Real-Time Optimization when the Plant and the Model have Different Inputs

Sean Costello * **Grégory François** * **Dominique Bonvin** *

* *Ecole Polytechnique Fédérale de Lausanne, Switzerland (sean.costello@epfl.ch, gregory.francois@epfl.ch, dominique.bonvin@epfl.ch).*

Abstract: Model-based optimization is an increasingly popular way of determining the values of the degrees of freedom in a process. The difficulty is that the available model is often inaccurate. Iterative set-point optimization, also known as modifier adaptation, overcomes this obstacle by incorporating process measurements into the optimization framework. We extend this technique to optimization problems where the model inputs do not correspond to the plant inputs. Using the example of an incineration plant, we argue that this occurs in practice when a complex process cannot be fully modeled and the missing part encompasses additional degrees of freedom. This paper shows that the modifier-adaptation scheme can be adapted accordingly. This extension makes modifier adaptation much more flexible and applicable, as a wider class of models can be used. The proposed method is illustrated through a simulated CSTR.

Keywords: Real-time optimization, modifier adaptation, plant-model differences, convergence, optimality

1. INTRODUCTION

Industrial processes are usually designed such that the operator can manipulate a certain number of degrees of freedom (or inputs). Some of these inputs are chosen to ensure that safety and operating constraints are respected, while the remaining ones can be chosen to optimize some performance measure such as profit or product quality. In practice, these decisions are often made online based on operator experience.

Alternatively, these degrees of freedom can be determined in a systematic way, by using process optimization techniques. Two radically different optimization frameworks exist: evolutionary operation and numerical optimization. Evolutionary operation is a structured approach for gradually varying the plant (real process) inputs based on observing the measured response to these variations, similarly to the way an operator would (Box and Draper, 1969). In numerical optimization, powerful computational algorithms are applied to a *model* of the plant to compute optimal values for the inputs. Numerical optimization is more suited to complex and constrained optimization problems than evolutionary optimization, especially when the number of inputs is large. However, if the model does not accurately match the plant, the inputs computed through numerical optimization are neither optimal nor even feasible for the plant.

To face these issues, the field of *measurement-based optimization* has emerged over the last 2-3 decades. With these techniques, measurements are incorporated in the optimization framework to offset the effect of modeling errors and disturbances. In some way, it is a fusion of evolutionary operation and numerical optimization, as the advantages of using process measurements to characterize the plant behavior are combined with the power of numerical optimization and its capability of handling large constrained systems. There are three main ways of incorporating measurements in the optimization framework (Chachuat et al. (2009)): (i) adapt the process model that is used subsequently for optimization as in the so-called two-step approach (Jang et al., 1987), (ii) adapt the optimization problem and repeat the optimization (Tatjewski, 2002; Gao and Engell, 2005; Marchetti et al., 2009), and (iii) directly adapt the inputs through an appropriate feedback strategy (Skogestad, 2000; Srinivasan and Bonvin, 2007)). In this article, we focus on explicit measurement-based optimization techniques of type (ii), for which the optimization problem is repeatedly solved online.

Although the two-step approach is very appealing in that it is a logical way of improving both model accuracy and plant performance, it is widely admitted that the scheme is very unlikely to converge to the true plant optimum in the presence of structural plant-model mismatch (Forbes et al., 1994). With modifier adaptation, measurements are used to implement zeroth- and first-order corrections to the cost and constraint functions, while the process model is kept unchanged. A major advantage of modifier adaptation is that the adequacy conditions (which are necessary conditions for convergence to the *plant* optimum) are much simpler to satisfy than the corresponding conditions of the two-step approach in the case of structural plant-model mismatch (Forbes et al., 1994; Marchetti, 2009). As structural mismatch is almost invariably present in complex processes (i.e. there are always simplifying assumptions made at the modeling stage), this is a very valuable property. However, experimental plant gradients need to be estimated, an onerous task that has received

much attention recently in the literature (Marchetti et al., 2010; Bunin et al., 2013).

Although modifier adaptation has been designed specifically to resolve plant-model mismatch, the model must still satisfy the following conditions:

(1) have the same inputs as the plant,
(2) predict a locally convex (concave) cost function at the plant minimum (maximum).

Condition (2) is likely to be satisfied by any reasonable model. Furthermore, it has recently been shown that it can be enforced by the use of a convex approximation of the process model (François and Bonvin, 2013). The present article proposes a more general modifier-adaptation formulation that can be applied when Condition (1) does not hold, for example when the plant and the model have different inputs.

The paper is organized as follows. After a short review of modifier adaptation in Section 2, the motivating example of an 80-MW incineration plant is presented in Section 3. Section 4 presents the general modifier-adaptation scheme, which is tested in simulation on a continuous stirred-tank reactor in Section 5. Finally, Section 6 concludes the paper.

2. REAL-TIME OPTIMIZATION VIA MODIFIER ADAPTATION

2.1 Problem Formulation

The problem of improving the performances of a process, while meeting certain constraints, can be formulated mathematically as a nonlinear program (NLP):

$$\begin{aligned} \mathbf{u}_p^* := \arg\min_{\mathbf{u}} \; & \phi_p(\mathbf{u}) \\ \text{s.t.} \quad & \mathbf{G}_p(\mathbf{u}) \leq \mathbf{0}, \end{aligned} \tag{2.1}$$

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

In practice, the functions ϕ_p and $\mathbf{G}_p$ are not known, and a plant model is used instead, leading to the following model-based NLP:

$$\begin{aligned} \mathbf{u}^* := \arg\min_{\mathbf{u}} \; & \phi(\mathbf{u}, \boldsymbol{\theta}) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \leq \mathbf{0}, \end{aligned} \tag{2.2}$$

where ϕ and $\mathbf{G}$ represent the models of the cost and constraint functions. These models require the identification of model parameters, here represented by the n_θ-dimensional vector $\boldsymbol{\theta}$. We will assume in this paper that ϕ and $\mathbf{G}$ are differentiable.

If the model matches the plant perfectly, solving Problem (2.2) is sufficient to obtain a solution to Problem (2.1). Unfortunately, this is rarely the case since the structure of the models ϕ and $\mathbf{G}$ as well as the model parameters $\boldsymbol{\theta}$ are likely to be incorrect, which implies that the model-based optimal inputs $\mathbf{u}^*$ will not correspond to $\mathbf{u}_p^*$, the solution to Problem (2.1).

2.2 Real-Time Optimization via Modifier Adaptation

With modifier adaptation, process measurements are used to iteratively modify the model-based Problem (2.2) in such a way that, upon convergence, the necessary conditions of optimality (NCO) of the *modified* optimization problem match those of the plant. This is made possible by using modifiers that, at each iteration, are computed as the differences between the predicted and measured values of the constraints and the predicted and measured cost and constraint gradients.

At the kth iteration, the optimal inputs computed using the modified model are applied to the plant, and the resulting values of the plant constraints and the cost and constraint gradients are compared to the model-based predictions. Then, the following optimization problem is solved to determine the next $\mathbf{u}_{k+1}^*$:

$$\begin{aligned} \mathbf{u}_{k+1}^* := \; & \arg\min_{\mathbf{u}} \; \phi_m(\mathbf{u}) := \phi(\mathbf{u}) + \lambda_k^{\phi^T}(\mathbf{u} - \mathbf{u}_k^*) \\ \text{s.t.} \; & \mathbf{G}_m(\mathbf{u}) := \mathbf{G}(\mathbf{u}) + \epsilon_k^G + \boldsymbol{\lambda}_k^{G^T}(\mathbf{u} - \mathbf{u}_k^*) \leq \mathbf{0} \\ \text{with} \; & \epsilon_k^G := \mathbf{G}_p(\mathbf{u}_k^*) - \mathbf{G}(\mathbf{u}_k^*), \\ & \lambda_k^{\phi^T} := \left.\frac{\partial \phi_p}{\partial \mathbf{u}}\right|_{\mathbf{u}_k^*} - \left.\frac{\partial \phi}{\partial \mathbf{u}}\right|_{\mathbf{u}_k^*}, \\ & \lambda_k^{G^T} := \left.\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}\right|_{\mathbf{u}_k^*} - \left.\frac{\partial \mathbf{G}}{\partial \mathbf{u}}\right|_{\mathbf{u}_k^*}, \end{aligned}$$

where the n_G-dimensional vector ϵ_k^G encompasses the zeroth-order modifiers, and the n_u-dimensional column vector λ_k^ϕ and the $(n_u \times n_G)$ matrix λ_k^G are the first-order modifiers. These gradient terms must be estimated using measurements collected at a number of different operating points close to $\mathbf{u}_k^*$, for example using finite differences, or with more elaborate methods (Marchetti et al., 2010; Bunin et al., 2013). If gradients are available, then it is relatively easy to show that, if the modifier adaptation scheme converges, it will do so to the plant optimum, provided the process model is adequate (Marchetti et al., 2009) or adequacy is enforced by the use of convex model approximations (François and Bonvin, 2013).

3. MOTIVATING EXAMPLE: INCINERATION PLANT

The authors had the experience of working on a practical optimization problem that did not satisfy the conditions for standard modifier adaptation. The plant is the steam cycle of an 80-MW incineration plant, a combined heat and power regenerative Rankine cycle. Energy released by incinerating refuse is used to heat water to 400°C at 50 bar, which drives a turbine to generate electricity. Steam is bled from the turbine at two intermediate stages and passed through heat exchangers that heat water for district heating. A simplified diagram of this complex system is show in Figure 1. The optimization objective is to adjust the pressures, temperatures and mass flowrates of the two intermediate bleeds from the turbine in order to maximize the electrical efficiency for a given district heating demand.

The available system model has the following 5 inputs: the temperature and mass flowrate at point A, T_A and

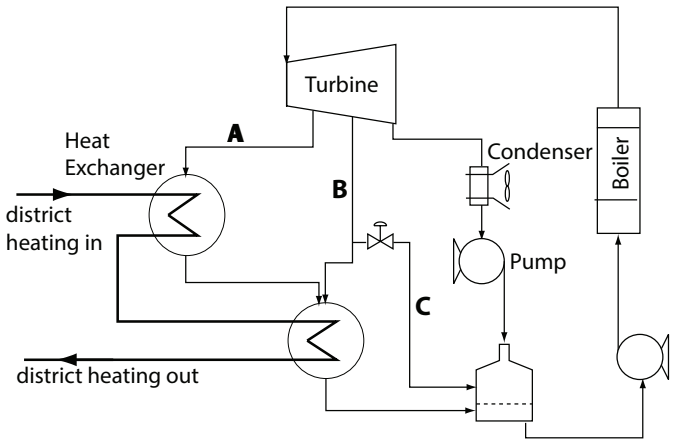

Fig. 1. The steam cycle of the 80-MW incineration plant.

w_A; the temperature and mass flowrate at point B, T_B and w_B; and the mass flowrate at point C, w_C. All the variables in the steam cycle can be calculated from these inputs. These 5 variables were chosen as the model inputs, not necessarily because they correspond to the actual plant inputs, but because they help solve the system equations for this complex cycle. In fact, it was later established that, from the operator's point of view, the plant has only 2 real inputs, the pressure at point A, p_A, and the pressure at point B, p_B. The block diagrams for the model and the plant are shown in Figure 2. The model has more inputs than the plant because certain relationships between variables are not modeled: 1) reliable equations for modeling the steam turbine are not available, and 2) the details of implementation of the control loop that adjusts w_C are not known. As a result, the model is missing three equations, which results in three additional inputs. Furthermore, note that the true plant inputs are not among the model inputs. Although the model is useful for offline numerical optimization to compute the five inputs $\mathbf{u}$, it cannot be used for standard modifier adaptation to compute the three plant inputs $\mathbf{c}$ because Condition (1) described in the Introduction is not satisfied. One option would be to improve the model such that it encompasses the same inputs as the plant, but this would require detailed models of the turbine and the controller for w_C, which unfortunately are not available. The manner in which the model equations are solved would also need to be changed. Hence, it is difficult to reformulate the model such that its inputs $\mathbf{u}$ are the same as those of the plant, $\mathbf{c}$. However, as we will be shown in the next section, re-modeling is not necessary, and modifier adaptation can be generalized such that the model can be used in its current form. This is particularly convenient because measurements (which are in abundance for this system) can be used to compensate for the three missing equations.

4. GENERALIZED MODIFIER ADAPTATION

We now show how the standard modifier-adaptation scheme can be altered when the plant and the model have different inputs. The aim is to avoid remodeling the system. As we will show, this is completely unnecessary!

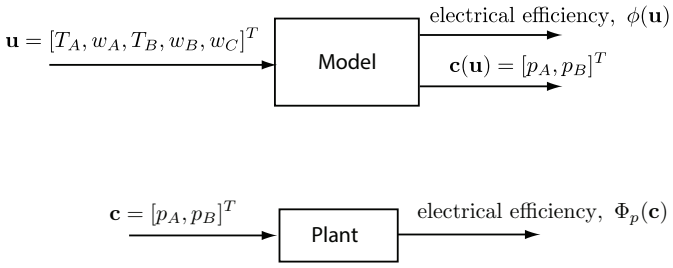

Fig. 2. Model and plant inputs for the incineration plant.

We first present the generalized modifier-adaptation algorithm. Then, a theorem will show that the proposed algorithm can be seen as a specific version of standard modifier adaptation. Finally, a corollary will state that, if the proposed algorithm converges, it will do so to the plant optimum.

4.1 Basic Idea of Generalized Modifier Adaptation

The method described next can be applied in the following context:

(1) The plant cost function $\Phi_p(\mathbf{c})$ depends on the n_c plant inputs $\mathbf{c}$.
(2) The model cost function $\phi(\mathbf{u})$ has n_u inputs $\mathbf{u}$, with $n_u \geq n_c$.
(3) A model $\mathbf{c}(\mathbf{u})$ expressing the mapping from $\mathbf{u}$ to $\mathbf{c}$ is available.

This allows using a model in virtually any form, as long as it has more inputs than the plant. In contrast, standard modifier adaptation can only be applied if $\mathbf{c}$ and $\mathbf{u}$ are the same as indicated by Condition (1) in the Introduction. Although the proposed approach is fairly general, we will only present the unconstrained scenario in the remainder for the sake of simplicity. Future work will treat the constrained case in details.

The generalized modifier-adaptation scheme proceeds as follows. At the kth iteration, the plant inputs $\mathbf{c}_{k+1}$ are obtained by solving the following model-based optimization problem:

Problem 1.

$$\mathbf{u}^*_{k+1} := \underset{\mathbf{u}}{\text{argmin}} \quad \phi(\mathbf{u}) + \boldsymbol{\lambda}_k^T(\mathbf{c}(\mathbf{u}) - \mathbf{c}_k), \tag{4.1}$$

$$\mathbf{c}_{k+1} := \mathbf{c}(\mathbf{u}^*_{k+1}), \tag{4.2}$$

$$\text{with} \quad \boldsymbol{\lambda}_k^T = \frac{\partial \Phi_p}{\partial \mathbf{c}}(\mathbf{c}_k) - \frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u}^*_k)\left(\frac{\partial \mathbf{c}}{\partial \mathbf{u}}(\mathbf{u}^*_k)\right)^+, \tag{4.3}$$

with $(.)^+$ indicating the Moore-Penrose pseudo-inverse. We claim that all fixed points of this iterative procedure satisfy the necessary conditions of optimality for the plant.

4.2 Convergence Analysis

Let us consider standard modifier adaptation with the cost model $\Phi(\mathbf{c})$ in terms of the plant inputs $\mathbf{c}$. The input values $\mathbf{c}_{k+1}$ are obtained by solving the following model-based optimization problem:

Problem 2.

$$\mathbf{c}^*_{k+1} := \underset{\mathbf{c}}{\text{argmin}} \quad \Phi(\mathbf{c}) + \tilde{\boldsymbol{\lambda}}_k^T(\mathbf{c} - \mathbf{c}^*_k), \tag{4.4}$$

$$\text{with} \quad \tilde{\boldsymbol{\lambda}}_k^T = \frac{\partial \Phi_p}{\partial \mathbf{c}}(\mathbf{c}^*_k) - \frac{\partial \Phi}{\partial \mathbf{c}}(\mathbf{c}^*_k). \tag{4.5}$$

Theorem 4.1. [Equivalent problems]
Consider Problems 1 and 2. If the cost model $\Phi(\mathbf{c})$ is chosen as:

$$\begin{aligned} \Phi(\mathbf{c}) := & \underset{\mathbf{u}}{min} \quad \phi(\mathbf{u}) \\ & s.t. \quad \mathbf{G}(\mathbf{u}, \mathbf{c}) := \mathbf{c}(\mathbf{u}) - \mathbf{c} = \mathbf{0}, \end{aligned} \tag{4.6}$$

then Problems 1 and 2 are equivalent.

Proof: We will first show that the non-modified Problems 1 and 2 are equivalent, that is the $\mathbf{c}_{k+1}$ from

$$\mathbf{u}^*_{k+1} := \underset{\mathbf{u}}{\text{argmin}} \quad \phi(\mathbf{u}), \tag{4.7}$$

$$\mathbf{c}_{k+1} := \mathbf{c}(\mathbf{u}^*_{k+1}), \tag{4.8}$$

is equal to

$$\mathbf{c}^*_{k+1} := \underset{\mathbf{c}}{\text{argmin}} \quad \Phi(\mathbf{c}). \tag{4.9}$$

The solution to problem (4.7) is $\mathbf{u}^*_{k+1}$ and the corresponding cost value is $\phi^* = \phi(\mathbf{u}^*_{k+1})$. From (4.6) we see that $\Phi(\mathbf{c}) \geq \phi^* \ \forall \ \mathbf{c}$. Now according to the definition of $\mathbf{c}_{k+1}$ (Equation (4.8)), the constraint in the sub optimization problem (4.6) is satisfied by $\mathbf{c} = \mathbf{c}_{k+1}$ and $\mathbf{u} = \mathbf{u}^*_{k+1}$, and since $\mathbf{u}^*_{k+1}$ is the unconstrained minimizer of ϕ, we have that $\Phi(\mathbf{c}_{k+1}) = \phi(\mathbf{u}^*_{k+1}) = \phi^*$. But as $\Phi(\mathbf{c}) \geq \phi^*$, $\mathbf{c}_{k+1}$ must be the minimizer for $\Phi(\mathbf{c})$, i.e. $\mathbf{c}^*_{k+1} = \mathbf{c}_{k+1}$.

It remains to be shown that the modifier terms, $\boldsymbol{\lambda}_k^T\big(\mathbf{c}(\mathbf{u}) - \mathbf{c}_k\big)$ and $\tilde{\boldsymbol{\lambda}}_k^T(\mathbf{c} - \mathbf{c}_k)$ are the same. Noting that $\mathbf{c}(\mathbf{u}) = \mathbf{c}$ from (4.6), we need to show that the modifiers $\boldsymbol{\lambda}_k$ and $\tilde{\boldsymbol{\lambda}}_k$ are the same. For this, we will apply post-optimal sensitivity analysis to the optimization problem (4.6). It is a standard result that the variation of the optimal inputs with respect to a parameter may be expressed in terms of the derivatives of the cost and constraints of the optimization problem (Fiacco, 1983). The parameter in this case is $\mathbf{c}$. For any parameter value $\mathbf{c}_0$, Fiacco (1983) showed that, under mild conditions (calling for the existence of an optimal solution for $\mathbf{c}_0$ with associated non-zero, unique Lagrange multipliers $\boldsymbol{\mu}^*$), a unique optimal solution function $\mathbf{u}^*(\mathbf{c})$, and a unique optimal cost function $\Phi^*(\mathbf{c})$ exist and are continuously differentiable in the neighborhood of $\mathbf{c}_0$. We will use this result, along with the first-order necessary conditions of optimality. The latter state that the following conditions hold at $\mathbf{u}^*(\mathbf{c}_0)$ and $\mathbf{c}_0$: [1]

$$\frac{\partial \phi}{\partial \mathbf{u}} + \boldsymbol{\mu}^{*^T} \frac{\partial \mathbf{G}}{\partial \mathbf{u}} = \mathbf{0}, \tag{4.10}$$

which implies

$$\boldsymbol{\mu}^{*^T} = -\frac{\partial \phi}{\partial \mathbf{u}} \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \right)^+. \tag{4.11}$$

This is because the Lagrange multipliers express the sensitivity of the cost function to variations in the constraints. Also, in order for the constraint $\mathbf{G}$ to remain satisfied when $\mathbf{c}$ varies, the following must hold:

$$\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{c}} + \frac{\partial \mathbf{G}}{\partial \mathbf{c}} = \mathbf{0}. \tag{4.12}$$

[1] Function arguments are omitted in the following derivation to make it more readable. All functions and partial derivatives are evaluated at $\mathbf{u}^*(\mathbf{c}_0)$ and $\mathbf{c}_0$.

Next, we are interested in calculating

$$\frac{\partial \Phi}{\partial \mathbf{c}} = \frac{\partial \phi}{\partial \mathbf{c}} = \frac{\partial \phi}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{c}}, \tag{4.13}$$

which, using (4.10), can be expressed as

$$\frac{\partial \Phi}{\partial \mathbf{c}} = -\boldsymbol{\mu}^{*^T} \frac{\partial \mathbf{G}}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{c}}. \tag{4.14}$$

And, using (4.11) and (4.12), we finally obtain

$$\frac{\partial \Phi}{\partial \mathbf{c}} = -\frac{\partial \phi}{\partial \mathbf{u}} \left(\frac{\partial \mathbf{G}}{\partial \mathbf{u}} \right)^+ \frac{\partial \mathbf{G}}{\partial \mathbf{c}} \tag{4.15}$$

$$= \frac{\partial \phi}{\partial \mathbf{u}} \left(\frac{\partial \mathbf{c}}{\partial \mathbf{u}} \right)^+. \tag{4.16}$$

Thus, Equations (4.3) and (4.5) are identical, and Problems 1 and 2 are equivalent. ■

Corollary 4.1. [Optimality upon convergence]
If the generalized modifier-adaptation scheme converges, it will do so to a point satisfying the plant first-order necessary conditions of optimality.

Proof: Upon convergence after K iterations, $\mathbf{c}_{K+1} = \mathbf{c}_K$, thereby satisfying the first-order necessary conditions of optimality for Problem (2):

$$\frac{\partial \Phi}{\partial \mathbf{c}}(c_K) + \tilde{\boldsymbol{\lambda}}_K^T = \mathbf{0}. \tag{4.17}$$

On the other hand, (4.5) gives:

$$\frac{\partial \Phi_p}{\partial \mathbf{c}}(\mathbf{c}_K) = \tilde{\boldsymbol{\lambda}}_K^T + \frac{\partial \Phi}{\partial \mathbf{c}}(\mathbf{c}_K), \tag{4.18}$$

which implies

$$\frac{\partial \Phi_p}{\partial \mathbf{c}}(\mathbf{c}_K) = \mathbf{0}. \tag{4.19}$$

Hence, if the scheme converges, it converges to a point satisfying the *plant* necessary conditions of optimality. ■

5. SIMULATED EXAMPLE

The method is illustrated on the Williams-Otto reactor (Williams and Otto, 1960). We will use the version from Roberts (1979), which has become a standard test problem for real-time optimization techniques (Marchetti et al., 2010). The plant (simulated reality) is an ideal continuous stirred-tank reactor with the following reactions:

$$A + B \xrightarrow{k_1} C, \qquad k_1 = k_{10} e^{-E_1/(RT)}, \tag{5.1}$$

$$C + B \xrightarrow{k_2} P + E, \quad k_2 = k_{20} e^{-E_2/(RT)}, \tag{5.2}$$

$$C + P \xrightarrow{k_3} G, \qquad k_3 = k_{30} e^{-E_3/(RT)}, \tag{5.3}$$

where the plant inputs, $\mathbf{c} = [X_A, F_B]^T$, are the mass fraction of A in the reactor and the inlet flowrate of B. An ideal controller adjusts the reactor temperature T to ensure that the value of X_A specified by the plant operator is reached. The inlet flowrate of A is handled by an (assumed unknown) controller to satisfy $F_A = \frac{F_B}{2.4}$. The desired products are P and E and the reactor mass holdup is 2105 kg.

The model is a two-reaction approximation:

$$A + 2B \xrightarrow{k_1^*} P + E, \quad k_1^* = k_{10}^* e^{-E_1^*/(RT)}, \tag{5.4}$$

$$A + B + P \xrightarrow{k_2^*} G, \quad k_2^* = k_{20}^* e^{-E_2^*/(RT)}, \tag{5.5}$$

where k_{10}^* and k_{20}^* are the two model parameters that are adjusted to fit the plant data. The model inputs

$\mathbf{u} = [F_A, F_B, T]^T$ are the flowrates of A and B, and the reactor temperature. The material balance equations for the plant and the model are given in Appendix A. The profit function to be maximized is

$$\begin{aligned}\text{Profit} =& 1143.38 X_P(F_A + F_B) + 25.92 X_E(F_A + F_B) \\ & - 76.23 F_A - 114.34 F_B, \end{aligned} \quad (5.6)$$

where X_P and X_E are the mass fractions of the products P and E. The plant and model cost functions $\Phi_p(\mathbf{c})$ and $\phi(\mathbf{u})$ are the combination of this cost function with the plant and the model, respectively. Block diagrams for the model and the plant are shown in Figure 3. Table 1 gives the numerical values of the plant and model parameters.

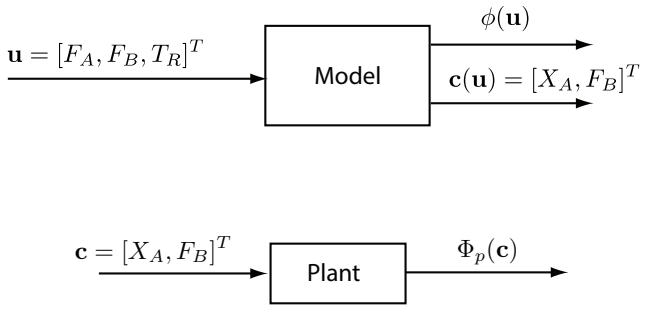

Fig. 3. Comparison of the model and plant inputs for the Williams-Otto reactor.

Table 1. Values of the plant parameters and the two fixed model parameters (the other model parameters are adjusted as shown in Table 2 to generate the investigation cases A-C).

parameter	units	value
k_{10}	s^{-1}	1.660×10^6
k_{20}	s^{-1}	7.212×10^8
k_{30}	s^{-1}	2.675×10^{12}
E_1	$kJmol^{-1}$	5.5427×10^4
E_2	$kJmol^{-1}$	6.9280×10^4
E_3	$kJmol^{-1}$	9.2377×10^4
E_1^*	$kJmol^{-1}$	6.7157×10^4
E_2^*	$kJmol^{-1}$	1.0341×10^5

Generalized modifier adaptation was found to work extremely well on this system, converging to the plant optimum for virtually any degree of plant-model mismatch. One important practical aspect regards the filtering of the modifiers. As in Marchetti et al. (2009), we use a first-order low-pass filter:

$$\begin{aligned}\boldsymbol{\lambda}_k =& (\mathbf{I} - \mathbf{K})\boldsymbol{\lambda}_{k-1} \\ & + \mathbf{K}\left(\frac{\partial \Phi_p}{\partial \mathbf{c}}(\mathbf{c}_k) - \frac{\partial \phi}{\partial \mathbf{u}}(\mathbf{u}_k)\left(\frac{\partial \mathbf{c}}{\partial \mathbf{u}}(\mathbf{u}_k)\right)^+\right)^T. \end{aligned} \quad (5.7)$$

This equation replaces Equation (4.3). The choice of the filter matrix $\mathbf{K}$ is discussed in detail in Marchetti et al. (2009). As can be expected, with more filtering the method is more likely to converge, but it will do so more slowly. In a practical implementation, the filter would need to be tuned manually.

Another key issue in the implementation of this scheme is the evaluation of the plant gradient, which is done via finite differences. At the k^{th} iteration, three different values of $\mathbf{c}$ are applied to the plant, $\mathbf{c}_k$, $\mathbf{c}_k + [\Delta X_A, 0]^T$ and $\mathbf{c}_k + [0, \Delta F_B]^T$, where ΔX_A and ΔF_B are small scalar perturbations. The gradient is then computed as:

$$\left(\frac{\partial \Phi_p}{\partial \mathbf{c}}\right)^T(\mathbf{c}_k) = \begin{bmatrix} \frac{\Phi_p(\mathbf{c}_k) - \Phi_p(\mathbf{c}_k - [\Delta X_A, 0]^T)}{\Delta X_A} \\ \frac{\Phi_p(\mathbf{c}_k) - \Phi_p(\mathbf{c}_k - [0, \Delta F_B]^T)}{\Delta F_B} \end{bmatrix}. \quad (5.8)$$

As gradient estimation is not the focus of this paper, our simulations assume no measurement noise. In practice, the gradient calculation method should be robust to measurement noise. While this is outside the scope of this paper, the interested reader is referred to Marchetti et al. (2010), and Marchetti and Basualdo (2012).

Figures 4 and 5 show the evolution of the plant inputs and the profit for the 3 pairs of adjusted parameters given in Table 2.

Table 2. Values of the adjusted model parameters for the three different cases

Case	k_{10}^* (s^{-1})	k_{20}^* (s^{-1})
A	7900	12500
B	8100	12500
C	8100	12300

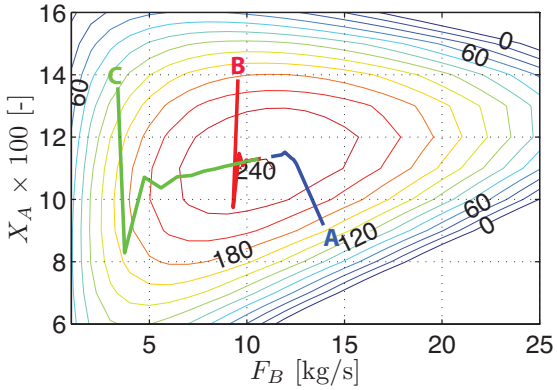

Fig. 4. Evolution of the plant inputs during the first 20 iterations of the generalized modifier-adaptation scheme, for 3 different values of the adjusted model parameters. The contour lines are for the plant cost.

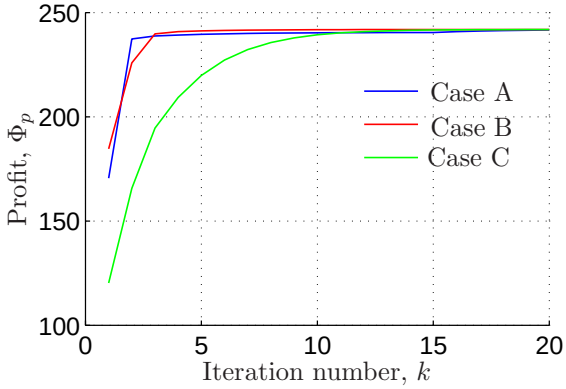

Fig. 5. The profit as a function of the iteration number k. Note that, at each iteration, the plant must be evaluated at 3 slightly different operating points in order to estimate the gradient according to (5.8).

6. CONCLUSION

Optimization via modifier adaptation (also known as iterative set-point optimization) relies on a model of the process. It has typically been assumed that the model and plant inputs are the same. As the motivating example of an incineration plant shows, this will often not be the case. For example, the plant inputs may be controller set points, while the model inputs may correspond to manipulated variables. Reformulating the model such that its inputs and the plant inputs are the same can be extremely difficult if the model is complex (calling for model inversion). In addition, as the model is only an approximation of the plant, its inputs might not include all the plant degrees of freedom, in which case model inversion is generally impossible. Generalized modifier adaptation avoids remodeling the system, at no extra computational cost. This means that a much broader class of optimization problems can be tackled, in particular problems where the plant has an unmodeled control structure. The present work represents a proof of concept and needs to be extended to handle constrained optimization problems.

REFERENCES

Box, G.E. and Draper, N.R. (1969). *Evolutionary Operation: A Statistical Method for Process Improvement*, volume 25. Wiley New York.

Bunin, G.A., Francois, G., and Bonvin, D. (2013). From discrete measurements to bounded gradient estimates: A look at some regularizing structures. *Ind. Eng. Chem. Res. (in press)*. doi: http://dx.doi.org/10.1021/ie303309a.

Chachuat, B., Srinivasan, B., and Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Comp. Chem. Eng.*, 33(10), 1557–1567.

Fiacco, A.V. (1983). *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Academic Press.

Forbes, J., Marlin, T., and MacGregor, J. (1994). Model adequacy requirements for optimizing plant operations. *Comp. Chem. Eng.*, 18(6), 497–510.

François, G. and Bonvin, D. (2013). Use of convex model approximations for real-time optimization via modifier adaptation. *Ind. Eng. Chem. Res. (submitted)*.

Gao, W. and Engell, S. (2005). Iterative set-point optimization of batch chromatography. *Comp. Chem. Eng.*, 29(6), 1401–1409.

Jang, S.S., Joseph, B., and Mukai, H. (1987). On-line optimization of constrained multivariable chemical processes. *AIChE J.*, 33(1), 26–35.

Marchetti, A., Chachuat, B., and Bonvin, D. (2010). A dual modifier-adaptation approach for real-time optimization. *J. Process Contr.*, 20(9), 1027–1037.

Marchetti, A. and Basualdo, M. (2012). A new dual modifier-adaptation approach for real-time optimization with inaccurate models. In *22nd European Symp. on Computer Aided Process Engineering*, volume 30, 942. Elsevier.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.*, 48(13), 6022–6033.

Marchetti, A.G. (2009). *Modifier-Adaptation Methodology for Real-Time Optimization*. Ph.D. thesis, # 4449, EPFL, Lausanne.

Roberts, P. (1979). An algorithm for steady-state system optimization and parameter estimation. *Int. J. Systems Sci.*, 10(7), 719–734.

Skogestad, S. (2000). Plantwide control: The search for the self-optimizing control structure. *J. Process Contr.*, 10, 487–507.

Srinivasan, B. and Bonvin, D. (2007). Real-time optimization of batch processes by tracking the necessary conditions of optimality. *Ind. Eng. Chem. Res.*, 46(2), 492–504.

Tatjewski, P. (2002). Iterative optimizing set-point control - The basic principle redesigned. In *Proc IFAC World Congress*. Barcelona.

Williams, T.J. and Otto, R.E. (1960). A generalized chemical processing model for the investigation of computer control. *Trans. of the American Inst. of Elec. Engineers, Part I: Communication and Electronics*, 79(5), 458–473.

Zhang, Y. and Forbes, J. (2000). Extended design cost: A performance criterion for real-time optimization systems. *Comp. Chem. Eng.*, 24(8), 1829–1841.

Appendix A. MATERIAL BALANCES FOR THE CSTR

The simulated reality (plant) is governed by (Marchetti, 2009; Zhang and Forbes, 2000):

$$0 = F_A - (F_A + F_B)X_A - Wr_1, \tag{A.1}$$

$$0 = F_B - (F_A + F_B)X_B - \frac{M_B}{M_A}Wr_1 - Wr_2, \tag{A.2}$$

$$0 = -(F_A + F_B)X_C + \frac{M_C}{M_A}Wr_1 - \frac{M_C}{M_B}Wr_2 - Wr3, \tag{A.3}$$

$$0 = -(F_A + F_B)X_P + \frac{M_P}{M_B}Wr_2 - \frac{M_P}{M_C}Wr_3, \tag{A.4}$$

$$0 = -(F_A + F_B)X_G + \frac{M_G}{M_C}Wr_3, \tag{A.5}$$

$$X_E = \frac{M_E}{M_P}X_P + \frac{M_E}{M_G}X_G, \tag{A.6}$$

with

$$r_1 = k_1 X_A X_B, \tag{A.7}$$

$$r_2 = k_2 X_B X_C, \tag{A.8}$$

$$r_3 = k_3 X_C X_P. \tag{A.9}$$

The model equations are:

$$0 = F_A - (F_A + F_B)X_A - Wr_1 - Wr_2, \tag{A.10}$$

$$0 = F_B - (F_A + F_B)X_B - \frac{M_B}{M_A}2Wr_1 - \frac{M_B}{M_A}Wr_2, \tag{A.11}$$

$$0 = -(F_A + F_B)X_P + \frac{M_P}{M_A}Wr_1 - \frac{M_P}{M_A}Wr_2, \tag{A.12}$$

$$0 = -(F_A + F_B)X_E + \frac{M_E}{M_A}Wr_1, \tag{A.13}$$

$$X_G = \frac{M_G}{M_E}X_E + \frac{M_G}{M_P}X_P, \tag{A.14}$$

with

$$r_1 = k_1 X_A X_B^2, \tag{A.15}$$

$$r_2 = k_2 X_A X_B X_P. \tag{A.16}$$

By assuming $M_A = M_B = M_P$, all the molecular weight ratios X_i are defined from the stoichiometry of the reactions.