Use of Convex Model Approximations for Real-Time Optimization 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) via modifier adaptation is a class of methods for which measurements are used to iteratively adapt the model via input-affine additive terms. The modifier terms correspond to the deviations between the measured and predicted constraints on the one hand, and the measured and predicted cost and constraint gradients on the other. If the iterative scheme converges, these modifier terms guarantee that the converged point satisfies the KKT conditions *for the plant*. Furthermore, if upon convergence the plant model predicts the correct curvature of the cost function, convergence to a (local) plant optimum is guaranteed. The main advantage of modifier adaptation lies in the fact that these properties do not rely on specific assumptions regarding the nature of the uncertainty. In other words, in addition to rejecting the effect of parametric uncertainty like most RTO methods, modifier adaptation can also handle process disturbances and structural plant-model mismatch. This paper shows that the use of a convex model approximation in the modifier-adaptation framework implicitly enforces model adequacy. The approach is illustrated through both a simple numerical example and a simulated continuous stirred-tank reactor.

*To whom correspondence should be addressed

Keywords : Real-time optimization, Modifier adaptation, Convex optimization, Plant-model mismatch, Model adequacy.

Introduction

The two main decisions that affect process performance are the design of the production facility and the choice of the operating conditions. Computing the set of operating conditions that both meet the operating constraints and optimize performance is the main goal of process optimization. Long considered an academic exercise applicable only to lab-scale setups, process optimization has now become a sound and economically viable technology[1,2].

Schematically, process optimization consists of three steps: (1) *modeling*, where a plant model is built from, for example, first principles, (2) *problem formulation*, where the cost and constraint functions are defined, and (3) *solution*, where an appropriate numerical method is used to compute the optimal inputs. Note that these inputs, which are optimal for the model at hand, will lead to plant optimality only if the model is a perfect representation of the plant. This is typically not the case in the presence of uncertainty in the form of parametric uncertainty, process disturbances or structural plant-model mismatch. Hence, and since optimizing a system implies the satisfaction of both feasibility and sensitivity conditions, the application of these inputs to the plant will generally lead to the violation of some plant constraints and/or sub-optimal performance.

To face these issues, the field of Real-Time Optimization (RTO) has emerged in the last 20 years. The general idea of RTO techniques has been to use plant measurements to somehow improve the model-based 'optimal" inputs. The inability of the classical model-based optimization techniques to determine the plant optimum can be seen as the inability of the model to predict the necessary conditions of optimality (NCO) of the plant. Hence, RTO techniques are designed to either directly or indirectly correct this prediction. Plant measurements can be used at any of the three main steps of the optimization procedure, and the RTO techniques are classified accordingly as described next[3]:

1. The deviation between the predicted and measured outputs can be used to refine the model parameters. This approach is referred to as the‘two-step approach” of repeated parameter estimation and performance optimization. It has become an industrial standard[4], although it is widely accepted that the two-step approach can only work well if the structure of the model is adequate[5,6] and reliable parameter identification can be achieved with the input changes provided by the optimization scheme. Also, note that two optimization tasks are performed at each iteration, and it has been argued that the two optimization objectives do not necessarily work hand in hand[7].

2. There are still a significant number of difficulties that limit the wide application of RTO in industry[8,9]. The main limitation is without any doubt the inability of the RTO schemes to find the plant optimum in the presence of structural plant-model mismatch[5,10]. This can be overcome in principle if measurements are used to modify the NCO of the optimization problem so as to make them match those of the plant. With the "integrated system optimization and parameter estimation (ISOPE) method"[11,12], a gradient modification term is added to the cost function of the optimization problem. Also, the modifier-adaptation approach[10,13] proposes to perform this correction via the addition of modifier terms to both the cost and constraint functions. It was shown in[10] that this enforces convergence to a KKT point of the plant, that is, a point satisfying the Karush-Kuhn-Tucker necessary conditions of optimality. In addition, if model-adequacy conditions are satisfied[5], necessary conditions for convergence can be established[10] and the converged inputs represent a local extremum for the plant. These adequacy conditions correspond to the ability of the plant model to predict the correct curvature of the cost function in the vicinity of the converged point, and are indeed a small subset of the corresponding adequacy conditions for the standard two-step approach. However, since the adequacy conditions are defined at the unknown plant optimum, they cannot be checked a priori. Another difficulty raised in[8] addresses the numerical re-optimization, for which, in most cases, deterministic methods such as sequential quadratic programming are typically used, while it is widely known (though less reported in the lit-

erature) that these numerical algorithms can fail to converge[14], which is unacceptable for industrial applications[15]. Finally, note that the modifier-adaptation scheme relies on a *single* optimization per iteration, since the modifiers are calculated as the (filtered) differences between the plant and model KKT elements, that is, not through optimization.

3. Alternatively, it is also possible to use plant measurements to directly adjust the inputs optimally. In this case, the plant model and the optimization are only used off-line to compute the *nominal* optimal solution that is used to design the optimizing control scheme. The inputs are adjusted on-line using control laws that enforce the NCO, that is, without on-line numerical optimization. These so-called self-optimizing approaches[16–18] are also referred to as "implicit schemes", as the optimization problem is recast as a control problem, while the schemes that involve repeating the optimization on-line fall in the category of "explicit schemes".

To ease the issues of model adequacy and numerical re-optimization, this paper proposes to use convex approximations of the cost and constraints rather than the plant model itself, even when the model at hand is nonconvex. This is possible in the context of modifier adaptation since the (possibly inaccurate) cost and constraint functions are corrected using plant measurements. This correction being input-affine, the convexity property is preserved. Since modifier adaptation is able to deal with structurally incorrect models, replacing the plant model by another model is clearly not a critical issue. This paper shows that this modification implicitly enforces the adequacy conditions at the *unknown* plant optimum and thus eases convergence to a KKT point of the plant. In addition, since the modified optimization problem is now formulated as a convex program, it is possible to use convex solvers that are very efficient for large-scale systems.

This paper is organized as follows. Section II describes real-time optimization via modifier adaptation. Section III proposes to use convex approximations and discusses the resulting benefits. Section IV illustrates the proposed approach via two simulated examples, while Section V concludes the paper.

Real-Time Optimization via Modifier Adaptation

Formulation of the Optimization Problem

Process optimization aims at improving the performance of a given plant. In addition to the optimization of a cost function, it is generally necessary to meet certain plant constraints. All this can be formulated mathematically as a nonlinear program (NLP):

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

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.

Note that this formulation encompasses different plant optimization problems:

1. Static optimization problems,
2. Steady-state optimization problems, for which the cost and constraints functions are defined at the steady state of some dynamical process,
3. Dynamic optimization problems that are reformulated as general NLPs, e.g. by parameterization of both the input and state profiles and reformulation of the path constraints as point-wise constraints[19],
4. Run-to-run dynamic optimization of batch processes, where the input profiles are parametrized and the cost and constraints are defined at final time. In this case, since the input parameters are typically chosen before the run starts and the cost and constraints are evaluated at final time, the dynamics get lumped into a static map as shown in[17], similarly to what happens with the direct sequential methods[20].

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

$$\begin{aligned} \mathbf{u}^* \quad := \quad & \arg\min_{\mathbf{u}} \phi\left(\mathbf{u}, \theta\right) \qquad \text{(Problem 2)} \\ & \text{s.t.} \quad \mathbf{G}\left(\mathbf{u}, \theta\right) \leq \mathbf{0}\,, \end{aligned} \tag{2}$$

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 θ. We will assume in this paper that ϕ and $\mathbf{G}$ are differentiable.

If the model matches the plant perfectly, solving Problem 2 is sufficient to obtain a solution to Problem 1. Unfortunately, this is rarely the case since the model parameters θ and the structure of the models ϕ and $\mathbf{G}$ are likely to be incorrect, which in turn implies that the model-based optimal inputs $\mathbf{u}^*$ will not correspond to $\mathbf{u}_p^*$, the solution to Problem 1. Real-time optimization is a family of methods for which plant measurements are used to update $\mathbf{u}^*$ in order to determine $\mathbf{u}_p^*$.

Necessary Conditions of Optimality

Modifier adaptation proposes to use plant measurements to iteratively modify the cost and constraint functions in Problem 2. We will make the classical assumption in nonlinear programming that constraint qualification (such as linear independence of the constraints around the optimal solution) holds, which ensures that the KKT conditions are indeed first-order necessary conditions of optimality[21]. Since Problems 1 and 2 differ, their NCO will also differ. For Problem 1, these NCO are:

$$\begin{aligned} \mathbf{G}_p\left(\mathbf{u}_p^*\right) \quad &\leq \quad \mathbf{0} \\ \nabla\phi_p\left(\mathbf{u}_p^*\right) + (\nu_p^*)^T \nabla\mathbf{G}_p\left(\mathbf{u}_p^*\right) \quad &= \quad \mathbf{0} \\ \nu_p^* \quad &\geq \quad \mathbf{0} \\ (\nu_p^*)^T \mathbf{G}_p\left(\mathbf{u}_p^*\right) \quad &= \quad 0\,, \end{aligned} \tag{3}$$

while, for Problem 2, the NCO read:

$$\begin{aligned} \mathbf{G}(\mathbf{u}^*, \theta) &\leq \mathbf{0} \\ \nabla\phi(\mathbf{u}^*, \theta) + (\nu^*)^T \nabla\mathbf{G}(\mathbf{u}^*, \theta) &= \mathbf{0} \\ \nu^* &\geq \mathbf{0} \\ (\nu^*)^T \mathbf{G}(\mathbf{u}^*, \theta) &= 0, \end{aligned} \tag{4}$$

where the superscript $(^*)$ denotes optimality, ν and ν_p are the n_G-dimensional vectors of Lagrange multipliers associated with the constraints of the model-based NLP and the plant NLP, respectively.

As already mentioned, if $\mathbf{u}_p^*$ is a local minimum of Problem 1 and a regular point for the constraints, then Conditions (3) have to hold. If, in addition, the functions ϕ_p and $\mathbf{G}_p$ are convex, these conditions are sufficient for $\mathbf{u}_p^*$ to be the global minimizer of the plant optimization Problem 1. The same can be said about Conditions (4) and the model-based optimization Problem 2. Note that the convex assumptions are not always verified for the model-based problem and impossible to verify for the plant optimization problem.

Modifier-Adaptation Scheme

The main idea behind modifier adaptation is to use plant measurements and iteratively modify the model-based Problem 2 in such a way that, upon convergence, the NCO of the *modified* optimization 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 the predicted and measured cost and constraint gradients. These modifiers are used to both shift the modeled constraints and adjust the slope of the modeled cost and constraint functions by the addition of input-affine corrections, according to the intuitive observation that first-order corrections are required to achieve matched first-order optimality conditions.

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 of the cost and constraint gradients are compared to the model predictions. Then, the following optimization problem is solved to determine the next $\mathbf{u}_{k+1}^*$:

$$\begin{aligned}
\mathbf{u}_{k+1}^* \quad &:= \quad \arg\min_{\mathbf{u}} \phi_m(\mathbf{u}) := \phi(\mathbf{u}) + \varepsilon_k^{\phi} + \lambda_k^{\phi}(\mathbf{u} - \mathbf{u}_k^*) && (5)\\
\text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}) := \mathbf{G}(\mathbf{u}) + \varepsilon_k^{G} + \lambda_k^{G}(\mathbf{u} - \mathbf{u}_k^*) \leq \mathbf{0} && (6)\\
\text{with} \quad & \varepsilon_k^{\phi} := \phi_p(\mathbf{u}_k^*) - \phi(\mathbf{u}_k^*) && (7)\\
& \varepsilon_k^{G} := \mathbf{G}_p(\mathbf{u}_k^*) - \mathbf{G}(\mathbf{u}_k^*) && (8)\\
& \lambda_k^{\phi} := \nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi(\mathbf{u}_k^*) && (9)\\
& \lambda_k^{G} := \nabla\mathbf{G}_p(\mathbf{u}_k^*) - \nabla\mathbf{G}(\mathbf{u}_k^*), && (10)
\end{aligned}$$

where the scalar ε_k^{ϕ} and the n_G-dimensional vector ε_k^{G} are the zeroth-order modifiers, and the n_u-dimensional row vector λ_k^{ϕ} and the $n_G \times n_u$ matrix λ_k^{G} represent the first-order modifiers.

The main advantage of the modifier-adaptation scheme described by Eqns (5)-(10) lies in its ability to converge to a KKT point of the plant[10]. Eqns (6) and (8) show that, upon convergence at $\mathbf{u}_\infty^*$, one has:

$$\mathbf{G}_m(\mathbf{u}_\infty^*) = \mathbf{G}(\mathbf{u}_\infty^*) + \varepsilon_\infty^{G} = \mathbf{G}_p(\mathbf{u}_\infty^*) \leq \mathbf{0}. \tag{11}$$

Hence, the zeroth-order modifier terms ε_k^{G} allow enforcing the feasibility conditions $\mathbf{G}_p(\mathbf{u}_\infty^*) \leq \mathbf{0}$ upon convergence. Note that the correction term ε_k^{ϕ} simply shifts the cost function up or down, without changing the location of its minimum, and thus is generally discarded.

Similarly, upon differentiating Eqns (5) and (6) with respect to $\mathbf{u}$ and using Eqns (9) and (10), one obtains upon convergence:

$$\begin{aligned}
\nabla\phi_m(\mathbf{u}_\infty^*) \quad &= \quad \nabla\phi(\mathbf{u}_\infty^*) + \lambda_\infty^{\phi} = \nabla\phi_p(\mathbf{u}_\infty^*) && (12)\\
\nabla\mathbf{G}_m(\mathbf{u}_\infty^*) \quad &= \quad \nabla\mathbf{G}(\mathbf{u}_\infty^*) + \lambda_\infty^{G} = \nabla\mathbf{G}_p(\mathbf{u}_\infty^*). && (13)
\end{aligned}$$

Hence, the first-order correction terms in the cost and constraint functions (with slopes λ_k^{ϕ} and λ_k^{G})

modify the model gradients to make them match the corresponding plant gradients.

Remarks: Several remarks are in order:

1. The zeroth- and first-order corrections lead to matched Lagrange multipliers upon convergence, thus ensuring the correct set of active constraints. Hence, modifier adaptation forces the NCO of the model-based optimization Problem 2 to match those of the plant optimization Problem 1.

2. In the absence of constraint qualification, modifier adaptation still forces the KKT conditions of the model-based optimization Problem 2 to match those of the plant optimization Problem 1, which, however, are no longer NCO.

3. Denoting by $\phi_{err}(\mathbf{u}) := \phi_p(\mathbf{u}) - \phi(\mathbf{u})$ and $\mathbf{G}_{err}(\mathbf{u}) := \mathbf{G}_p(\mathbf{u}) - \mathbf{G}(\mathbf{u})$ the modeling errors of the cost and constraints functions, it is straightforward to show that the modifier terms correspond to the parameters of a first-order Taylor series expansion of $\phi_{err}(\mathbf{u})$ and $\mathbf{G}_{err}(\mathbf{u})$ around $\mathbf{u}_k^*$:

$$\begin{aligned}
\phi_{err}(\mathbf{u}) &\simeq \phi_{err}(\mathbf{u}_k^*) + \nabla\phi_{err}(\mathbf{u}_k^*)(\mathbf{u} - \mathbf{u}_k^*) \\
&\simeq \phi_p(\mathbf{u}_k^*) - \phi(\mathbf{u}_k^*) + \Big(\nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi(\mathbf{u}_k^*)\Big)(\mathbf{u} - \mathbf{u}_k^*) \\
&\simeq \varepsilon_k^{\phi} + \lambda_k^{\phi}(\mathbf{u} - \mathbf{u}_k^*) .
\end{aligned} \tag{14}$$

And similarly:

$$\mathbf{G}_{err}(\mathbf{u}) \simeq \varepsilon_k^{G} + \lambda_k^{G}(\mathbf{u} - \mathbf{u}_k^*) . \tag{15}$$

This indicates that modifier adaptation can also be seen as a method that uses measurements to update the plant model, similarly to what is performed in the two-step approach. However, the model update is performed here in the form of input-affine corrections and not as parametric adjustments. Note that these corrections are performed directly at the NCO level (via the cost and constraint functions), that is, with the optimization objective in mind along the line of "modeling for optimization"[7,22].

4. Modifier adaptation requires the gradients of the plant cost $\nabla\phi_p(\mathbf{u}_k^*)$ and constraints $\nabla\mathbf{G}_p(\mathbf{u}_k^*)$ to be known. The estimation of plant gradients, or more generally speaking of an unknown function, is a difficult ill-posed inverse problem [23]. Several techniques for obtaining gradient estimates are available in the literature, which mainly differ on whether or not a plant model is used. A brief overview of these techniques is given in the next subsection.

5. The modifiers in Eqns (7)-(10) are computed as the differences between measured and predicted KKT elements. Exponential filtering of the modifiers is often implemented in order to enforce convergence in the presence of uncertainty [10]. Introducing the notation,

$$\Lambda_k = \begin{bmatrix} \phi_p(\mathbf{u}_k^*) - \phi(\mathbf{u}_k^*) \\ \mathbf{G}_p\left(\mathbf{u}_k^*\right) - \mathbf{G}\left(\mathbf{u}_k^*\right) \\ \left(\nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi(\mathbf{u}_k^*)\right)^T \\ \left(\nabla\mathbf{G}_p(\mathbf{u}_k^*) - \nabla\mathbf{G}(\mathbf{u}_k^*)\right)^T \end{bmatrix}, \tag{16}$$

the filtered modifiers are computed as:

$$\Lambda_{k+1} = \Gamma\,\Lambda_k + (\mathbf{I} - \Gamma)\,\Lambda_{k-1}\,, \tag{17}$$

with the gain matrix Γ.

Gradient Estimation

Finite differences is probably the most straightforward approach for estimating the gradients of the plant cost $\nabla\phi_p(\mathbf{u}_k^*)$ and constraints $\nabla\mathbf{G}_p(\mathbf{u}_k^*)$ on the basis of discrete measurements, that is, the plant is presented with slightly different inputs and the gradients are computed as differences. This is however unlikely to be an adequate approach with a large number of inputs. Furthermore, for steady-state optimization problems, the performances of such schemes are worsen by the fact that, for each perturbation of the inputs, one has to wait for steady state before the subsequent RTO iteration can be performed. In addition, the presence of measurement noise is detrimental to the quality of the gradient estimates since finite differences typically amplifies the effect of noise. To overcome these limitations, a novel regularization-based technique has recently been proposed to estimate the gradients of a unknown function and, at the same time, obtain bounds on the estimates[24].

When a plant model is available, model-based techniques can also be used, keeping in mind that the use of an inaccurate model leads to inaccurate gradient estimates. For instance, it is possible to use transient measurements to identify dynamic models whose static gains correspond to the gradients at steady state[25]. Again, the performance of such an approach will generally decrease when the number of inputs and plant constraints increases. Alternatively, gradient estimates can be obtained by using (i) a variational analysis of the modeled cost and constraint functions to obtain the gradients in terms of the inputs and outputs, and (ii) plant measurements to compute the gradient values at the current iteration[26]. This approach can lead to perfect gradient estimates provided the model uncertainty is of parametric nature and the identity of the uncertain parameters is known.

Recently, the authors have compared several model-based and data-driven gradient estimation methods that are typically used in the context of RTO with gradient control[27]. As a general trend, it has been confirmed that data-driven techniques have the potential of being more accurate when unexpected disturbances occur, while model-based techniques converge faster but are more sensitive to unknown disturbances and plant-model mismatch.

If the gradient estimates are not accurate, modifier adaptation converges to a point that is suboptimal for the plant. Indeed, since gradients are also required for determining the set of active constraints, the use of inaccurate gradient estimates leads potentially to a wrong set of active constraints. On the other hand, the plant constraints will be satisfied upon convergence since the zeroth-order modifiers ε_k^G, which enforce the matching of the primal feasibility parts of the NCO, do not rely on the plant gradients. Fortunately, it has been observed that a large part of suboptimality induced by the use of inaccurate models can be discarded[27].

The authors have also recently proposed a set of sufficient conditions for feasibility and optimality[28], which are indeed a framework that can guide *any RTO scheme* to converge to the plant optimum. These conditions can be met even when the gradient estimates are inaccurate[29], provided they can be bounded, which is possible with the method described in[24].

Model Adequacy

For convergence to a (local) minimum, an additional condition has to be satisfied. This so-called model-adequacy condition requires to define the reduced Hessian for Problem 2.

Definition 1 (*Reduced Hessian*)

Consider the model-based optimization Problem 2. Let us assume that, at the optimum ($\mathbf{u}^*$, ν^*), n_a inequality constraints of $\mathbf{G}$ are active. Denoting by $\mathbf{G}^a$ and $\nu^{*,a}$ the active constraints and the corresponding Lagrange multipliers, the null space of the Jacobian of the active constraints is defined from the relation:

$$\nabla \mathbf{G}^a(\mathbf{u}^*)\,\mathbf{Z} \;=\; \mathbf{0}, \tag{18}$$

where the columns of the $n_u \times (n_u - n_a)$ matrix $\mathbf{Z}$ represent a set of basis vectors for the null space of the active constraints. The $(n_u - n_a)$-dimensional reduced gradient and the $(n_u - n_a) \times (n_u - n_a)$ reduced Hessian of the cost function are given by[30]:

$$\nabla_r \phi\,(\mathbf{u}^*, \theta) \;:=\; \nabla \phi\,(\mathbf{u}^*, \theta)\,\mathbf{Z} \tag{19}$$

$$\nabla_r^2 \phi\,(\mathbf{u}^*, \theta) \;:=\; \mathbf{Z}^T \nabla^2 L\,(\mathbf{u}^*, \theta)\,\mathbf{Z}, \tag{20}$$

where $L(\mathbf{u}) := \phi(\mathbf{u}) + (\nu^{*,a})^T \mathbf{G}^a(\mathbf{u})$ is the restricted Lagrangian function associated with Problem 2.

We can now state the model-adequacy condition for the modifier-adaptation scheme.

Definition 2 (*Model adequacy*)

Let $\mathbf{u}_p^*$ be a local optimum for the plant (that is, a solution to Problem 1) and a regular point for the *constraints $\mathbf{G}_p$ (that is, a point at which the constraint qualification "linear independence" holds), with the associated optimal values of the Lagrange multipliers ν_p^*. If $\nabla_r^2 \phi\left(\mathbf{u}_p^*, \theta\right) > \mathbf{0}$, then the plant model is said to be adequate for use with the modifier-adaptation scheme (5)-(10).*

If the model is adequate and the modifier-adaptation scheme (5)-(10) converges, then convergence will be to the local minimum $\mathbf{u}_p^*$ as described by Criterion 2 in [10]. The main advantage of the modifier-adaptation scheme lies in the fact that the model-adequacy condition is much less restrictive than that for the two-step approach [10]. Yet, the model-adequacy condition cannot be verified a priori without knowledge of $\mathbf{u}_p^*$ and the set of active constraints (for determining the reduced Hessian). We will show later how the use of convex models can overcome this limitation.

Modifier Adaptation using Convex Model Approximations

Basic Idea

The previous section has shown the capability of modifier adaptation to detect a (local) plant minimum despite the absence of an accurate model. This is done via measurement-based correction terms that help the inaccurate model predict correctly the plant NCO. Since the only requirement for the model to be adequate is that the reduced Hessian be positive definite, this section proposes to take advantage of this relative freedom by choosing the convex model approximations $\phi_c(\mathbf{u})$ and $\mathbf{G}_c(\mathbf{u})$. The errors resulting from these approximations will be incorporated in the linear terms $\phi_{err}(\mathbf{u})$ or $\mathbf{G}_{err}(\mathbf{u})$ and treated as modeling errors.

With the convex cost and constraint models, the modifier-adaptation scheme (5)-(10) becomes:

$$\mathbf{u}_{k+1}^* \quad := \quad \text{argmin} \;\; \phi_m(\mathbf{u}) = \phi_c(\mathbf{u}) + \varepsilon_k^\phi + \lambda_k^\phi(\mathbf{u} - \mathbf{u}_k^*) \tag{21}$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u}) := \mathbf{G}_c(\mathbf{u}) + \varepsilon_k^G + \lambda_k^G(\mathbf{u} - \mathbf{u}_k^*) \leq \mathbf{0} \tag{22}$$

$$\text{with} \quad \varepsilon_k^\phi := \phi_p(\mathbf{u}_k^*) - \phi_c(\mathbf{u}_k^*) \tag{23}$$

$$\varepsilon_k^G := \mathbf{G}_p(\mathbf{u}_k^*) - \mathbf{G}_c(\mathbf{u}_k^*) \tag{24}$$

$$\lambda_k^\phi := \nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi_c(\mathbf{u}_k^*) \tag{25}$$

$$\lambda_k^G := \nabla\mathbf{G}_p(\mathbf{u}_k^*) - \nabla\mathbf{G}_c(\mathbf{u}_k^*)\,. \tag{26}$$

Adequate Model for Modifier Adaptation

The advantage of using convex cost and constraint functions are stated in the following theorem and the associated corollary.

Theorem 1

Consider the plant optimization Problem 1 with the optimal inputs and Lagrange multipliers $\mathbf{u}_p^*$ *and* ν_p^**. Let the the cost and the constraints be modeled by the strictly convex functions* ϕ_c *and* $\mathbf{G}_c$*. Then, the following properties hold:*

1. *The modified optimization problem (21)-(26) is a convex program for all* k*.*

2. *The model-adequacy condition of Definition 2 is verified.*

Proof: The proof of the first property is straightforward. As the modification of the cost and constraint functions is affine in the inputs, ϕ_m and $\mathbf{G}_m$ share the same second-order derivatives as ϕ_c and $\mathbf{G}_c$ and thus are convex functions for all k.

We now prove the second property. Since $\mathbf{u}_p^*$ is the (unknown) solution to Problem 1, the (also unknown) corresponding Lagrange multipliers ν_p^* are non-negative. Also, since the constraints $\mathbf{G}_c$ are strictly convex functions, the (unknown) active constraints $\mathbf{G}_c^a$ are also strictly convex. As the linear combination of strictly convex functions preserve strict convexity when the coefficients are non-negative, the restricted Lagrangian function $L_c(\mathbf{u}) := \phi_c(\mathbf{u}) + (\nu_p^{*,a})^T\, \mathbf{G}_c^a(\mathbf{u})$ is a strictly convex function. The fact that $\nabla_r^2 \phi\left(\mathbf{u}_p^*,\, \theta\right)$ is positive definite can be inferred from the positive definiteness of the Hessian of L_c. Indeed, for $\nabla_r^2 \phi\left(\mathbf{u}_p^*,\, \theta\right)$ to be positive definite, $\mathbf{x}^T \nabla_r^2 \phi\left(\mathbf{u}_p^*,\, \theta\right) \mathbf{x}$ has to be positive for all non-zero $(n_u - n_a)$-dimensional vectors $\mathbf{x}$. Hence, with Eq. (20), $\mathbf{x}^T \nabla_r^2 \phi\left(\mathbf{u}_p^*,\, \theta\right) \mathbf{x} = (\mathbf{Z}\mathbf{x})^T\, \nabla^2 L_c(\mathbf{u}_p^*)\, (\mathbf{Z}\mathbf{x})$. It follows from the positive definiteness of $\nabla^2 L_c$ that $\mathbf{z}^T \nabla^2 L_c \mathbf{z} > \mathbf{0}$ for all n_u-dimensional non-zero vectors $\mathbf{z}$. This also holds for any non-zero vector $\mathbf{z}$ of the form $\mathbf{z} = \mathbf{Z}\,\mathbf{x}$. Hence, the fact that the Hessian matrix of L is positive definite enforces the model-adequacy condition (20). □

Remarks

1. The main message of this theorem is that, by using strictly convex cost and constraint models in modifier adaptation, model adequacy is guaranteed *without requiring prior knowledge of the plant optimum*, which is important for implementation.

2. Furthermore, by using convex cost and constraint models, one generates a sequence of convex programs that are known to be faster and potentially easier to solve than general NLPs.

3. If convex models are adequate for minimization problems as the curvature around the optimal solution will always be predicted correctly, a concave approximation of the cost will have to be used for maximization problems.

4. Model adequacy is inherited from the strict convexity of the cost and constraint functions. Note, however, that strict convexity is not needed for all functions, as stated in the following corollary.

Corollary 1

Consider the plant optimization Problem 1 with the optimal inputs and Lagrange multipliers $\mathbf{u}_p^*$ and ν_p^* and the modifier-adaptation scheme (5)-(10). Let $\mathbf{G}^{s,a}$ denote the $n_{s,a}$ strongly active constraints with $\nu_p^{*s,a}$ the corresponding Lagrange multipliers such that $\mathbf{G}^{s,a}(\mathbf{u}_p^*) = \mathbf{0}$ and $\nu_p^{*s,a} > \mathbf{0}$, and $\mathbf{G}^{w,a}$ the $n_a - n_{s,a}$ weakly active constraints with $\nu_p^{*w,a}$ the corresponding Lagrange multipliers such that

$\mathbf{G}^{w,a}(\mathbf{u}_p^*) = \boldsymbol{0}$ *and* $\nu_p^{*w,a} = \boldsymbol{0}$*. For Theorem 1 to be applicable, it is sufficient that either (i) the cost function be strictly convex with the active constraints* $\mathbf{G}^a$ *being convex functions, or (ii) at least one of the strongly active constraints be strictly convex with the cost and the other active constraints being convex functions.*

Proof: The key assumption in Theorem 1 is the strict convexity of the restricted Lagrangian function $L_c(\mathbf{u})$. We show next that, under the assumption that at least one of the strictly active constraints or the cost function be strictly convex, convexity of the remaining functions is sufficient to enforce strict convexity of L_c. For this purpose, we form $L_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big)$, with θ being a real scalar between 0 and 1 and $\mathbf{u}_1$ and $\mathbf{u}_2$ any two input vectors:

$$L_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) \;=\; \phi_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) + (\nu_p^{*a})^T\mathbf{G}_c^a\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big). \quad (27)$$

From the definition of weakly and strongly active constraints, it follows that the Lagrangian (and its curvature) is only affected by ϕ and by the strictly active constraints $\mathbf{G}_c^{s,a}$, which allows reducing Eq. 27 to:

$$L_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) \;=\; \phi_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) + (\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big). \quad (28)$$

Since all functions are convex, the terms on the right-hand side can be bounded as follows:

$$\phi_c\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) \;\leq\; \theta\phi_c(\mathbf{u}_1) + (1-\theta)\,\phi_c(\mathbf{u}_2) \quad (29)$$

$$(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}\Big(\theta\mathbf{u}_1 + (1-\theta)\,\mathbf{u}_2\Big) \;\leq\; \theta(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}(\mathbf{u}_1) + (1-\theta)\,(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}(\mathbf{u}_2)\,. \quad (30)$$

If either ϕ or one of the constraints $\mathbf{G}_c^{s,a}$ is a strictly convex function, then either Eq. (29) or (30) will be a strict inequality. The sum of Eqns (29) and (30) will also be a strict inequality, and combining it with Eq. (27) gives:

$$
\begin{aligned}
L_c\Big(\theta\mathbf{u}_1+(1-\theta)\,\mathbf{u}_2\Big) \;&<\; \theta\phi_c\left(\mathbf{u}_1\right)+\left(1-\theta\right)\phi_c\left(\mathbf{u}_2\right)\left(\mathbf{u}_1\right) \\
&\quad +\theta(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}+\left(1-\theta\right)(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}\left(\mathbf{u}_2\right) \\
&<\; \theta\left(\phi_c\left(\mathbf{u}_1\right)+(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}\left(\mathbf{u}_1\right)\right) \\
&\quad +\left(1-\theta\right)\left(\phi_c\left(\mathbf{u}_2\right)+(\nu_p^{*s,a})^T\mathbf{G}_c^{s,a}\left(\mathbf{u}_2\right)\right) \\
&<\; \theta L_c\left(\mathbf{u}_1\right)+\left(1-\theta\right)L_c\left(\mathbf{u}_2\right),
\end{aligned}
\tag{31}
$$

which shows that L_c is a strictly convex function. □

Remarks

1. The main interest in approximating the cost and the constraints by convex functions is to enforce the model-adequacy condition without prior knowledge of the plant optimum. If this property can be verified by having one strictly convex strongly active constraint (according to Condition (ii) above), this requires the knowledge of the set of active constraints at the optimum. Hence, although interesting by itself, Condition (ii) is generally speaking of little practical value. Note, however, that it can be useful in some cases. For instance, in the run-to-run minimization of batch time for polymerization reactors, one of the constraints is on final conversion[17,31]. Hence, it is obvious that this particular constraint will be strongly active, as any batch with a conversion higher than the minimal requirement (and thus an inactive constraint on conversion) will last longer and thus be suboptimal. Also the corresponding optimal Lagrange multiplier (which can be interpreted as the sensitivity of the cost function - here the batch time - to a change in the constraint on conversion) will be non-zero, as asking for lower conversion will lead to shorter batches (and vice-versa).

2. Hence, apart from very specific cases, it is probably easier to enforce Condition (i) by building a strictly convex approximation of the cost function. Then, the main message of Corollary 1 is that a strictly convex cost function, combined with convex constraints, will repre-

sent an adequate model for modifier adaptation. In particular, a strictly convex cost function, combined with linear approximations of the constraints, will be adequate. This is particularly interesting for cases where a model constraint exhibits a concave shape, for which a linear approximation will be preferred.

Advantages of Convex Models

The use of (strictly) convex cost and constraint functions has several advantages:

- The model-adequacy condition will always be satisfied.
- The sequence of general NLPs is replaced by a sequence of convex programs, as the use of modifier terms preserves the strict convexity of the restricted Lagrangian.
- The use of convex approximations does not preclude the intrinsic ability of the modifier-adaptation scheme to reach, upon converge, a KKT point of the plant[8].
- Fast convex solvers can be used. Although more iterations might be required since the convex approximations typically decrease the accuracy of the model predictions, these iterations can be performed much faster.
- The success rate of modern convex solvers is much higher than that of SQP solvers and can virtually reach 100%,[32] which is a very nice feature for industrial RTO applications[8].

Illustrative Examples

This section illustrates via two examples the benefits of using adequate models in the form of convex approximations of inadequate models. Since the goal of this paper is not to discuss gradient estimation methods, we will assume the availability of perfect gradient estimates.

Numerical Example

A simple unconstrained optimization problem is chosen to illustrate the effect of model inadequacy on the convergence of the modifier-adaptation scheme. More specifically, this example will show that enforcing the cost function to be strictly convex is sufficient to meet the adequacy condition and guarantee the convergence of the modifier-adaptation scheme to a KKT point of the plant despite the use of a simplified inaccurate model.

The plant optimization problem with the scalar input u reads:

$$\min_u \quad \phi_p(u) := p_4\, u^4 + p_3\, u^3 + p_2\, u^2 + p_1\, u + p_0 \tag{32}$$

$$-5 \leq u \leq 5\,, \tag{33}$$

with $p_4 = 0.0172$, $p_3 = 0.0082$, $p_2 = -0.2927$, $p_1 = -0.3699$ and $p_0 = 2.3856$ being the unknown plant parameters.

Modifier adaptation with inadequate model

Let us assume that a linearized model of the plant is available around the operating point u_0, that is, $\phi(u) = \phi_p(u_0) + \nabla\phi_p(u_0)\,(u - u_0)$. As the input box constraints are assumed to be known with certainty and since shifting the predicted value of the cost function up or down does not change the location of the optimal input, the modifiers ε_k^{ϕ}, ε_k^{G} and λ_k^{G} are not needed, and the modifier-adaptation scheme of Eqns (5)-(10) reduces to:

$$u_{k+1}^* \quad := \quad \text{argmin}\ \ \phi_m\,(u) = \phi\,(u) + \lambda_k^{\phi}\,(u - u_k^*) \tag{34}$$

$$\text{s.t.} \quad -5 \leq u \leq 5 \tag{35}$$

$$\text{with} \quad \lambda_k^{\phi} = \nabla\phi_p(u_k^*) - \nabla\phi\,(u_k^*)\,. \tag{36}$$

A linear program is solved at each iteration since the linear model $\phi(u)$ is corrected by the linear term $\lambda_k^{\phi}\,\left(u - u_k^*\right)$. Note that a linear model is convex but not strictly convex, that is, not necessarily adequate for modifier adaptation. It turns out that the solution of this sequence of linear programs will lie on the boundaries of the feasible domain and will jump from one bound to the other when the correction term changes the sign of the slope of ϕ_m.

Modifier adaptation with adequate model

Next, we modify $\phi(u)$ slightly to make it strictly convex, and thus adequate, by augmenting it with αu^2 as suggested in[12]. If $\alpha > 0$, $\phi_c(u) = \phi(u) + \alpha u^2$ is a strictly convex function everywhere in $[-5 \;\; 5]$, that is, also at the unknown plant optimum u_p^*.

Simulation results

We consider the three different initialization points $u_0^a = -0.61$, $u_0^b = 4.5$ and $u_0^c = -4.5$. The parameter α is set to 0.5 and modifier adaptation is implemented without input filtering. Figures 1, 2 and 3 depict the evolution of the input and the cost function for both the inadequate and the convexified models for the three initialization points, respectively.

Remarks: Some remarks are in order:

- In all three cases, convergence cannot be achieved with the inadequate model as the input jumps from one bound to the other.
- The use of an adequate model makes convergence to a local minimum possible. As modifier adaptation is by nature a local method, the converged KKT point will depend on the domain of attraction associated with the initialization point. For instance, with Case c, convergence to a local minimum is observed, while for Cases a and b, the global optimum can be reached.
- The convergence rate for Case a is low as it is penalized by the fact that the initial input is chosen very close to a local maximum, that is, with an initial plant gradient close to zero. Nevertheless, the algorithm is able to leave the region of the local maximum and converge to the global solution.

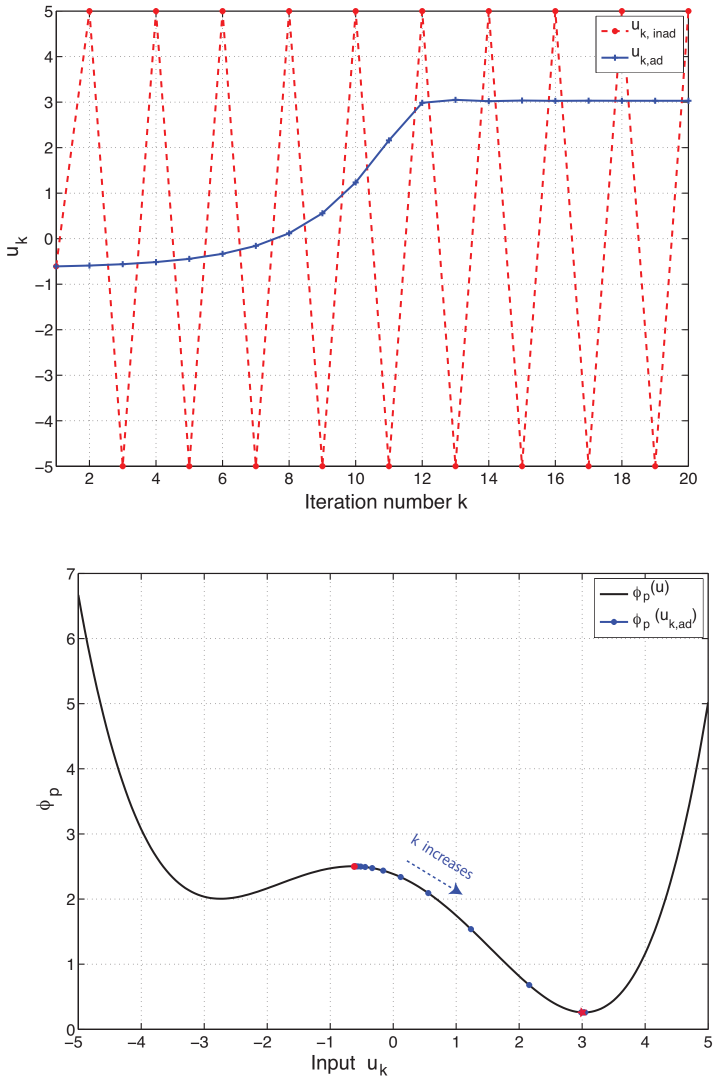


Figure 1: Case a. Top: evolution of the input with the inadequate (red) and the adequate (blue) models. Bottom: plant cost function (black) and successive plant cost values obtained with the adequate model (blue).

Steady-State Optimization of a Continuous Stirred-Tank Reactor

The second example is intended to show that the use of convex approximations does not necessarily slow down convergence and does not preclude the ability of modifier adaptation to detect the correct set of active constraints. For this purpose, we consider the continuous stirred-tank reactor described in[27,33].

In this isothermal continuous stirred-tank reactor, the reactions $A+B \rightarrow C$, $2B \rightarrow D$ take place.

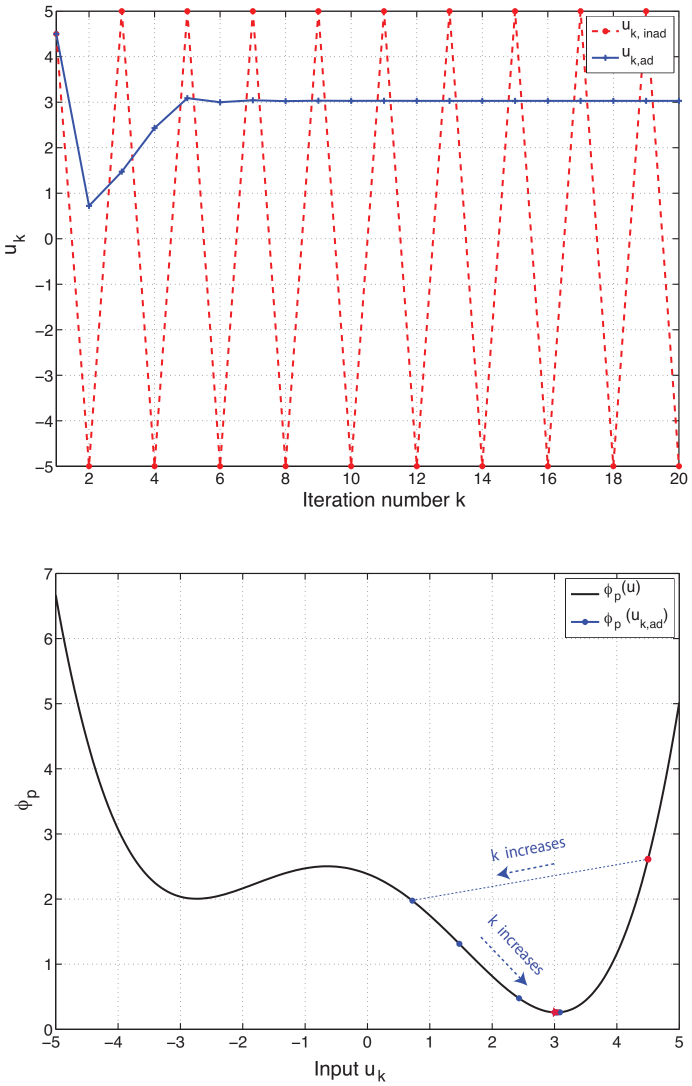


Figure 2: Case b. Top: evolution of the input with the inadequate (red) and the adequate (blue) models. Bottom: plant cost function (black) and successive plant cost values obtained with the adequate model (blue).

The two manipulated variables are the feed rates of A and B. The reactor mass and heat balances

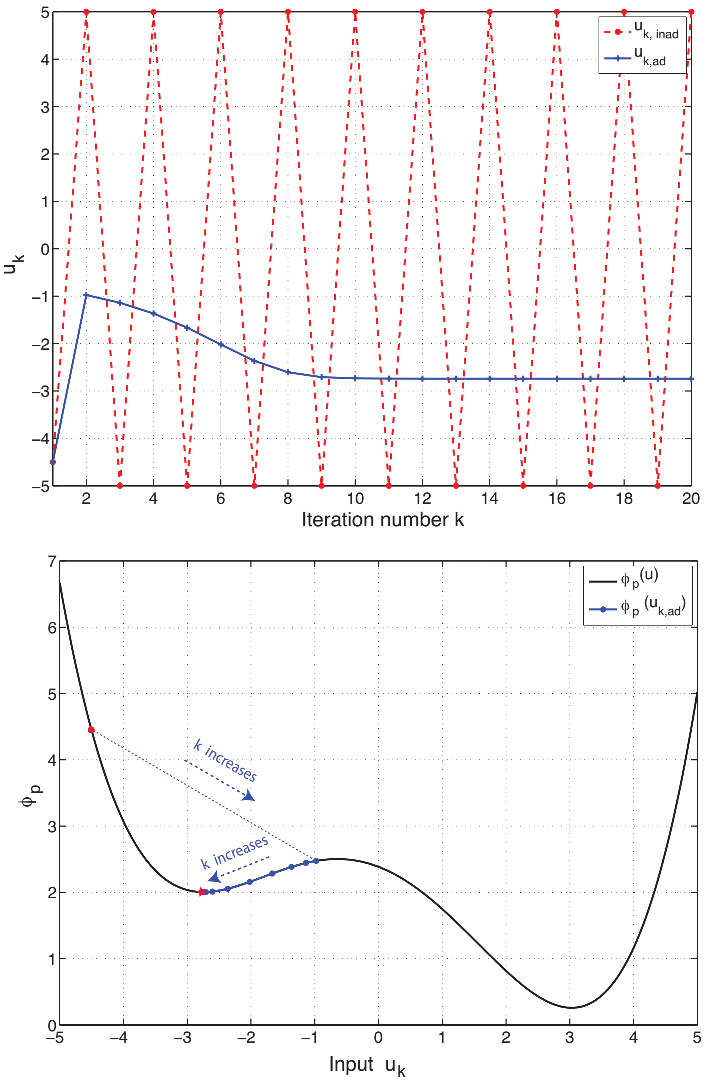


Figure 3: Case c. Top: evolution of the input with the inadequate (red) and the adequate (blue) models. Bottom: plant cost function (black) and successive plant cost values obtained with the adequate model (blue).

lead to the following steady-state model:

$$0 = -k_1\, c_A\, c_B + \frac{u_A}{V} c_{Ain} - \frac{u_A + u_B}{V} c_A \tag{37}$$

$$0 = -k_1\, c_A\, c_B - 2\, k_2\, c_B^2 + \frac{u_B}{V} c_{Bin} - \frac{u_A + u_B}{V} c_B \tag{38}$$

$$0 = k_1\, c_A\, c_B - \frac{u_A + u_B}{V} c_C \tag{39}$$

$$0 = k_2\, c_B^2 - \frac{u_A + u_B}{V} c_D \tag{40}$$

$$Q = -\, V \left(k_1\, c_A\, c_B\, \Delta H_{r,1} + k_2\, c_B^2\, \Delta H_{r,2} \right) , \tag{41}$$

where c_X denotes the steady-state 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 exothermic 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 nominal values of the model parameters are given in Table 1.

Table 1: Nominal values of the model parameters

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

Optimization problem

The optimization problem is formulated mathematically as follows:

$$\max_{u_A, u_B} \quad J := \left(\frac{c_C^2\,(u_A + u_B)^2}{u_A\, c_{Ain}} - w(u_A^2 + u_B^2) \right) \tag{42}$$

$$\text{s.t.} \quad \text{model equations (37)-(41)}$$

$$G_1 := \frac{Q}{Q_{max}} - 1 \leq 0 \tag{43}$$

$$G_2 := \frac{D}{D_{max}} - 1 \leq 0 \tag{44}$$

$$0 \leq u_A \leq u_{max} \tag{45}$$

$$0 \leq u_B \leq u_{max}\,. \tag{46}$$

Note that this problem differs from the optimization problems in [27,33] as constraints on the maximal heat generation and the final molar fraction of $D := \frac{c_D}{(c_A+c_B+c_C+c_D)}$ are introduced through the inequalities (43) and (44). The optimization problem is formulated to maximize the productivity of C, while penalizing the control action by means of $w\,(u_A^2 + u_B^2)$, w being a weighting parameter. The numerical values of the weighting parameter and the bounds are given in Table 2.

Since Theorem 1 and the associated corollary propose to use convex approximations in the case of minimization problems, the optimization problem of Eqns (42)-(46) is reformulated in terms of a minimization problem by noticing that the problem of maximizing J is formally equivalent to that of minimizing $\phi = -J$. We will therefore construct a strictly convex approximation of ϕ.

Table 2: Parameters of the optimization problem

w	0.004	$\frac{mol\,min}{l^2}$
Q_{max}	110	kcal
D_{max}	0.1	–
u_{max}	50	$\frac{\mathrm{l}}{\mathrm{min}}$

We consider parametric uncertainty in the kinetic parameters, with the plant characterized by the following (unknown) values $k_{1,p} = 1.4\,\frac{\mathrm{l}}{\mathrm{mol\,min}}$, $k_{2,p} = 0.4\,\frac{\mathrm{l}}{\mathrm{mol\,min}}$ and $c_{Ain,p} = 2.5\,\frac{\mathrm{mol}}{\mathrm{l}}$. Structural uncertainty is not introduced as it will occur upon constructing the convex approximations for ϕ, G_1 and G_2.

The plant is described by Eqns (37)-(41), with the model differing only in the values of the kinetic parameters and the inlet concentration c_{Ain}. The functions $\phi = -J$, G_1 and G_2 are given by Eqns (42)-(46). The optimal solutions for the plant (which is supposed to be unknown and thus will not be used thereafter) and the model are given in Table 3. It is seen that, not only does the parametric uncertainty lead to a different optimum, but also to a different set of active constraints, as G_1 is active and G_2 inactive in the plant optimal solution, whereas the model predicts the opposite.

Convex approximations

Several techniques are available in the literature to construct convex approximations[34,35]. Most methods construct local convex approximations with high local accuracy, the idea being to iteratively update these approximate models. This is for example the main philosophy behind SQP

Table 3: Solutions to the model and plant optimization problems

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

methods. The framework is different here since the model (and thus also its convex approximation) is kept unchanged.

A straightforward approach with a fixed pre-specified Hessian of the cost function is used in this work. The model equations are simulated for different values of u_A and u_B in the range $[0 \;\; u_{max}]$, and the corresponding values of the modeled cost and constraints are computed. Since the modifier-adaptation scheme is likely to be initialized at the nominal inputs, the following convex approximations ϕ_c, $G_{1,c}$ and $G_{2,c}$ are constructed by solving three least-squares regression problems:

$$\phi_c(\mathbf{u}) = \phi^* + \left[a_\phi,\, b_\phi\right](\mathbf{u} - \mathbf{u}^*) + \frac{1}{2}(\mathbf{u} - \mathbf{u}^*)^T \mathbf{Q}_\phi (\mathbf{u} - \mathbf{u}^*) \tag{47}$$

$$G_{1,c}(\mathbf{u}) = G_1(\mathbf{u}^*) + [a_{G_1},\, b_{G_1}](\mathbf{u} - \mathbf{u}^*) \tag{48}$$

$$G_{2,c}(\mathbf{u}) = G_2(\mathbf{u}^*) + [a_{G_2},\, b_{G_2}](\mathbf{u} - \mathbf{u}^*) \tag{49}$$

where $\mathbf{u} = [u_A \;\; u_B]^T$. The scalars a_ϕ, b_ϕ, a_{G_1}, b_{G_1}, a_{G_2}, b_{G_2} and the (2×2) matrix $\mathbf{Q}_\phi$ are the degrees of freedom of the least-squares regression problems. As seen from Eqns (48) and (49), the constraints are approximated by linear functions. This is mainly motivated by the fact that G_1 tends to exhibit a concave behavior in the region of interest, while G_2 is neither globally concave nor globally convex . Hence, it is easiest to model them as linear functions and enforce strict convexity of the cost function ϕ_c according to Corollary 1. The diagonal elements and one off-diagonal element of $\mathbf{Q}_\phi$ are determined (together with a_ϕ, b_ϕ) to force ϕ_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 some user-specified strictly positive values, here chosen both equal to 0.08. From a conceptual viewpoint, the choice of these eigenvalues is not important since strictly positive values guarantee the positive definiteness of $\mathbf{Q}_\phi$. In practice, however, these values will affect the quality of the approximation of ϕ and, in turn, the convergence rate. Hence, these values should be chosen with care, which is indeed possible as ϕ_c is not designed to fit the plant cost function ϕ_p (which is unknown and thus would require experiments), but rather the modeled cost ϕ. The results of the aforementioned constrained least-squares regression are given in Table 4.

Table 4: Parameters of the convex approximations

a_ϕ	-0.8305	b_ϕ	-0.9121
a_{G_1}	0.0051	b_{G_1}	0.0126
a_{G_2}	-0.0643	b_{G_2}	0.0857
$\mathbf{Q}_\phi =$	$\begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \end{bmatrix}$		

Simulation results

The two modifier-adaptation schemes, on the one hand the basic scheme (5)-(10) with the plant model (37)-(41), on the other hand the scheme using convex approximations consisting of Eqns (21)-(26) and the approximations (47)-(49) are implemented with the initial operating guess corresponding to the nominal inputs $\mathbf{u}^*$.

For both approaches, the modifiers are filtered exponentially according to (17), with the gain matrices chosen diagonal as $\Gamma = \gamma\,\mathbf{I}$ and $\Gamma_c = \gamma_c\,\mathbf{I}$, with k and k_c tuned manually to the values where damped oscillations around $\mathbf{u}_p^*$ start to occur ($\gamma = 0.4$ and $\gamma_c = 0.8$). Furthermore, it is assumed that perfect plant gradients are available, since both approaches are affected by gradient inaccuracies in a similar way.

Figure 4 depicts the evolution of the inputs with the RTO iteration number. With the convex approximations, convergence to the optimal values $\mathbf{u}_p^*$ occurs within 5 iterations, whereas it takes nearly 10 iterations with the plant model. This difference is smaller when the plant cost is considered as illustrated in Figure 5. The slightly faster convergence of the scheme using the convex approximations results from being able to use higher gains (less filtering) without oscillating around the solution. One possible interpretation for this surprising observation is that the linear and quadratic approximations "regularize" the cost and constraint functions, thereby providing the optimization scheme with a more global picture, while the original model being more detailed can have gradients that change more significantly between consecutive iterations. Note also that this observed feature is not likely to be generalizable to any convex approximation of any model-based optimization problem. For example, the use of a higher bound (0.24) for the eigenvalues of $\mathbf{Q}_\phi$ changes significantly the convex approximation and results in slower converge to the plant optimum (about 15 iterations), although a higher filter gain ($\gamma_c = 0.9$) can be used.

Figure 5 also shows that, when the plant model is used, some iterates exhibit a better performance than the true plant optimum, which is due to a violation of the constraint $G_{1,p}$ (see iterates 2-6 in Figure 6). In contrast, no violation of the plant constraints occurs when the convex model is used. This is a nice observation, which again cannot be generalized, although one expects to have fewer plant constraint violations by defining conservative linearized model constraints than by using the original model constraints. This conservatism is progressively reduced as the iterative scheme goes on. Furthermore, one can use a significant amount of filtering to force the modifier-adaptation scheme to converge from the feasible side, at the price of more iterations. This would probably be more difficult to tune with the original model. Finally, Figure 6 also shows the ability of modifier adaptation to converge to the active set of plant constraints regardless of the model used.

Conclusions

This article has proposed to use strictly convex model approximations in the context of real-time optimization via modifier adaptation. These approximations were shown to enforce model ad-

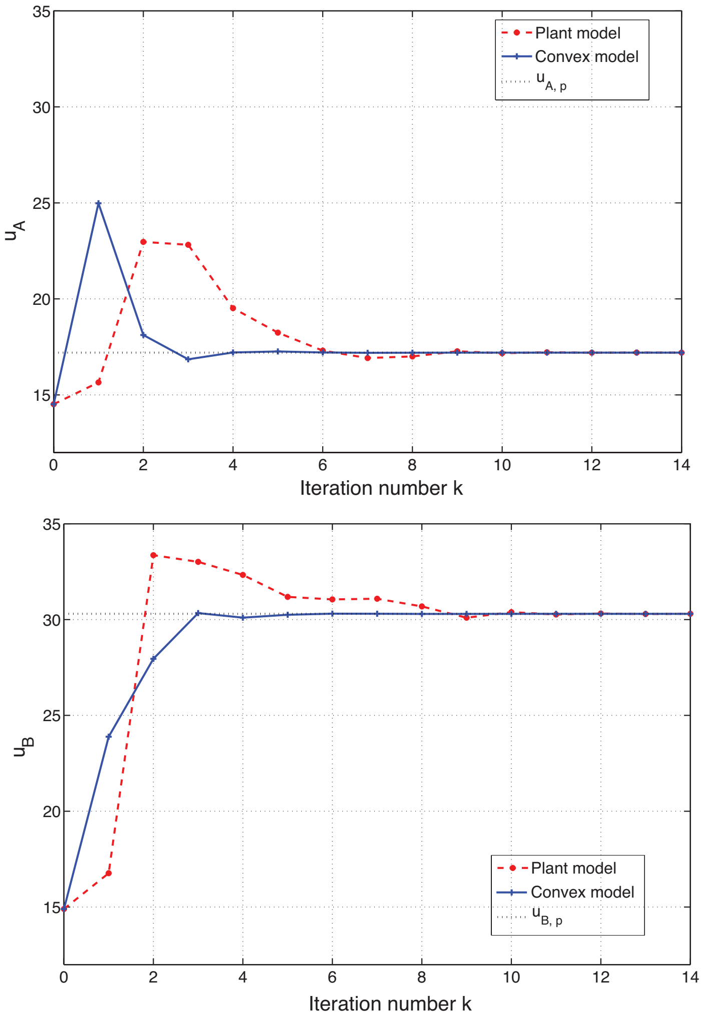


Figure 4: CSTR example. Evolution of the inputs u_A and u_B computed via modifier adaptation using the plant model (dotted line, red) and the convex approximations (solid line, blue).

equacy without affecting the ability of the modifier-adaptation scheme to converge to the true plant optimum. The original model-based optimization problem can therefore be approximated by a convex program, with a strictly convex cost function and convex (or linear) constraints. It is important to note that these convex functions are constructed as approximations to the plant *model* and thus they do not require any experiments in addition to those performed for building the original plant model. Such an approach may introduce model inaccuracies, which however are compensated for by the modifier-adaptation scheme, similarly to the way structural plant-model mismatch are accounted for. This approach paves the way toward using modern convex solvers for

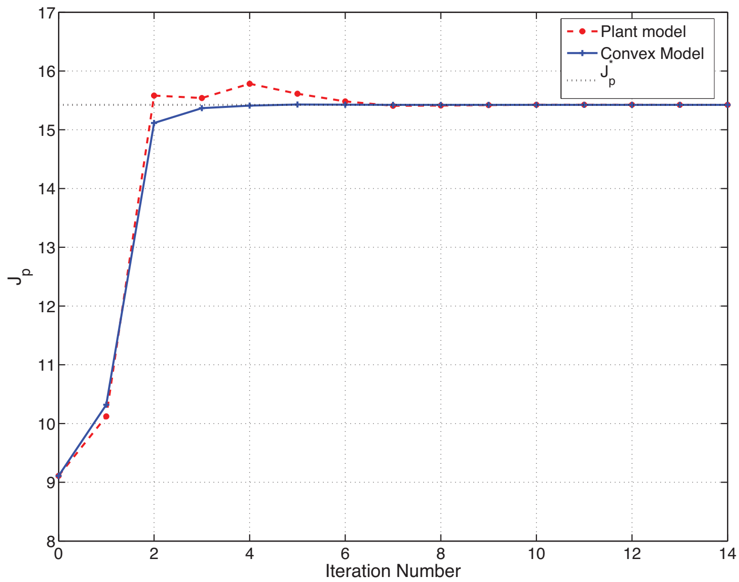
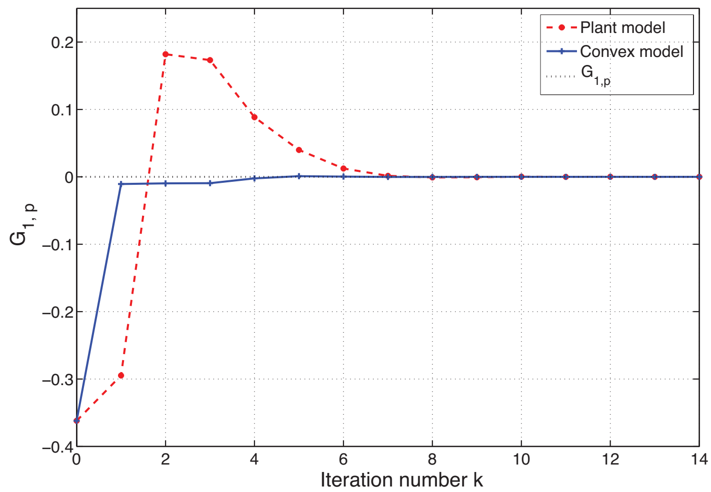


Figure 5: CSTR example. Evolution of plant cost J_p computed via modifier adaptation using the plant model (dotted line, red) and the convex approximations (solid line, blue).

modifier adaptation, with improved reliability and convergence properties compared to classical NLP solvers. The performance of the proposed scheme has been illustrated through two simulated examples, namely, a simple numerical example and a continuous stirred-tank reactor. It was observed that the number of iterations needed for convergence does not necessarily increase when convex approximations are used, although this could well be the case. Furthermore, this work has assumed that the plant gradients are available, as the objective was to compare the performance of modifier-adaptation schemes with the original and the convex models (and therefore the same assumption was made regarding the availability of plant gradients). Nonetheless, the estimation of plant gradients, and in particular their accuracy, is an important issue for RTO schemes. Note also that the convex approximations are used to perform the successive optimizations, while the original model could be used to estimate the plant gradients. The second example has shown that the use of convex approximations does not necessarily increase the likelihood of plant constraint violations.

More research is needed along the following lines: (i) the impact of convexification on the



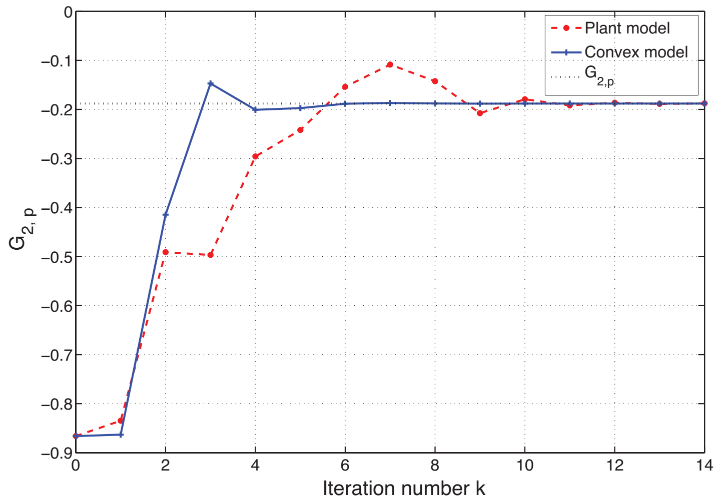


Figure 6: CSTR example. Evolution of the constraints $G_{1,p}$ and $G_{2,p}$ computed via modifier adaptation using the plant model (dotted line, red) and the convex approximations (solid line, blue).

performance of modifier-adaptation schemes, (ii) the development of efficient gradient estimation techniques, (iii) the impact of gradient inaccuracies on optimal performance, and (iv) ways of guaranteeing feasibility. Note also that these investigations can be performed in the general RTO context since all the mentioned issues refer to plant properties and not to model properties. A first step in this direction has been reported recently[28,29].

Nomenclature

Symbol	Signification	Dimension
ϕ	Cost function	$[1 \times 1]$
$\mathbf{G}$	Inequality constraints	$[n_G \times 1]$
θ	Model parameters	$[n_\theta \times 1]$
$\mathbf{u}$	Inputs	$[n_u \times 1]$
$\mathbf{G}^a$	Active constraints	$[n_a \times 1]$
$\mathbf{G}^{s,a}$	Strongly active constraints	$[n_{s,a} \times 1]$
$\mathbf{G}^{w,a}$	Weakly active constraints	$[(n_a - n_{s,a}) \times 1]$
L	Lagrangian function	$[1 \times 1]$
∇	Gradient/Jacobian w.r.t. **u**	$[1 \times n_u]/[n_G \times n_u]$
∇^2	Hessian w.r.t. **u**	$[n_u \times n_u]$
$(.)_p$	Quantity related to the plant	–
$(.)^*$	Quantity at optimum	–
$(.)_m$	Modified function	–
$(.)_c$	Convex approximation	–
ν	Lagrange multipliers	$[n_G \times 1]$
ν^a	- associated with active constraints	$[n_a \times 1]$
$\nu^{s,a}$	- associated with strongly active constraints	$[n_{s,a} \times 1]$
$\nu^{w,a}$	- associated with weakly active constraints	$[(n_a - n_{s,a}) \times 1]$
ε^ϕ	0^{th} – order modifier associated with ϕ	$[1 \times 1]$
λ^ϕ	1^{st} – order modifiers associated with ϕ	$[1 \times n_u]$
ε^G	0^{th} – order modifiers associated with **G**	$[n_G \times 1]$
λ^G	1^{st} – order modifiers associated with **G**	$[n_G \times n_u]$
Λ	Vector of modifiers	$\left[\left(n_G + (n_G+1)n_u\right) \times 1\right]$
k	Iteration number	–
Γ	Filter gain matrix	$\left[\left(n_G + (n_G+1)n_u\right) \times \left(n_G + (n_G+1)n_u\right)\right]$

References

1. Rotava, O.; Zanin, A. C. Multivariable control and real-time optimization – An industrial practical view. *Hydrocarbon Process.* **2005**, *84*, 61–71.

2. Boyd, S.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press: U.K., 2004.

3. Chachuat, B.; Srinivasan, B.; Bonvin, D. Adaptation strategies for real-time optimization. *Comput. Chem. Engng.* **2009**, *33*, 1557–1567.

4. Marlin, T.; Hrymak, A. Real-time operations optimization of continuous processes. *AIChE Symposium Series - CPC-V* **1997**, *93*, 156–164.

5. Forbes, J. F.; Marlin, T. Design Cost: A systematic approach to technology selection for model-based real-time optimization systems. *Comput. Chem. Eng.* **1996**, *20*, 717–734.

6. François, G.; Bonvin, D. Measurement-based real-time optimization of chemical processes. *Adv. Chem. Eng. - Control and Optimization of Process Systems* **2013**, *43*, 1–50.

7. Bonvin, D.; Srinivasan, B. On the role of the necessary conditions of optimality in structuring dynamic real-time optimization schemes. *Comput. Chem. Eng.* **2013**, *51*, 172–180.

8. Quelhas, A. D.; De Jesus, N. J. C.; Pinto, J. C. Common vulnerabilities of RTO implementations in real chemical processes. *Can. J. Chem. Eng.* **2012**, *91*, 652–668.

9. Friedman, Y. Z. Closed-loop optimization update - A step closer to fulfilling the dream. *Hydrocarbon Process.* **2000**, *79*, 15–16.

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

11. Roberts, P. D.; Williams, T. On an algorithm for combined system optimisation and parameter estimation. *Automatica* **1981**, *17*, 199–209.

12. Brdyš, M.; Tatjewski, P. *Iterative Algorithms for Multilayer Optimizing Control*; Imperial College Press: London, 2005.

13. Gao, W.; Engell, S. Iterative set-point optimization of batch chromatography. *Comput. Chem. Eng.* **2005**, *29*, 1401 – 1409.

14. Golshan, M.; Pishvaie, M. R.; Bozorgmehry Boozarjomehry, R. Stochastic and global real time optimization of Tennessee Eastman challenge problem. *Eng. App. of Artificial Int.* **2008**, *21*, 215–228.

15. Prata, D. M.; Schwaab, M.; Lima, E. L.; Pinto, J. C. Nonlinear dynamic data reconciliation and parameter estimation through particle swarm optimization: Application for an industrial polypropylene reactor. *Chem. Eng. Sci.* **2009**, *64*, 3953–3967.

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

17. 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.

18. Srinivasan, B. Real-time optimization of dynamic systems using multiple units. *Int. J. Robust Nonlinear Control* **2007**, *17*, 1183–1193.

19. Biegler, L.; Cervantes, A.; Wächter, A. Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.* **2002**, *57*, 575 – 593.

20. Vassiliadis, V.; Sargent, W.; Pantelides, C. Solution of a class of multistage dynamic optimization problems. 1. Problems without path constraints. *Ind. Eng. Chem. Res.* **1994**, *33*, 2123–2133.

21. Bazarra, M.; Sherali, H.; Shetty, C. *Nonlinear Programming: Theory and Algorithms*; John Wiley and Sons, 2nd edition: New York, USA, 1993.

22. Srinivasan, B.; Bonvin, D. Interplay between Identification and Optimization in Run-to-run Optimization Schemes. *In Proceedings of the American Control Conference* **2002**, 2174–2179.

23. Engl, H.; Hanke, M.; Neubauer, A. *Regularization of Inverse Problems*; Kluwer Academic Publishers: Dordrecht, the Netherlands, 2000.

24. Bunin, G.; Francois, G.; Bonvin, D. From discrete measurements to bounded gradient estimates: a look at some regularizing structures. *Ind. Eng. Chem. Res.* **2013**, in press, doi:10.1021/ie303309a.

25. Hjalmarsson, H. Iterative feedback tuning - An overview. *Int. J. Adapt. Control Signal Process.* **2002**, *16*, 373–395.

26. Gros, S.; Srinivasan, B.; Bonvin, D. Optimizing control based on output feedback. *Comput. Chem. Engng.* **2009**, *33*, 191–198.

27. Francois, G.; Srinivasan, B.; Bonvin, D. Comparison of six implicit real-time optimization schemes. *J. Européen des Systèmes Automatisés* **2012**, *46*, 291–305.

28. Bunin, G.; Francois, G.; Bonvin, D. Sufficient conditions for feasibility and optimality of real-time optimization schemes - I. Theoretical foundations. *Comput. Chem. Eng.* **2013**, Submitted.

29. Bunin, G.; Francois, G.; Bonvin, D. Sufficient conditions for feasibility and optimality of real-time optimization schemes - II. Implementation issues. *Comput. Chem. Eng.* **2013**, Submitted.

30. Gill, P. E.; Murray, W.; Wright, M. H. *Practical Optimization*; Academic Press: London, 2003.

31. Francois, G.; Srinivasan, B.; Bonvin, D.; Hernandez Barajas, J.; Hunkeler, D. Run-to-run adaptation of a semi-adiabatic policy for the optimization of an industrial batch polymerization process. *Ind. Eng. Chem. Res.* **2004**, *43*, 7238–7242.

32. Ferreau, J. *Model Predictive Control Algorithms for Applications with Millisecond Timescales*; KU Leuven, Belgium, 2011; Doctoral thesis.

33. Srinivasan, B.; Francois, G.; Bonvin, D. Comparison of gradient estimation methods for real-time optimization. *Comput.-Aided Chem. Eng.* **2011**, *29*, 607–611.

34. Fleury, C. First- and second-order convex approximations strategies in structural optimization. *Struct. Optim.* **1989**, *1*, 3–10.

35. Zhang, W.-H.; Fleury, C. A modification of convex approximation methods for structural optimization. *Comput. Struct.* **1997**, *64*, 89–95.