18th European Symposium on Computer Aided Process Engineering – ESCAPE 18
Bertrand Braunschweig and Xavier Joulia (Editors)

Model Parameterization Tailored to Real-time Optimization

Benoît Chachuat,[a] Bala Srinivasan,[b] Dominique Bonvin[a]

[a]*Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 9, CH-1015 Lausanne, Switzerland*
[b]*Département de Génie Chimique, Ecole Polytechnique de Montréal, C.P. 6079 Succ. centre ville, Montréal (QC), H3C 3A7, Canada*

Abstract

Challenges in real-time process optimization mainly arise from the inability to build and adapt accurate models for complex physico-chemical processes. This paper surveys different ways of using measurements to compensate for model uncertainty in the context of process optimization. A distinction is made between *model-adaptation methods* that use the measurements to update the parameters of the process model before repeating the optimization, *modifier-adaptation methods* that adapt constraint and gradient modifiers, and *direct-input-adaptation methods* that convert the optimization problem into a feedback control problem. This paper argues in favor of modifier-adaptation methods, since it uses a model parameterization, measurements, and an update criterion that are tailored to the tracking of the necessary conditions of optimality.

Keywords: Measurement-based optimization; Real-time optimization; Plant-model mismatch; Model adaptation; Model parameterization.

1. Introduction

Optimization of process performance has received attention recently because, in the face of growing competition, it represents the natural choice for reducing production costs, improving product quality, and meeting safety requirements and environmental regulations. Process optimization is typically based on a process model, which is used by a numerical procedure for computing the optimal solution. In practical situations, however, an accurate process model can rarely be found with affordable effort. Uncertainty results primarily from trying to fit a model of limited complexity to a complex process. The model-fitting task is further complicated by the fact that process data are usually noisy and signals do not carry sufficient excitation. Therefore, optimization using an inaccurate model might result in suboptimal operation or, worse, infeasible operation when constraints are present [8].

Two main classes of optimization methods are available for handling uncertainty. The essential difference relates to whether or not measurements are used in the calculation of the optimal strategy. In the absence of measurements, a robust optimization approach is typically used, whereby conservatism is introduced to guarantee feasibility for the entire range of expected variations [18]. When measurements are available, adaptive optimization can help adjust to process changes and disturbances, thereby reducing conservatism [9]. It is interesting to note that the above classification is similar to that found in control problems with the robust and adaptive techniques.

An optimal solution has to be feasible and, of course, optimal. In practice, feasibility is often of greater importance than optimality. In the presence of model uncertainty,

feasibility is usually enforced by the introduction of backoffs from the constraints. The availability of measurements helps reduce these backoffs and thus improve performance [6]. Generally, it is easier to measure or infer constrained quantities (e.g. temperature or pressure) than estimate gradients of the cost and constrained quantities. These elements clearly set a priority of actions in the framework of adaptive optimization.

This paper discusses three major approaches in adaptive optimization that differ in the way adaptation is performed, namely (i) *model-adaptation methods*, where the measurements are used to refine the process model, and the updated model is used subsequently for optimization [7,17]; (ii) *modifier-adaptation methods*, where modifier terms are added to the cost and constraints of the optimization problem, and measurements are used to update these terms [8,10,20]; and (iii) *direct-input-adaptation methods*, where the inputs are adjusted by feedback controllers, hence not requiring optimization but a considerable amount of prior information regarding control design [9,21,25].

These approaches are surveyed and compared in the first part of the paper. A critical discussion follows, which argues in favor of modifier-adaptation methods that share many advantages of the other methods.

An important issue not addressed herein concerns the availability of reliable measurements. Also, note that the intended purpose of the models presented here is optimization and not prediction of the system behavior.

2. Static Optimization Problems

For continuous processes operating at steady state, optimization typically consists in determining the operating point that minimize or maximize some performance of the process (such as minimization of operating cost or maximization of production rate), while satisfying a number of constraints (such as bounds on process variables or product specifications). In mathematical terms, this optimization problem can be stated as follows:

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{minimize:}} \quad & \Phi_p(\boldsymbol{\pi}) := \phi_p\left(\boldsymbol{\pi}, \mathbf{y}_p\right) \\ \text{subject to:} \quad & \mathbf{G}_p(\boldsymbol{\pi}) := \mathbf{g}_p\left(\boldsymbol{\pi}, \mathbf{y}_p\right) \le \mathbf{0} \end{aligned} \tag{1}$$

where $\boldsymbol{\pi} \in \mathbb{R}^{n_\pi}$ and $\mathbf{y}_p \in \mathbb{R}^{n_y}$ stand for the process input (set points) and output vectors, respectively; $\phi_p : \mathbb{R}^{n_\pi} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the plant performance index; and $\mathbf{g}_p : \mathbb{R}^{n_\pi} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_g}$ is the vector of constraints imposed on the input and output variables.

In contrast to continuous processes, the optimization of batch and semi-batch processes consists in determining time-varying control profiles, $\mathbf{u}(t)$, $t_0 \le t \le t_\mathrm{f}$. This typically involves solving a dynamic optimization problem, possibly with path and terminal constraints. A practical way of solving such problems is by parameterizing the control profiles using a finite number of parameters $\boldsymbol{\pi}$, e.g., a polynomial approximation of $\mathbf{u}(t)$ on finite elements. Although the process is dynamic in nature, a static map can be used to describe the relationship between the process inputs $\boldsymbol{\pi}$ and the outcome of the batch $\mathbf{y}(t_\mathrm{f})$. Hence, the problem can be regarded as a finite-dimensional static optimization problem similar to (1), and the optimization approaches discussed in the following sections can also be used in the framework of run-to-run optimization of

batch and semi-batch processes (see, e.g., [9]).
In practice, the mapping relating the process inputs and outputs is typically unknown, and only an approximate model is available,

$$\mathbf{y} = \mathbf{f}(\boldsymbol{\pi}, \boldsymbol{\theta}) \tag{2}$$

with $\mathbf{y} \in \mathbb{R}^{n_y}$ representing the model outputs, and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ the model parameters, and $\mathbf{f} : \mathbb{R}^{n_\pi} \times \mathbb{R}^{n_\theta} \to \mathbb{R}^{n_y}$ the input-output mapping. Accordingly, an approximate solution of problem (1) is obtained by solving the following model-based optimization problem:

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{minimize:}} \quad & \Phi(\boldsymbol{\pi}, \boldsymbol{\theta}) := \phi(\boldsymbol{\pi}, \mathbf{y}, \boldsymbol{\theta}) \\ \text{subject to:} \quad & \mathbf{y} = \mathbf{f}(\boldsymbol{\pi}, \boldsymbol{\theta}) \\ & \mathbf{G}(\boldsymbol{\pi}, \boldsymbol{\theta}) := \mathbf{g}(\boldsymbol{\pi}, \mathbf{y}, \boldsymbol{\theta}) \le \mathbf{0} \end{aligned} \tag{3}$$

Provided that the objective and constraint functions in (1) and (3) are continuous and the feasible domains of these problems are nonempty and bounded, optimal solution points $\boldsymbol{\pi}_p^*$ and $\boldsymbol{\pi}^*$ are guaranteed to exist for (1) and (3), respectively [2]. Note that such optimal points may not be unique due to nonconvexity. The KKT conditions – also called necessary conditions of optimality (NCO) – must hold at an optimal solution point provided that the active constraints satisfy a regularity condition at that point [2]. For Problem (3), the KKT conditions read:

$$\begin{gathered} \mathbf{G}(\boldsymbol{\pi}^*, \boldsymbol{\theta}) \le \mathbf{0}, \quad \boldsymbol{\nu}^* \ge \mathbf{0}, \\ \frac{\partial \Phi}{\partial \boldsymbol{\pi}}(\boldsymbol{\pi}^*, \boldsymbol{\theta}) + \boldsymbol{\nu}^{*\mathrm{T}} \frac{\partial \mathbf{G}}{\partial \boldsymbol{\pi}}(\boldsymbol{\pi}^*, \boldsymbol{\theta}) = \mathbf{0}, \\ \boldsymbol{\nu}^{*\mathrm{T}} \mathbf{G}(\boldsymbol{\pi}^*, \boldsymbol{\theta}) = 0 \end{gathered} \tag{4}$$

where $\boldsymbol{\nu}^* \in \mathbb{R}^{n_g}$ is the vector of Lagrange multipliers. The KKT conditions involve the quantities $\mathbf{G}, \frac{\partial \Phi}{\partial \boldsymbol{\pi}}$ and $\frac{\partial \mathbf{G}}{\partial \boldsymbol{\pi}}$, which are denoted collectively by $\mathbf{K}$ subsequently.

3. A Classification of Real-time Optimization Schemes

Real-time optimization (RTO) schemes improve process performance by adjusting selected optimization variables using available measurements. The goal of this closed-loop adaptation is to drive the operating point towards the true plant optimum in spite of inevitable structural and parameter model errors. RTO methods can be classified in different ways. This section presents one such classification based on the parameters that can be adapted, as illustrated in Fig. 1; note that repeated numerical optimization is used in the methods of columns 1 and 2, but not in those of column 3.

3.1. Model-Adaptation Methods

The standard way of devising a RTO scheme is the so-called *two-step approach* [1], also referred to as *repeated identification and optimization* in the literature. In the first step, the values of (a subset of) the adjustable model parameters $\boldsymbol{\theta}$ are estimated by using the available process measurements. This is typically done by minimizing the lack of closure in the steady-state model equations (2), such as the weighted sum of squared errors between measured outputs $\mathbf{y}_p$ and predicted outputs $\mathbf{y}$ [17].

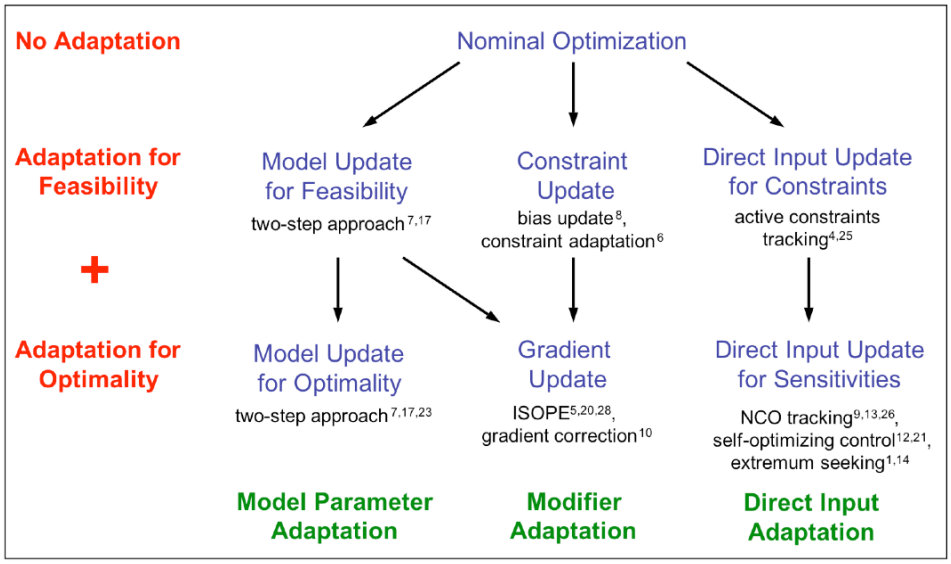

Figure 1: Optimization scenarios that use measurements to adapt for feasibility and optimality.

A key, yet difficult, decision in the model-update step is to select the parameters to be updated. These parameters should be identifiable, represent actual changes in the process, and contribute to approach the process optimum; also, model adequacy proves to be a useful criterion to select candidate parameters for adaptation [8]. Clearly, the smaller the subset of parameters, the better the confidence in the parameter estimates, and the lower the required excitation. But too low a number of adjustable parameters can lead to completely erroneous models, and thereby to a false optimum.

In the second step, the updated model is used to determine a new operating point, by solving an optimization problem similar to (3). Model-adaptation methods can be written generically using the following two equations (see Fig. 2):

$$\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1} + \boldsymbol{\psi}_{\text{upd}}\left(\mathbf{y}_p(\boldsymbol{\pi}^{k-1}) - \mathbf{y}(\boldsymbol{\pi}^{k-1}, \boldsymbol{\theta}^{k-1})\right), \tag{5}$$

$$\boldsymbol{\pi}^k = \boldsymbol{\psi}_{\text{opt}}\left(\boldsymbol{\theta}^k\right) \tag{6}$$

where $\boldsymbol{\psi}_{\text{upd}}$ is the map describing the model-update step, such that $\boldsymbol{\psi}_{\text{upd}}(\mathbf{0}) = \mathbf{0}$; $\boldsymbol{\psi}_{\text{opt}}$, the map describing the optimization step. Note that the handles for correction are a subset of the adjustable model parameters $\boldsymbol{\theta}$. The use of auxiliary measurements ($\mathbf{y}_p$) presents the advantage that *any* available measurement can be used.

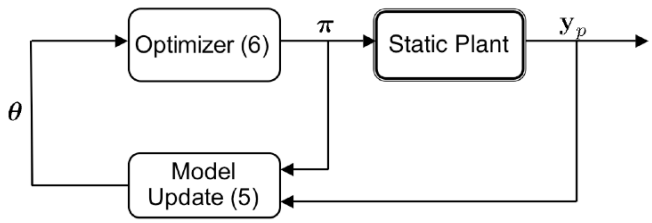

Figure 2. Model-adaptation method: Two-step approach

It is well known that the interaction between the model-update and reoptimization steps must be considered carefully for the two-step approach to achieve optimal performance. In the absence of plant-model mismatch and when the parameters are structurally and practically identifiable, convergence to the plant optimum may be achieved in one iteration. However, in the presence of plant-model mismatch, whether the scheme converges, or to which operating point the scheme converges, becomes anybody's guess. This is due to the fact that the update objective might be unrelated to the cost or constraints in the optimization problem, and minimizing the mean-square error in $\mathbf{y}$ may not help in our quest for feasibility and optimality. To alleviate this difficulty, Srinivasan and Bonvin [23] presented an approach where the criterion in the update problem is modified to account for the subsequent optimization objective.
Convergence under plant-model mismatch has been addressed by several authors [3,8]; it has been shown that an optimal operating point is reached if model adaptation leads to a matching of the KKT conditions for the model and the plant.

Theorem 1. *Let the parameter adaptation (5) be such that the plant measurements* $\mathbf{K}_p$ *match those predicted by the model,* $\mathbf{K}$ *. Then, upon convergence, the model-adaptation scheme (5-6) reaches an (local) optimum operating point of the plant.*

A proof of this result is readily obtained from the assumption that the KKT conditions predicted by the model equal those achieved by the plant. With such a matching, the converged solution corresponds to a (local) plant optimum.
Although Theorem 1 is straightforward, the KKT-matching assumption is difficult to meet in practice. It requires an "adequate" parameterization so that all the components of the KKT conditions can match, as well as "adequate" measurements and an "adequate" update criterion.

3.2. Modifier-Adaptation Methods

In order to overcome the modeling deficiencies and to handle plant-model mismatch, several variants of the two-step approach have been presented in the literature. Generically, they consist in modifying for the cost and constraints of the optimization problem for the KKT conditions of the model and the plant to match. The optimization problem with modifiers can be written as follows:

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{minimize:}} \quad & \tilde{\Phi}(\boldsymbol{\pi},\boldsymbol{\theta}) \coloneqq \Phi(\boldsymbol{\pi},\boldsymbol{\theta}) + \boldsymbol{\lambda}_{\Phi}^{\mathrm{T}}\boldsymbol{\pi} \\ \text{subject to:} \quad & \tilde{\mathbf{G}}(\boldsymbol{\pi},\boldsymbol{\theta}) \coloneqq \mathbf{G}(\boldsymbol{\pi},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_G + \boldsymbol{\lambda}_G^{\mathrm{T}}\left(\boldsymbol{\pi} - \boldsymbol{\pi}^k\right) \le \mathbf{0} \end{aligned} \tag{7}$$

where $\boldsymbol{\varepsilon}_G \in \mathbb{R}^{n_g}$ is the constraint bias, $\boldsymbol{\lambda}_{\Phi} \in \mathbb{R}^{n_\pi}$ the cost-gradient modifier, and $\boldsymbol{\lambda}_G \in \mathbb{R}^{n_\pi \times n_g}$ the constraint-gradient modifier; these modifiers are denoted collectively by $\boldsymbol{\Lambda}$ subsequently.

- The constraint bias $\boldsymbol{\varepsilon}_G$ represents the difference between the measured and predicted constraints, $\boldsymbol{\varepsilon}_G \coloneqq \mathbf{G}_p(\boldsymbol{\pi}) - \mathbf{G}(\boldsymbol{\pi},\boldsymbol{\theta})$, evaluated at the previous operating point $\boldsymbol{\pi}^k$. Adapting only $\boldsymbol{\varepsilon}_G$ leads to the so-called constraint-adaptation scheme [6,8]. Such a scheme is rather straightforward and corresponds to common industrial practice [17].
- The cost-gradient modifier $\boldsymbol{\lambda}_{\Phi}$ represents the difference between the estimated and predicted values of the cost gradient, $\boldsymbol{\lambda}_{\Phi}^{\mathrm{T}} \coloneqq \left[\dfrac{\partial \Phi_p}{\partial \boldsymbol{\pi}} - \dfrac{\partial \Phi}{\partial \boldsymbol{\pi}}\right]$, evaluated at the previous

operating point $\boldsymbol{\pi}^k$. The pertinent idea of adding a gradient modifier to the cost function of the optimization problem dates back to the work of Roberts [19] in the late 1970s. Note that it was originally proposed in the framework of two-step methods to better integrate the model update and optimization subproblems and has led to the so-called ISOPE approach [4].

- The constraint-gradient modifier $\boldsymbol{\lambda}_G$, finally, represents the difference between the estimated and predicted values of the constraint gradients, $\boldsymbol{\lambda}_G^{\mathrm{T}} := \left[\dfrac{\partial \mathbf{G}_p}{\partial \boldsymbol{\pi}} - \dfrac{\partial \mathbf{G}}{\partial \boldsymbol{\pi}}\right]$, evaluated at the previous operating point $\boldsymbol{\pi}^k$. The idea of adding such a first-order modifier term to the process-dependent constraints, in addition to the constraint bias $\boldsymbol{\varepsilon}_G$, was proposed recently by Gao and Engell [12]. This modification allows matching, not only the values of the constraints, but also their gradients.

Overall, the update laws in modifier-adaptation methods can be written as (see Fig. 3):

$$\boldsymbol{\Lambda}^k = \boldsymbol{\Lambda}^{k-1} + \boldsymbol{\psi}_{\mathrm{upd}}\left(\mathbf{K}_p(\boldsymbol{\pi}^{k-1}) - \tilde{\mathbf{K}}(\boldsymbol{\pi}^{k-1}, \boldsymbol{\theta})\right) \tag{8}$$

$$\boldsymbol{\pi}^k = \boldsymbol{\psi}_{\mathrm{opt}}\left(\boldsymbol{\Lambda}^k\right) \tag{9}$$

where $\tilde{\mathbf{K}} := \left(\tilde{\mathbf{G}}, \dfrac{\partial \tilde{\Phi}}{\partial \boldsymbol{\pi}}, \dfrac{\partial \tilde{\mathbf{G}}}{\partial \boldsymbol{\pi}}\right)$, with $\tilde{\Phi}, \tilde{\mathbf{G}}$ as defined in Problem (7); and the modifier update map, $\boldsymbol{\psi}_{\mathrm{upd}}$, is such that $\boldsymbol{\psi}_{\mathrm{upd}}(\mathbf{0}) = \mathbf{0}$. The handles for correction are the modifier parameters $\boldsymbol{\Lambda}$ instead of $\boldsymbol{\theta}$ used in the context of model-adaptation schemes. Also, the measurements $\mathbf{K}_p$ required to make the adaptation are directly related to the KKT conditions; auxiliary measurements are not used in this framework. Observe the one-to-one correspondence between the number of measurements/estimates and the number of adjustable parameters. In particular, identifiability is automatically satisfied, and so are the KKT-matching conditions.

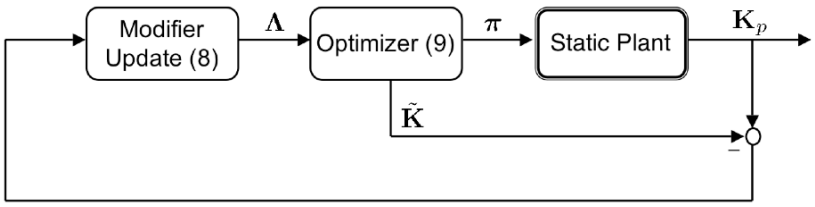

Figure 3. Modifier-adaptation method: Matching the KKT conditions

Modifier-adaptation methods possess nice theoretical properties, as summarized by the following theorem.

Theorem 2. *Let the cost and constraint functions be parameterized as in Problem (7). Also, let the information on the values of* $\mathbf{K}_p$ *be available and used to adapt the modifiers* $\boldsymbol{\Lambda}$. *Then, upon convergence, the modifier-adaptation scheme (8-9) reaches an (local) optimum operating point of the plant.*

A proof of this result is easily obtained by noting that, upon convergence, the modified constraints $\tilde{\mathbf{G}}$ in (7) match the plant constraints $\mathbf{G}_p$, and the gradients of the modified cost and constraint functions match those of the plant (see also [10]). It follows that the active set is correctly determined and the converged solution satisfies the KKT conditions.

Hence, there is a close link between the model- and modifier-adaptation methods in that the parameterization and the update procedure are both intended to match the KKT conditions. Essentially, modifier-adaptation schemes use a model-predictive control with a one-step prediction horizon. Such a short horizon is justified because the system is static. However, since the updated modifiers are valid only locally, modifier-adaptation schemes require some amount of filtering/regularization (either in the modifiers or in the inputs) to avoid too aggressive corrections that may destabilize the system.

3.3. Direct-Input-Adaptation Methods

This last class of methods provides a way of avoiding the repeated optimization of a process model by transforming it into a feedback control problem that directly manipulates the input variables. This is motivated by the fact that practitioners like to use feedback control of selected variables as a way to counteract plant-model mismatch and plant disturbances, due to its simplicity and reliability compared to on-line optimization. The challenge is to find functions of the measured variables which, when held constant by adjusting the input variables, enforce optimal plant performance [19,21]. Said differently, the goal of the control structure is to achieve a similar steady-state performance as would be realized by an (fictitious) on-line optimizing controller.

In the presence of uncertainty, the inputs determined from off-line solution of problem (3) for nominal parameter values satisfy the NCO (4) but typically violate the NCO related to the plant itself. Hence, a rather natural idea is to correct the input variables $\boldsymbol{\pi}$ so as to enforce the NCO for the plant [1,9,14]; in other words, the controlled variables are chosen as the NCO terms, with the corresponding set points equal to zero.

Tracking of the NCO (4) consists of three steps: (i) determining the active set (positivity condition on Lagrange multipliers), (ii) following the active constraints, and (iii) pushing the sensitivity to zero. Determining the active set requires a switching strategy, whereby a constraint is included in the active set when it is attained, and deactivated when its Lagrange multiplier goes negative [29]. This switching logic renders the scheme more complex, and in the interest of simplicity, it may be assumed that the active constraints do not change. Note that such an assumption is always verified in the neighborhood of an optimal solution and is observed in many practical situations.

Once the active set is known, the inputs are split into : (i) constraints-seeking directions that are used to track the active constraints, and (ii) sensitivity-seeking directions that are adapted to force the reduced gradients to zero. The active constraints $\mathbf{G}_p^{\mathbf{a}}$ and the reduced cost gradient $\nabla_{\boldsymbol{\pi}}^{\mathbf{r}}\Phi_p \coloneqq \dfrac{\partial \Phi_p}{\partial \boldsymbol{\pi}}\left[\mathbf{I}-\mathbf{P}^{+}\mathbf{P}\right]$, with $\mathbf{P} \coloneqq \dfrac{\partial \mathbf{G}_p^{\mathbf{a}}}{\partial \boldsymbol{\pi}}$, need to be measured.

Since, in general, the constraint terms are easily measured, or can be reliably estimated, adjusting the inputs in the constraint-seeking directions to track the active constraints is rather straightforward [4,25,27]. Adjusting the sensitivity-seeking directions is more involved, mainly due to the difficulty in the measurement of the gradient terms. François et al. [9] proposed a two-time-scale adaptation strategy, wherein adaptation in the sensitivity-seeking directions takes place at a much slower rate than in the constraint-seeking directions.

Direct-input-adaptation methods obey the following equations (see Fig. 4):

$$\boldsymbol{\pi}^k = \boldsymbol{\pi}^{k-1} + \boldsymbol{\psi}_{\text{con}}\left(\mathbf{G}_p^{\mathbf{a}}(\boldsymbol{\pi}^{k-1}), \nabla_{\boldsymbol{\pi}}^{\mathbf{r}}\Phi_p(\boldsymbol{\pi}^{k-1})\right) \tag{10}$$

$$\left(\mathbf{G}_p^{\mathbf{a}}(\boldsymbol{\pi}^k), \nabla_{\boldsymbol{\pi}}^{\mathbf{r}}\Phi_p(\boldsymbol{\pi}^k)\right) = \boldsymbol{\psi}_{\text{swi}}\left(\mathbf{K}_p(\boldsymbol{\pi}^k)\right) \tag{11}$$

where $\boldsymbol{\psi}_{\text{con}}$ is the map describing the controller, such that $\boldsymbol{\psi}_{\text{con}}(\mathbf{0},\mathbf{0}) = \mathbf{0}$; $\boldsymbol{\psi}_{\text{swi}}$, the map describing the switching logic for determination of the active set. The handles for correction are the process inputs $\boldsymbol{\pi}$, i.e., no specific parameterization is required here. Both the active constraints and the reduced cost gradient are forced to zero, e.g., with a discrete integral-type controller.

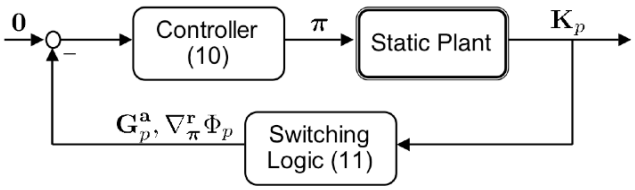

Figure 4. Direct-input-adaptation method: Tracking the NCO using control

Direct-input-adaptation methods also possess nice theoretical properties, as summarized by the following theorem.

Theorem 3. *Let the information on the values of* $\mathbf{K}_p$ *be available and used to adapt the inputs and the active set given by (10-11). Then, upon convergence, the direct-input-adaptation scheme (10-11) reaches an (local) optimum operating point of the plant.*

Note that the active process constraints and reduced gradients are both zero upon convergence. Moreover, since the positivity of the Lagrange multipliers is guaranteed by the switching logic, the active set is correctly identified and the NCO are satisfied.
The key question lies in the design of the controller. Unlike optimization-based schemes, the required smoothening is provided naturally via appropriate controller tuning.

3.4. Evaluation of the various methods

A systematic approach for evaluating the performance of adaptive optimization schemes, named the extended cost design, has been presented in [30]. It incorporates measures of both the convergence rate and the effect of measurement noise. Interestingly, it is shown that in the presence of noise, a standard two-step algorithm may perform better, in terms of the proposed metric, than modified algorithms compensating for plant-model mismatch such as ISOPE. Another approach to performance characterization for adaptive optimization has been proposed in [15], which considers the backoff from active inequality constraints required to ensure feasibility. Therein, better adaptive optimization approaches produce smaller backoffs.

4. Use of Measurements for Feasible and Optimal Operation

This section discusses the two main rows in Fig.1. The feasibility issue is addressed first, and various gradient estimation techniques are summarized next.

4.1. Feasible Operation

In practical applications, guaranteeing feasible operation is often more important than achieving the best possible performance. Hence, first priority is given to meeting the

process constraints (such as safety requirements and product specifications) and only second priority to improving process performance in terms of the objective function. Interestingly, the results of a variational analysis in the presence of small parametric error support the priority given to constraint satisfaction over the sensitivity part of the NCO [6]. More specifically, it has been shown that, in addition to inducing constraint violation, failure to adapt the process inputs in the constraint-seeking directions results in cost variations in the order of the parameter variations $\delta\boldsymbol{\theta}$; in contrast, failure to adapt the inputs in the sensitivity-seeking directions gives cost variations in the order of $\delta\boldsymbol{\theta}^2$ only.

The ability to guarantee feasible operation is addressed next for the three classes of methods presented above. In model-adaptation methods, since the plant constraints are predicted by the process model, constraint matching – but not necessarily full KKT matching – is needed to guarantee feasibility; however, this condition may be difficult to meet, e.g., when the model is updated by matching a set of outputs not directly related to the active constraints. With modifier-adaptation methods, feasibility is guaranteed upon convergence, provided that *all* the constraint terms are measured [6]; yet, ensuring feasibility does not necessarily imply that the correct active set has been determined due to the use of possibly inaccurate cost and constraint gradients, e.g., when gradient modifiers are not considered. Finally, in direct-input-adaptation methods, feasibility is trivially established when the active set is known and does not change with the prevailing uncertainty. However, as soon as the active set changes, tracking the current set of active constraints may lead to infeasibility. A switching logic can be used to remove this limitation, but it requires experimental gradient information to be available; the use of a barrier-penalty function approach has also been proposed [26].

If feasibility cannot be guaranteed, conservatism can be introduced in the form of constraint backoffs. Such backoffs are also introduced to enforce feasibility when some of the constraints are difficult to measure.

4.2. Gradient Estimation

Taking a system from a feasible to an optimal operating point requires accurate gradient information. In model-adaptation schemes, since the updated model is used to estimate the gradient, convergence is relatively fast. In the other two schemes, the gradient information has to be estimated experimentally, thereby slowing down convergence significantly.

Perhaps the major bottleneck in modifier- and direct-input-adaptation schemes lies in the estimation of this gradient information. The finite-difference scheme used in the original ISOPE paper [19] is known to be inefficient for large-scale, slow and noisy processes. Hence, alternative techniques have been developed, which can be classified as either *model-based approaches* or *perturbation-based approaches*.

Model-based approaches allow fast derivative computation by relying on a process model, yet only approximate derivatives are obtained. In self-optimizing control [12,21], the idea is to use a plant model to select linear combinations of outputs, the tracking of which results in "optimal" performance, also in the presence of uncertainty; in other words, these linear combinations of outputs approximate the process derivatives. Also, a way of calculating the gradient based on the theory of neighbouring extremals has been presented in [13]; however, an important limitation of this approach is that it provides only a first-order approximation and that the accuracy of the derivatives depends strongly on the reliability of the plant model.

The idea behind perturbation methods is to estimate process derivatives using variations in the operating point. Extremum-seeking control [1,14] attempts to obtain the cost

sensitivity by superposing a dither signal to the plant inputs. In dynamic model identification, the plant is approximated by a dynamic model during the transient phase between two successive steady states [16,31,11]. Since the derivatives are calculated from the identified dynamic model, the waiting time needed for reaching a new steady state is avoided. Other perturbation-based approaches, which remove the disadvantage of requiring additional dynamic perturbations, consist in using current and past (steady-state) measurements to compute a gradient estimate based on Broyden's formula [16]. For the case of multiple identical units operating in parallel, Srinivasan considered perturbations along the unit dimension rather than the time dimension, thereby allowing faster and more accurate derivative estimates [22]. In principle, the smaller the difference between the operating points, the more accurate the derivative approximation, but conditioning issues might arise due to measurement noise and plant disturbances. A way of avoiding this latter deficiency is presented in [10].

5. Discussion

In this section, we take a critical look at the three classes of adaptive optimization methods described above in terms of various criteria. We also argue in favor of modifier-adaptation methods, in the sense that they provide a parameterization that is tailored to the matching of the KKT conditions.

The analysis presented in Table 1 shows many facets of the problem. It is interesting to see that modifier-adaptation methods can be positioned between the model-adaptation methods and direct-input-tracking methods; several attractive features are shared between the first and second columns, while other features are shared between the second and third columns.

The methods differ mainly in the handles and in the measurements that are used for correction. The major drawback of model-adaptation schemes is that KKT matching is required for convergence to a (local) plant optimum, which can be very difficult to satisfy with the (arbitrary) parameterization $\boldsymbol{\theta}$ and (arbitrary) auxiliary measurements $\mathbf{y}_p$. In comparison, modifier-adaptation methods resolve the challenging task of selecting candidate parameters for adaptation by introducing the modifiers $\boldsymbol{\Lambda}$ as handles. Also, the measurements $\mathbf{K}_p$ are directly related to the KKT conditions, and their number is equal to that of the handles $\boldsymbol{\Lambda}$, i.e., there results a square update problem. Hence, since these parameters are essentially decoupled, no sophisticated technique is required for the update of $\boldsymbol{\Lambda}$. Moreover, KKT matching becomes trivial, and reaching a (local) plant optimum is guaranteed upon convergence. This leads us to argue that modifier-adaptation methods possess the "adequate" parameterization and use the "adequate" measurements" for solving optimization problems on-line.

Direct-input-adaptation methods differ from model- and modifier-adaptation methods in that a process model is not used on-line, thus removing much of the on-line complexity.

Another important element of comparison is the use of experimental gradient information. The modifier- and direct-input-adaptation methods make use of experimental gradients to guarantee (local) optimality. However, obtaining this information is usually time consuming and slows down the entire adaptation scheme.

Note that the use of an updated process model gives the ability to determine changes in the active set and typically provides faster convergence. Yet, in practice, the convergence of the model- and modifier-adaptation methods is often slowed down by the introduction of filtering that is required to avoid unstable behavior that would result because the corrections are local in nature.

	Model-adaptation methods	Modifier-adaptation methods	Direct-input adaptation methods
Adjustable parameters	$\boldsymbol{\theta}$	$\boldsymbol{\Lambda}$	$\boldsymbol{\pi}$
Dimension of parameters	n_θ	$n_g + n_\pi(n_g+1)$	n_π
Measurements	$\mathbf{y}_p$	$\mathbf{K}_p$	$\mathbf{K}_p$
Dimension of measurements	n_y	$n_g + n_\pi(n_g+1)$	$n_g + n_\pi(n_g+1)$
Update criterion	$\|\mathbf{y}-\mathbf{y}_p\|_2$	$\|\mathbf{K}-\mathbf{K}_p\|_2$	None
Exp. gradient estimation	No	Yes	Yes
Repeated optimization	Yes	Yes	No
On-line use of process model	Yes	Yes	No
Controller type	Model predictive	Model predictive	Any
Smoothening	External filter	External filter	Controller tuning
Choice of active sets	Optimization	Optimization	Switching logic
Requirement for feasibility (no gradient information)	Constraint matching	None	Correct active set
Requirement for optimality (with gradient information)	KKT matching	None	None

Table 1. Comparison of various real-time optimization schemes

6. Conclusions

This paper provides a classification of real-time optimization schemes and analyzes their ability to use measurements to track the necessary conditions of optimality of the plant. The similarities and differences between the various schemes are highlighted, and it is shown that modifier-adaptation schemes use a parameterization, measurements, and an update criterion that are tailored to the matching of KKT conditions.

To improve the performance of adaptive optimization, it may be useful to combine specific features of the various methods. For example, the combination of model adaptation (which ensures fast convergence for the first few iterations and detects changes in the active set) with direct-input adaptation (which provides the necessary gradients in the neighborhood of the plant optimum) has been demonstrated in [24]. Another interesting combination would be to use a modifier-adaptation approach at one time scale and perform model adaptation at a slower rate, thus giving rise to a two-time-scale adaptation strategy.

References

1. Ariyur K. B. and Kristic M., “Real-Time Optimization by Extremum Seeking Feedback”, Wiley, 2003.
2. Bazaraa M. S., Sherali H. D. and Shetty C. M., “Nonlinear Programming: Theory and Algorithms”, second ed., John Wiley and Sons, New York, 1993.
3. Biegler L. T., Grossmann I. E. and Westerberg A. W., “A note on approximation techniques used for process optimization”, *Comput Chem Eng* **9**(2):201-206, 1985.

4. Bonvin D. and Srinivasan B., "Optimal operation of batch processes via the tracking of active constraints", *ISA Trans* **42**(1):123-134, 2003.
5. Brdys M. A. and Tatjewski P., "Iterative Algorithms For Multilayer Optimizing Control", World Scientific Pub Co, London UK, 2005.
6. Chachuat B., Marchetti A. and Bonvin D., "Process optimization via constraints adaptation", *J Process Control*, in press.
7. Chen C. Y. and Joseph B., "On-line optimization using a two-phase approach: an application study", *Ind Eng Chem Res* **26**:1924-1930, 1987.
8. Forbes J. F. and Marlin T. E., "Model accuracy for economic optimizing controllers: the bias update case", *Ind Eng Chem Res* **33**:1919-1929, 1994.
9. François G., Srinivasan B. and Bonvin D., "Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty", *J Process Control* **15**:701-712, 2005.
10. Gao W. and Engell S., "Iterative set-point optimization of batch chromatography", *Comput Chem Eng* **29**(6):1401-1409, 2005.
11. Golden M. P. and Ydstie B. E., "Adaptive extremum control using approximate process models", *AIChE J* **35**(7):1157-1169, 1989.
12. Govatsmark M. S. and Skogestad S., "Selection of controlled variables ans robust setpoints", *Ind Eng Chem Res* **44**(7):2207-2217, 2005.
13. Gros S., Srinivasan B. and Bonvin D., "Static optimization via tracking of the necessary conditions of optimality using neighboring extremals", *Proc ACC 2005*, Portland OR, pp. 251-255, 2005.
14. Guay M. and Zang T., "Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainty", *Automatica* **39**:1283–1294, 2003.
15. de Hennin S. R., Perkins J. D. and Barton G. W., "Structural descisions in on-line optimization", *Proc Int Conf PSE'94*, pp. 297-302, 1994.
16. Mansour M. and Ellis J. E., "Comparison of methods for estimating real process derivatives in on-line optimization", *Appl Math Mod* **27**:275-291, 2003.
17. Marlin T. E. and Hrymak A. N., "Real-time operations optimization of continuous process", *Proc 5th Int Conf on Chemical Process Control (CPC-5)*, Tahoe City NV, 1997.
18. Mönnigmann M. and Marquardt W., "Steady-state process optimization with guaranteed robust stability and feasibility", *AIChE J* **49**(12):3110-3126, 2003.
19. Morari M., Stephanopoulos G. and Arkun Y., "Studies in the synthesis of control structures for chemical processes, Part I", *AIChE J* **26**(2):220-232, 1980.
20. Roberts P. D., "An algorithm for steady-state system optimization and parameter estimation", *Int J Syst Sci* **10**:719-734, 1979.
21. Skogestad S. "Plantwide control: The search for the self-optimizing control structure". *J Process Control* **10**:487–507, 2000.
22. Srinivasan B., "Real-time optimization of dynamic systems using multiple units", *Int J Robust Nonlinear Control* **17**:1183–1193, 2007.
23. Srinivasan B. and Bonvin D., "Interplay between identification and optimization in run-to-run optimization schemes", *Proc ACC 2002*, Anchorage AK, pp. 2174–2179, 2002.
24. Srinivasan B. and Bonvin D., "Convergence analysis of iterative identification and optimization schemes", *Proc ACC 2003*, Denver CO, pp. 1956-1961, 2003.
25. Srinivasan B., Primus C. J, Bonvin D. and Ricker N. L., "Run-to-run Optimization via Constraint Control", *Control Eng Pract* **9**(8):911-919, 2001.
26. Srinivasan B., Biegler L.T. and Bonvin D., "Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function", *Comput Chem Eng* **32**(3):572-579, 2008.
27. Stephanopoulos G. and Arkun Y., "Studies in the synthesis of control structures for chemical processes, Part IV", *AIChE J* **26**(6):975-991, 1980.
28. Tatjewski P., "Iterative optimizing set-point control–The basic principle redesigned", *Proc 15th IFAC World Congress*, Barcelona, 2002.

29. Woodward L., Perrier M. and Srinivasan B., "Multi-unit optimization with gradient projection on active constraints", *Proc 8th Int Symp on Dynamics and Control of Process Systems (DYCOPS)*, Vol 1, pp. 129-134, 2007.
30. Zhang Y. and Forbes J. F., "Extended design cost: A performance criterion for real-time optimization systems", *Comput Chem Eng* **24**:1829-1841, 2000.
31. Zhang Y. and Forbes J. F., "Performance analysis of perturbation-based methods for real-time optimization", *Can J Chem Eng* **84**:209-218, 2006.