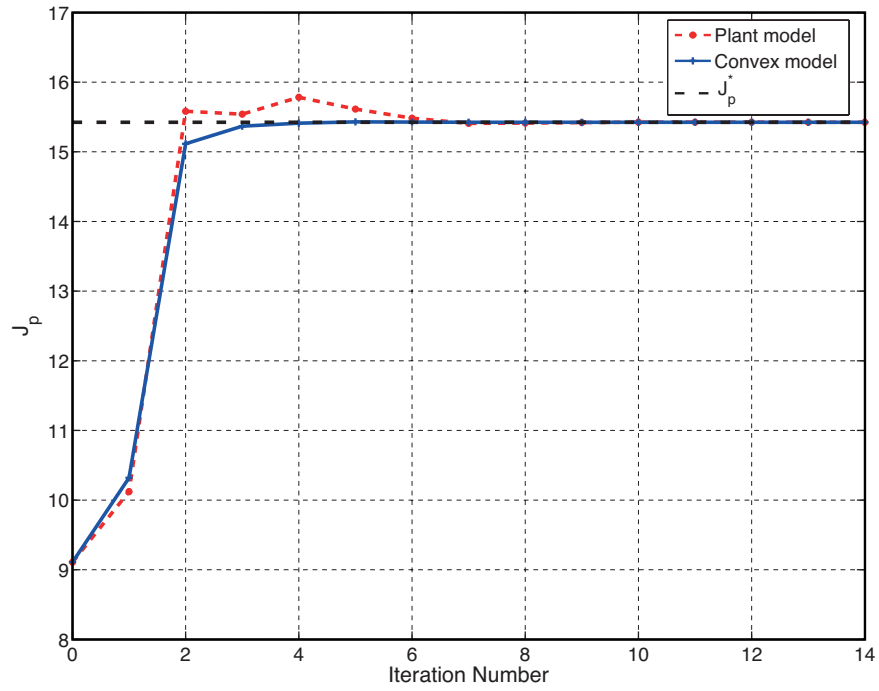


Figure 4: Evolution of the plant cost using static MA with both the plant model and a convex approximation.

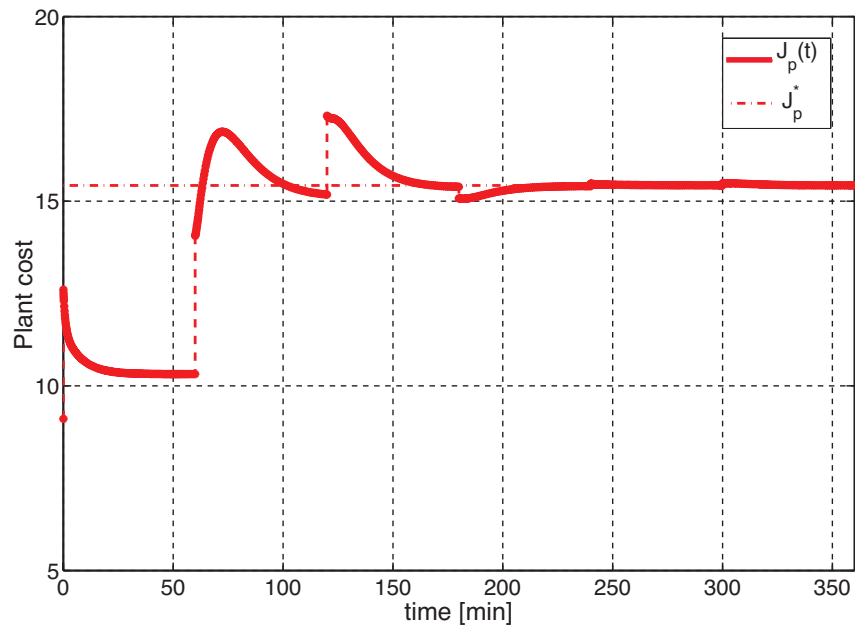


Figure 5: Evolution of the plant cost using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 60$  min and true plant gradients.

takes 4 iterations (or 240 min) to converge to the plant optimum. Note that the cost values during transient can be larger than the optimal cost, which is defined at steady state. Figure 6 depicts

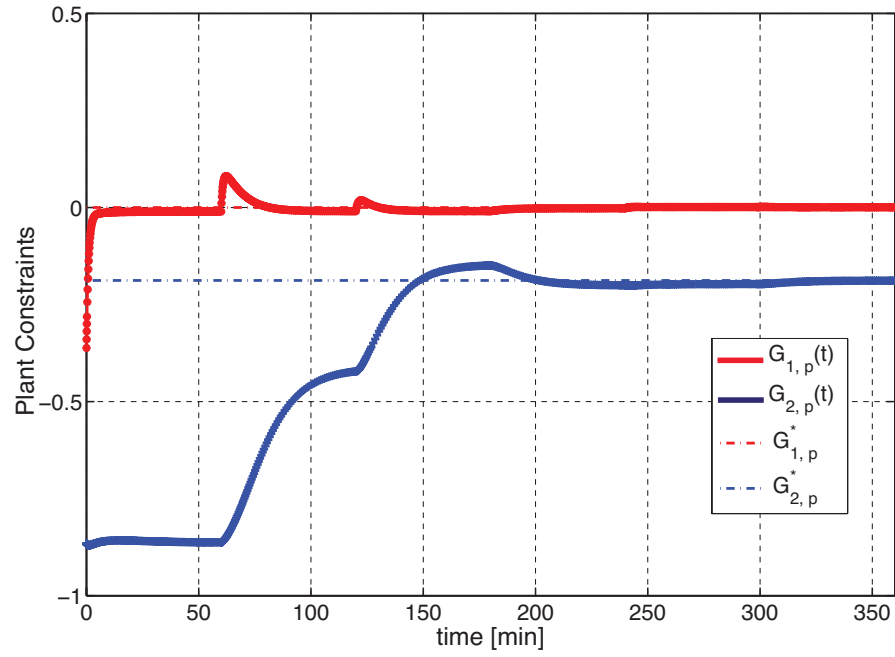


Figure 6: Evolution of the plant constraints using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 60$  min and true plant gradients.

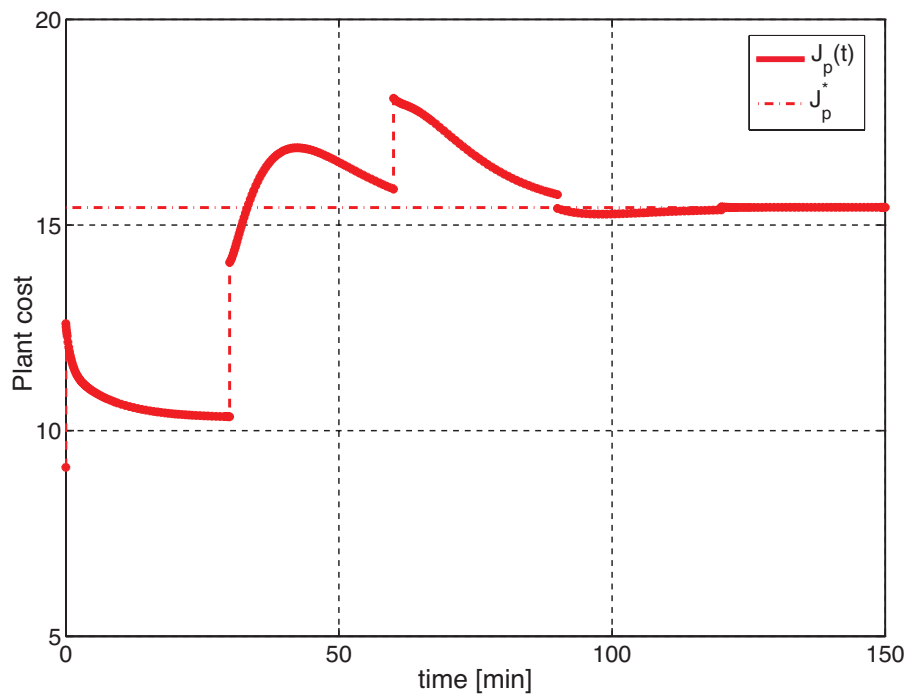


Figure 7: Evolution of the plant cost using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 30$  min and true plant gradients.

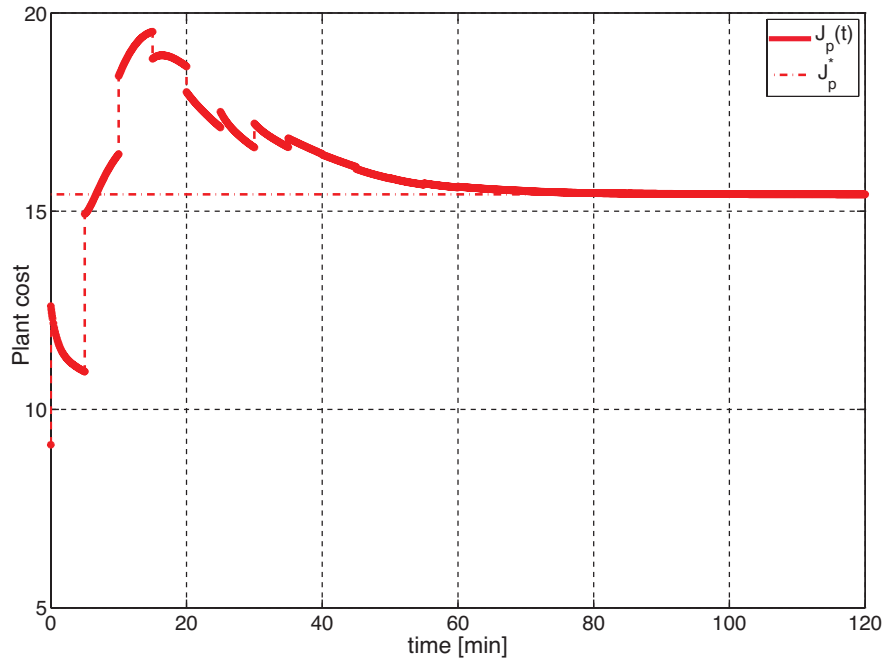


Figure 8: Evolution of the plant cost using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 5$  min and true plant gradients.

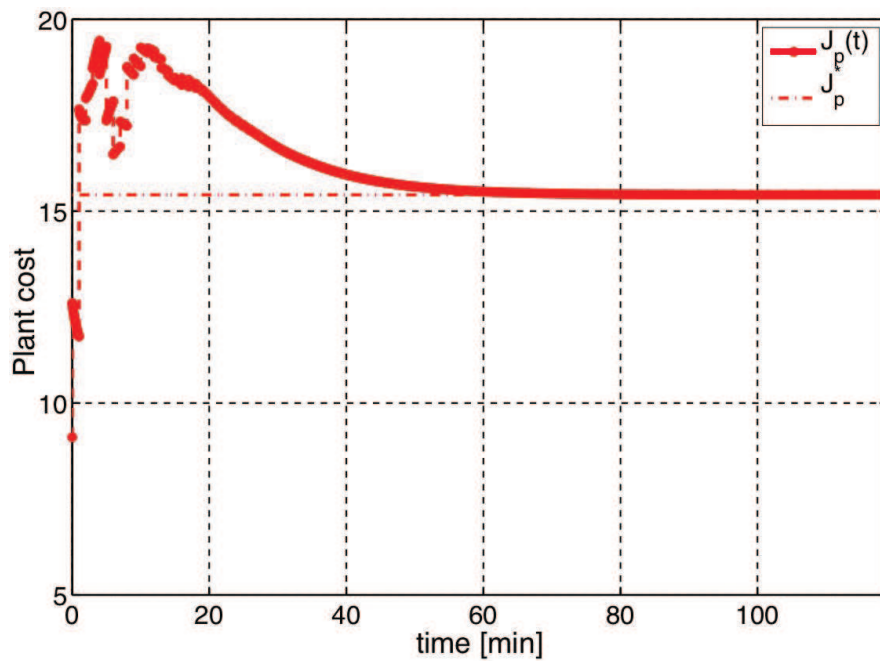


Figure 9: Evolution of the plant cost using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 1$  min and true plant gradients.

the corresponding behavior of the plant constraints and illustrates that the correct set of active constraints is determined.

With  $\tau_{RTO} = 30$  min and  $\tau_{RTO} = 5$  min, the convergence time reduces to 120 min (Figure 7) and 70 min (Figure 8), respectively. Finally, with  $\tau_{RTO} = 1$  min, the convergence time of 60 min is of the order of the plant settling time, which means that the steady-state optimum is reached in the time it takes to reach steady state. The discontinuities observed in the measured cost are due to the abrupt changes in the inputs that occur every time the RTO problem is solved, as illustrated in Figure 10.

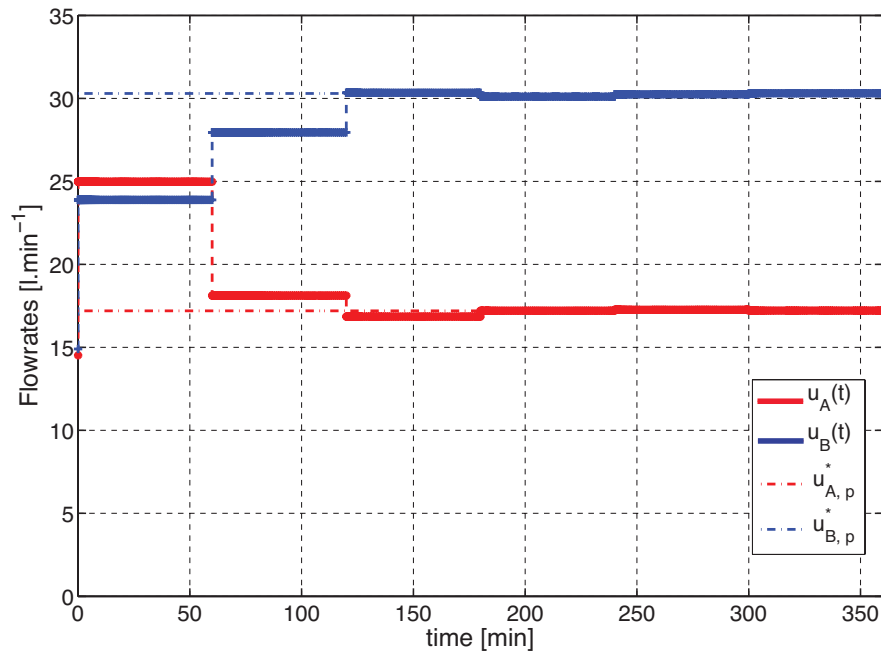


Figure 10: Evolution of the inputs using MA with transient information, a filter gain of 0.8,  $\tau_{RTO} = 60$  min and true plant gradients.

### MA using transient information and NE-based gradients

This section investigates the use of NE-based estimated gradients. We still use the convex approximation to the plant model and Eq. (27) for estimating  $\hat{\varepsilon}_j^G$ , but we push the illustration one step further and use the steady-state plant model given by Eq. (43) to construct the matrices required by Eqns (36)-(37) to estimate the first-order modifiers. We also add 2%-noise to the output measurements, consisting here of the four concentrations. With 4 outputs and 3 uncertain parameters, i.e.  $p > q$ , the conditions for gradient estimation using NE are satisfied. Convergence to the plant

optimum, although theoretically possible, depends on the quality of the linearization-based method for estimating the gradients, as illustrated in the following figures.

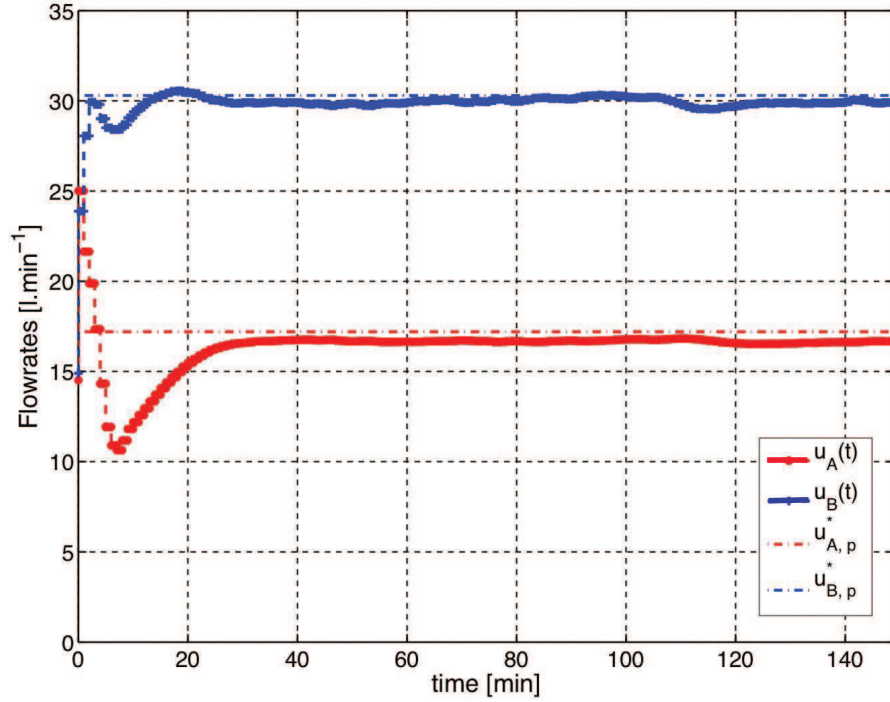


Figure 11: Evolution of the inputs using MA with transient information, a filter gain of 0.6,  $\tau_{RTO} = 1$  min and NE-based gradients.

We restrict ourselves to the case  $\tau_{RTO} = 1$  min and illustrate the effect of the filter gain. Figure 11 depicts the input evolution for a filter gain of 0.6, while Figures 12 and 13 show the corresponding plant cost and constraints. Convergence to the neighborhood of the plant optimum is achieved in 40 min, confirming the results obtained with gradient control<sup>9</sup>, where it was argued that model-based gradient-estimation techniques show the best convergence rates. The small offsets observed between the converged values of the inputs/constraints and the corresponding optimal values are due to the inaccuracy of the linearization, which by the way does not affect much the converged value of the plant cost. Increased filtering leads to slower convergence, a consequence of the optimization problem being “less” modified between successive iterations (Figure 14). Despite the significant amount of filtering, convergence is achieved in about 55 min, which is slightly less than the settling time of the CSTR. Finally, note that the convergence times obtained here are of the same order of magnitude as those observed with gradient control<sup>9</sup>.

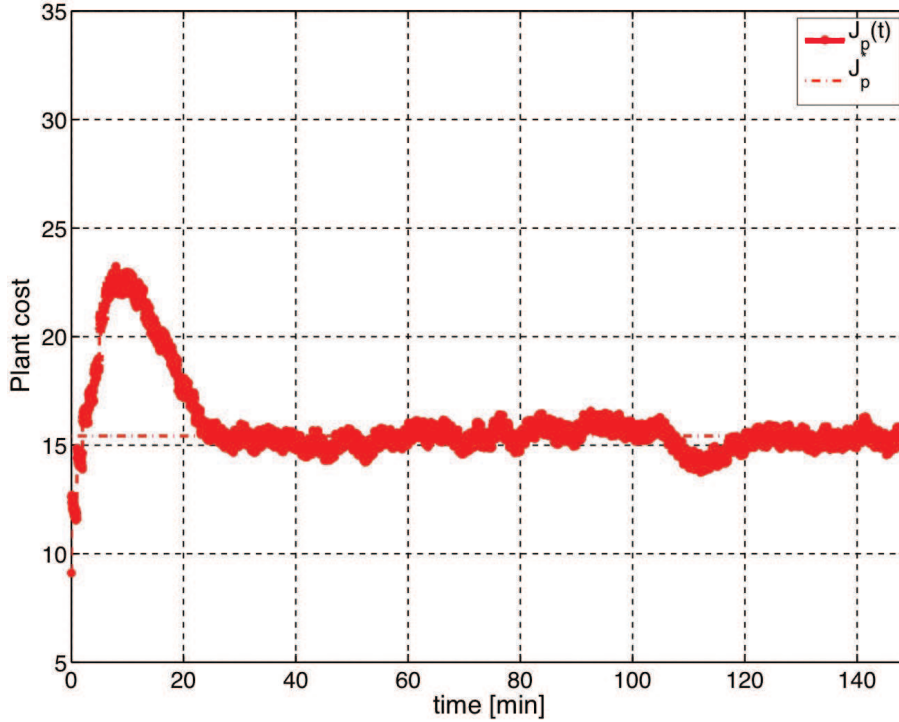


Figure 12: Evolution of the plant cost using MA with transient information, a filter gain of 0.6,  $\tau_{RTO} = 1$  min and NE-based gradients.

### MA using transient information and MU-based gradients

Finally, we proceed in a similar fashion for the case where the gradients are estimated using the MU method. This is the most challenging case since output measurements are not used. No measurement noise is considered to support the case of strictly identical units. This could be interpreted as an intrinsic limitation of the MU method, although there have been recent efforts to overcome this limitation<sup>19,21,22</sup>. Still, MU is interesting in that it is a purely data-driven gradient-estimation method.

Here again,  $\tau_{RTO}$  is set to 1 min, and the filter gain is varied between 0.06 and 0.02. These very low gains, which are justified by the necessity of turning transient measurements into steady-state information, have already been observed in the unconstrained case<sup>9</sup>. Figures 15 - 17 show the evolution of the input profiles for the filter gains 0.06, 0.04 and 0.02, respectively.

As expected, insufficient filtering leads to oscillations, which penalizes the convergence time, while more filtering reduces the aggressivity of the control changes. With “intermediate” filtering, a

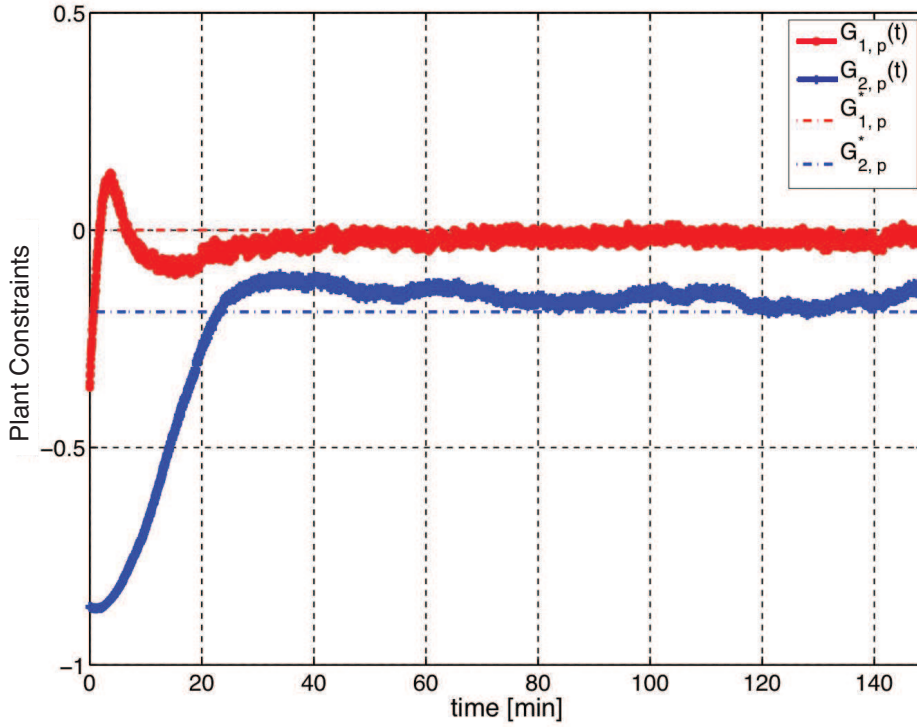


Figure 13: Evolution of the plant constraints using MA with transient information, a filter gain of 0.6,  $\tau_{RTO} = 1$  min and NE-based gradients.

convergence time of about 140 min is observed (Figure 16), while strong filtering slows down the convergence rate and leads to an observed convergence time of 180 min (Figure 17). Interestingly enough, this convergence time of about three times the settling time was also observed in the context of gradient control<sup>9</sup>. With other data-driven gradient-estimation techniques, the convergence time will be much larger, mainly due to the need for time-scale separation<sup>9</sup>. This further justifies the choice of the MU for illustrating the potential performances of MA with transient measurements and data-driven estimates of the gradients since, despite its intrinsic limitations in terms of applicability to real systems, it does not over-penalize the convergence time.

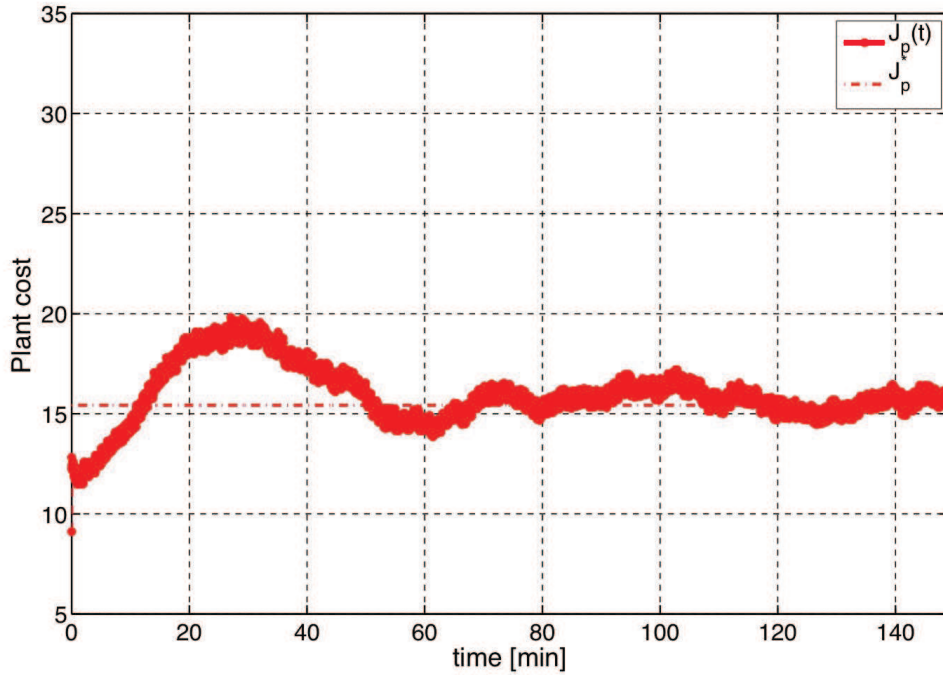


Figure 14: Evolution of the plant cost using MA with transient information, a filter gain of 0.1,  $\tau_{RTO} = 1$  min and NE-based gradients.

## Conclusions

This article has proposed a framework that allows using transient measurements toward the solution of steady-state RTO via MA, similarly to what is done with gradient control in the unconstrained case. Despite plant-model mismatch, convergence to the plant optimum is possible in a single iteration to steady state, provided the plant gradients are estimated accurately. Both a NE-based scheme and an extension of the multiple-unit method have been proposed to estimate the cost and constraint gradients in the presence of parametric uncertainty and constraints. The MA scheme using transient information appears to be quite powerful, whereby the advantages of model-based explicit RTO methods (in particular the possibility of handling plant constraints) can be combined with the advantages of implicit control-inspired methods (in particular fast convergence).

The MA methodology using transient information has been applied to a simulated CSTR in the presence of noise, constraints and a large amount of parametric uncertainty. The time needed for convergence to the plant optimum varies from about two thirds to twice the plant settling time



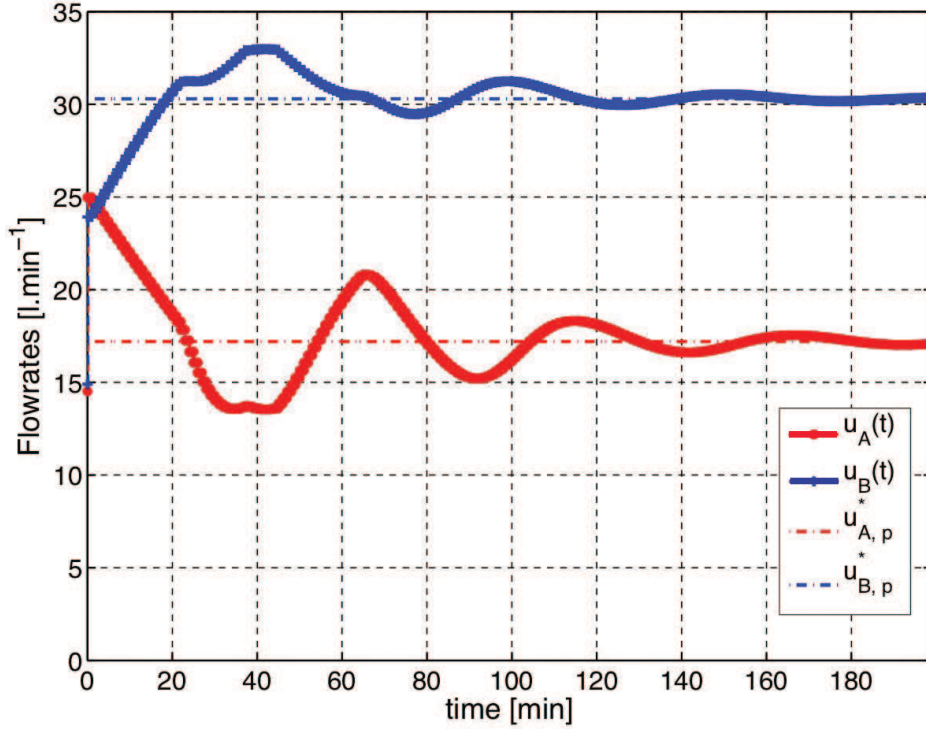


Figure 15: Evolution of the inputs using MA with transient information, a filter gain of 0.06,  $\tau_{RTO} = 1$  min and MU-based gradients.

depending on the gradient-estimation method used, that is, a factor 2-6 reduction compared to static MA under the same uncertain scenario<sup>10</sup>. Note that virtually any gradient-estimation technique can be used to compute the modifiers. A good source of inspiration is to look at gradient-control RTO schemes, which include a more or less explicit gradient-estimation scheme. The choice of a scheme may also depend on the number and nature of the plant measurements (cost and constraints vs. outputs). Although workable schemes have been proposed in this paper, more work is needed to improve the estimation of plant gradients. For example, a regularization-based method developed recently could be tailored to estimate plant gradients<sup>23</sup>.

Although it has recently been suggested to replace the plant model with a convex approximation in the MA problem formulation<sup>10</sup>, the plant model can still be used for estimating the plant gradients. This is what happens with NE, where the gradient estimates result from a variational analysis using the plant model and plant measurements (see Eqns 36-37). Another possible combination is MA for optimization and the classical two-step approach<sup>1</sup> for gradient estimation. The

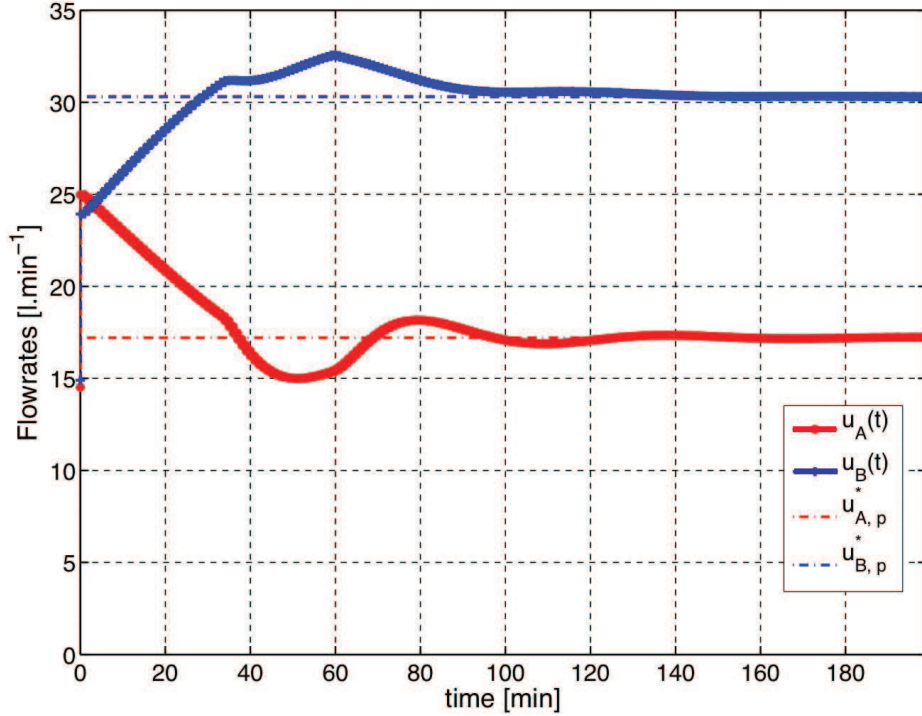


Figure 16: Evolution of the inputs using MA with transient information, a filter gain of 0.04,  $\tau_{RTO} = 1$  min and MU-based gradients.

uncertain model parameters can be updated during transient operation using the differences between measured and predicted outputs. Then, the updated model is used to estimate the cost and constraint gradients at each RTO instant  $t_j$ , while the convex approximation used for MA remains unchanged. We did implement it this way, and the results (not included in this article) are very similar to those of NE, with NE being faster but slightly less accurate. Such an outcome was expected since (i) the convergence time of the two-step approach is penalized by the dynamics of the parameter update and (ii) the accuracy of the NE is penalized by linearization. Again, this confirms the fact that virtually any gradient-estimation method can be used for estimating the modifiers.

Finally, it is important to ensure that the use of static RTO to optimize a dynamic plant does not preclude its stabilization, which calls for a stability analysis. In that context, MA with MU-based gradients has been applied successfully to a simulated CSTR involving unstable internal dynamics (not included here due to space limitation). The price to pay –slower convergence– is not surprising, since the unstable internal dynamics call for either a reduction of the RTO frequency

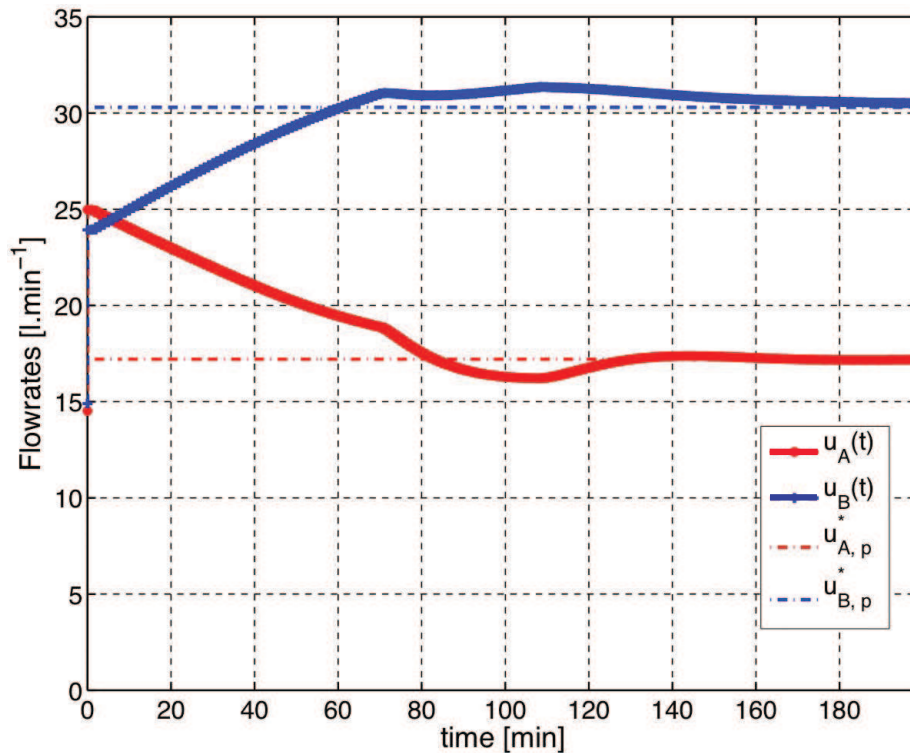


Figure 17: Evolution of the inputs using MA with transient information, a filter gain of 0.02,  $\tau_{RTO} = 1$  min and MU-based gradients.

or strong filtering (or both).

## References

1. Marlin, T.; Hrymak, A. *AIChE Symposium Series - CPC-V*; 1997; Vol. 93; pp 156–164.
2. Roberts, P. D. An algorithm for steady-state system optimization and parameter estimation. *Int. J. Systems Sci.* **1979**, *10*, 719–734.
3. Roberts, P. D.; Williams, T. W. C. On an Algorithm for Combined System Optimization and Parameter Estimation. *Automatica* **1981**, *17*, 199–209.
4. Brdyś, M.; Tatjewski, P. *Iterative Algorithms for Multilayer Optimizing Control*; Imperial College Press, London UK, 2005.

5. Gao, W.; Engell, S. Iterative Set-point Optimization of Batch Chromatography. *Comp. and Chem. Engng.* **2005**, *29*, 1401 – 1409.
6. Marchetti, A.; Chachuat, B.; Bonvin, D. Modifier-adaptation Methodology for Real-time Optimization. *Ind. Eng. Chem. Res.* **2009**, *48*, 6022–6033.
7. Skogestad, S. Plantwide Control: The search for the Self-optimizing Control Structure. *J. Process Contr.* **2000**, *10*, 487–507.
8. Francois, G.; Srinivasan, B.; Bonvin, D. Use of Measurements for Enforcing the Necessary Conditions of Optimality in the Presence of Constraints and Uncertainty. *J. Process Contr.* **2005**, *15*, 701–712.
9. Francois, G.; Srinivasan, B.; Bonvin, D. Comparison of Six Implicit Real-time Optimization Schemes. *J. Européen des Systèmes Automatisés* **2012**, *46*, 291–305.
10. Francois, G.; Bonvin, D. Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation. *Ind. Eng. Chem. Res.* **2013**, *52*, 11614–11625.
11. Srinivasan, B. Real-time Optimization of Dynamic Systems using Multiple Units. *Int. J. Robust Nonlinear Control* **2007**, *17*, 1183–1193.
12. Gros, S.; Srinivasan, B.; Bonvin, D. Optimizing Control Based on Output Feedback. *Comput. Chem. Engng.* **2009**, *33*, 191–198.
13. Alstad, V.; Skogestad, S. Null Space Method for Selecting Optimal Measurement Combinations as Controlled Variables. *Ind. Eng. Chem. Res.* **2007**, *46*, 846–853.
14. Arrow, K.; Hurwicz, L.; Uzawa, H. *Studies in Linear and Nonlinear Programming*; Stanford University Press, 1958.
15. Dürr, H.-B.; Ebenbauer, C. *Preprints of 4th IFAC Nonlinear Model Predictive Control Conference*; 2012; pp 291–298.

16. Ariyur, K.; Krstic, M. *Real-Time Optimization by Extremum-Seeking Control*; John Wiley: New York, 2003.
17. DeHaan, D.; Guay, M. Extremum-Seeking Control of State-Constrained Nonlinear Systems. *Automatica* **2005**, *41*, 1567–1574.
18. Guay, M.; Dochain, D.; Perrier, M. Adaptive Extremum-seeking Control of Nonisothermal Continuous Stirred Tank Reactors. *Chem. Eng. Sci.* **2013**, *60*, 3671–3682.
19. Woodward, L.; Perrier, M.; Srinivasan, B. *Int. Symp. DYCOPS, Leuven, Belgium*; 2010; pp 749–754.
20. Srinivasan, B.; Francois, G.; Bonvin, D. Comparison of Gradient Estimation Methods for Real-time Optimization. *Computer Aided Chemical Engineering* **2011**, *29*, 607–611.
21. Reney, F.; Perrier, M.; Srinivasan, B. *Int. Symp. ADCHEM, Istanbul (Turkey)*; 2009; pp 557–562.
22. Woodward, L.; Perrier, M.; Srinivasan, B. Improved Performance in the Multi-unit Optimization Method with Non-identical Units. *J. Process Contr.* **2009**, *19*, 205–215.
23. Bunin, G.; Francois, G.; Bonvin, D. From Discrete Measurements to Bounded Gradient Estimates: A Look at Some Regularizing Structures. *Ind. and Eng. Chem. Res.* **2013**, *52*, 12500–12513.