

# Real-time Optimization with Estimation of Experimental Gradients

A. Marchetti\* B. Chachuat\*\* D. Bonvin\*

\* *Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne (EPFL), Station 9, CH-1015 Lausanne, Switzerland*

\*\* *Department of Chemical Engineering, McMaster University, 1280 Main Street West, Hamilton, ON L8S 4LT, Canada*

---

**Abstract:** For good performance in practice, real-time optimization schemes need to be able to deal with the inevitable plant-model mismatch problem. Unlike the two-step schemes combining parameter estimation and optimization, the modifier-adaptation approach uses experimental gradient information and does not require the model parameters to be estimated on-line. The dual modifier-adaptation approach presented in this paper drives the process towards optimality, while paying attention to the accuracy of the estimated gradients. The gradients are estimated from successive operating points generated by the optimization algorithm. The novelty lies in the development of an upper bound on the norm of the gradient errors, which is used as a constraint when determining the next operating point. The proposed approach is demonstrated in simulation via the real-time optimization of a continuous reactor.

*Keywords:* Real-time optimization, estimation of experimental gradients, modifier adaptation.

---

## 1. INTRODUCTION

Real-time optimization (RTO) of continuous plants aims at improving some steady-state performance index (Marlin and Hrymak [1997]). Since the majority of RTO schemes uses a model of the plant, reaching optimal performance in the presence of plant-model mismatch is a difficult task, which necessitates adaptation based on measured information. Chachuat et al. [2009] proposed a three-way classification of RTO schemes. One class includes the so-called modifier-adaptation approach (Marchetti et al. [2009]), whereby appropriate terms are added to the optimization problem and identified so that the KKT conditions of the model match those of the plant. In this context, the modifier-adaptation approach requires to be able to estimate on-line the experimental gradients, i.e., the derivatives of the plant outputs with respect to the inputs. This paper investigates the estimation of experimental gradients and their use in modifier-adaptation schemes.

A comparison of different approaches for on-line gradient estimation is given in Mansour and Ellis [2003]. Finite-difference techniques can be used to estimate the gradients experimentally. The most straightforward approach consists in perturbing each input individually around the current operating point to get an estimate of the corresponding gradient elements. This is the case, e.g., when forward finite differencing (FFD) is applied at each RTO iteration. An alternative approach, which was introduced in the ISOPE (Integrated System Optimization and Parameter Estimation) literature under the name *dual ISOPE*, is to estimate the gradients based on the current and past operating points (Brdyś and Tatjewski [1994, 2005]). The key issue therein is the ability to estimate the experimental gradients reliably while updating the operating point. Indeed, there are two conflicting objec-

tives: the “primal objective” consists in solving the optimization problem, while the “dual objective” aims at estimating accurate gradients. These conflicting tasks can be accommodated by adding a constraint in the optimization problem so as to ensure sufficiently rich information in the measurements and guarantee gradient accuracy. Brdyś and Tatjewski [1994, 2005] proposed a constraint that prevents ill-conditioning in gradient computation. The present paper goes further and investigates the two main sources of errors, namely the error introduced by numerical approximation of a derivative (truncation error) and measurement noise. A constraint that enforces an upper bound on the gradient error norm is proposed. Since the constraint for ensuring sufficient information might compromise optimality in the vicinity of the optimum, Gao and Engell [2005] suggested using the ill-conditioning measure not to constrain the optimization problem but rather to determine whether an additional input perturbation is needed. Note that such a scheme could also be used in the context of the dual-modifier approach proposed here.

The paper is organized as follows. Section 2 formulates the optimization problem. The modifier-adaptation scheme is reviewed in Section 3. Analysis of the errors in the gradient estimates obtained from past operating points is carried out in Section 4. Based on this analysis, Section 5 proposes a norm-based constraint, which is incorporated into the dual modifier-adaptation algorithm presented in Section 6. The approach is illustrated via the reactor of the Williams-Otto plant in Section 7, and Section 8 concludes the paper and presents directions for future work.

## 2. PROBLEM FORMULATION

For the sake of simplicity, an unconstrained optimization problem is considered throughout. This way, only the

gradient of the objective function needs to be estimated, while in the constrained case, the constraint gradients would need to be evaluated as well. A possible way of tackling constrained optimization problems will be sketched in the conclusion section.

The unconstrained optimization problem reads:

$$\min_{\mathbf{u}} \Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \quad (1)$$

where  $\mathbf{u} \in \mathbb{R}^{n_u}$  are the decision (or input) variables,  $\mathbf{y}_p \in \mathbb{R}^{n_y}$  are the measured (or output) variables, and  $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$  is the scalar cost function to be minimized. The notation  $(\cdot)_p$  will be used for the variables that are associated with the plant. Also, it is assumed that  $\phi(\mathbf{u}, \mathbf{y}_p)$  is a known function of  $\mathbf{u}$  and  $\mathbf{y}_p$ . On the other hand, the steady-state input-output mapping of the plant,  $\mathbf{y}_p(\mathbf{u})$ , is typically unknown, and only the approximate model  $f(\mathbf{u}, \mathbf{y}, \boldsymbol{\theta}) = \mathbf{0}$  is available, where  $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$  is the set of model parameters. Assuming that the model outputs  $\mathbf{y}$  can be expressed explicitly as functions of  $\mathbf{u}$  and  $\boldsymbol{\theta}$ , the cost function predicted by the model becomes  $\Phi(\mathbf{u}, \boldsymbol{\theta}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}))$ .

It is furthermore assumed that the decision variables  $\mathbf{u}$  are of the same order of magnitude, which can be achieved via scaling. For example, if the decision variable  $u_i$  remains within the interval  $[u_{i,a}, u_{i,b}]$ , it can be scaled as  $u_i^{\text{scaled}} = (u_i - u_{i,a}) / (u_{i,b} - u_{i,a})$ . For notional simplicity, the superscript indicating a scaled variable will be omitted in the sequel.

### 3. MODIFIER-ADAPTATION SCHEME

In the modifier-adaptation scheme, a gradient-correction term is added to the cost function of the model-based optimization problem (Marchetti et al. [2009]). At the  $k$ th iteration, the next input  $\mathbf{u}_{k+1}$  is obtained as:

$$\mathbf{u}_{k+1} = \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) := \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^\top \mathbf{u} \quad (2)$$

where  $\boldsymbol{\lambda}_k$  is the cost-gradient modifier at the  $k$ th iteration. This modifier is adapted at each iteration based on the difference between the gradient of the plant and that predicted by the model. For example, upon implementation of a first-order exponential filter, the gradient modifier is calculated as:

$$\boldsymbol{\lambda}_k^\top = (1 - d)\boldsymbol{\lambda}_{k-1}^\top + d \left[ \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta}) \right] \quad (3)$$

with the filter gain  $d \in (0, 1]$ . Computation of the modifier  $\boldsymbol{\lambda}_k$  requires the knowledge of the plant gradient  $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k)$ .

An appealing property of the modifier-adaptation scheme is that, upon convergence and in the absence of noise, the optimum  $\mathbf{u}_\infty$  for the modified model-based optimization problem (2) satisfies the first-order necessary conditions of optimality of the optimization problem (1) (Marchetti et al. [2009]). Note that this is the case despite plant-model mismatch. Note also that the need to match the plant outputs  $\mathbf{y}_p(\mathbf{u})$  by means of a parameter estimation problem, as this is the case for the ISOPE modifier (Roberts [1979]), is removed. However, the downside of modifier adaptation lies in the need to estimate the experimental gradient  $\frac{\partial \Phi_p}{\partial \mathbf{u}}$ .

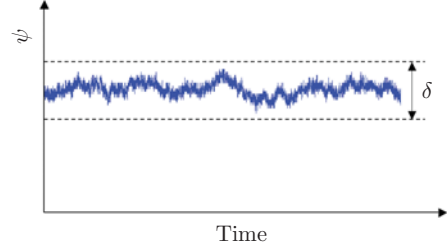


Fig. 1. Noisy cost function at steady state.

### 4. EXPERIMENTAL GRADIENT COMPUTED FROM PAST OPERATING POINTS

It is assumed that the cost function can be evaluated from the noisy output measurements as follows:

$$\psi(\mathbf{u}) = \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) = \Phi_p(\mathbf{u}) + v \quad (4)$$

where  $\boldsymbol{\nu}$  is the measurement noise on the outputs and  $v$  the induced noise in the cost estimates. Note that, even if  $\boldsymbol{\nu}$  is zero mean,  $v$  might have a nonzero mean if the function  $\phi(\mathbf{u}, \mathbf{y})$  is nonlinear in  $\mathbf{y}$ .

The forthcoming analysis is conducted assuming that the cost estimates remain within the interval  $\delta$  at steady-state operation, as illustrated in Figure 1. Based on a statistical description of  $v$ ,  $\delta$  could be selected by considering a desired confidence interval. Values that fall outside the selected confidence interval can simply be discarded.

Consider the  $k$ th iteration and the  $n_u$  past operating points,  $\mathbf{u}_{k-j}$ ,  $j = 0, \dots, n_u - 1$ , and let us evaluate the cost as a function of the next operating point, which will generically be labeled  $\mathbf{u}$ . Using a first-order approximation of  $\Phi_p(\mathbf{u}_{k-j})$  in the neighborhood of  $\mathbf{u}$ , the value of  $\psi$  at the past operating points is given by:

$$\begin{aligned} \psi(\mathbf{u}_{k-j}) &= \Phi_p(\mathbf{u}_{k-j}) + v_{k-j} = (\psi(\mathbf{u}) - v) \\ &+ \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})[\mathbf{u}_{k-j} - \mathbf{u}] + O(\|\mathbf{u}_{k-j} - \mathbf{u}\|^2) + v_{k-j} \end{aligned} \quad (5)$$

and, by neglecting the higher-order and noise terms:

$$\psi(\mathbf{u}_{k-j}) = \psi(\mathbf{u}) + \hat{\boldsymbol{\beta}}(\mathbf{u})[\mathbf{u}_{k-j} - \mathbf{u}], \quad (6)$$

where  $\hat{\boldsymbol{\beta}}(\mathbf{u})$  is an estimate of the experimental cost gradient  $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})$ .  $\hat{\boldsymbol{\beta}}(\mathbf{u})$  can be computed from the  $n_u$  past operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$  and the corresponding noisy cost values  $\psi(\mathbf{u}_k), \dots, \psi(\mathbf{u}_{k-n_u+1})$  by writing (6) in the following matrix form (Brdyś and Tatjewski [2005]):

$$\hat{\boldsymbol{\beta}}(\mathbf{u}) = \mathcal{Y}(\mathbf{u}) \mathcal{U}^{-1}(\mathbf{u}) \quad (7)$$

with

$$\mathcal{U}(\mathbf{u}) := [\mathbf{u} - \mathbf{u}_k \ \dots \ \mathbf{u} - \mathbf{u}_{k-n_u+1}] \in \mathbb{R}^{n_u \times n_u} \quad (8)$$

$$\mathcal{Y}(\mathbf{u}) := [\psi(\mathbf{u}) - \psi(\mathbf{u}_k) \ \dots \ \psi(\mathbf{u}) - \psi(\mathbf{u}_{k-n_u+1})] \quad (9)$$

The gradient error is defined as  $\boldsymbol{\epsilon}(\mathbf{u}) := \hat{\boldsymbol{\beta}}(\mathbf{u}) - \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})$ , which, from (7) together with (4), can be split as  $\boldsymbol{\epsilon}(\mathbf{u}) = \boldsymbol{\epsilon}^t(\mathbf{u}) + \boldsymbol{\epsilon}^n(\mathbf{u})$ , with:

$$\boldsymbol{\epsilon}^t(\mathbf{u}) = [\Phi_p(\mathbf{u}) - \Phi_p(\mathbf{u}_k) \ \dots \quad (10)$$

$$\dots \ \Phi_p(\mathbf{u}) - \Phi_p(\mathbf{u}_{k-n_u+1})] \mathcal{U}^{-1}(\mathbf{u}) - \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u})$$

$$\boldsymbol{\epsilon}^n(\mathbf{u}) = [v - v_k \ \dots \ v - v_{k-n_u+1}] \mathcal{U}^{-1}(\mathbf{u}) \quad (11)$$

where  $\epsilon^t$  and  $\epsilon^n$  represent the errors due to truncation and noise, respectively. Next, we investigate these two components of the gradient error.

*Gradient Error due to Truncation.* An upper bound on the norm of this error is given in the next proposition.

*Proposition 1.* Let  $\Phi_p(\mathbf{u})$  be twice continuously differentiable with respect to  $\mathbf{u}$ . Then, given the  $n_u$  past operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$ , an upper bound on  $\|\epsilon^t(\mathbf{u})\|$  is given by

$$\|\epsilon^t(\mathbf{u})\| \leq \mathcal{E}^t(\mathbf{u}), \quad (12)$$

with

$$\mathcal{E}^t(\mathbf{u}) := \frac{d_2}{2} \left\| \left[ (\mathbf{u} - \mathbf{u}_k)^\top (\mathbf{u} - \mathbf{u}_k) \dots \dots (\mathbf{u} - \mathbf{u}_{k-n_u+1})^\top (\mathbf{u} - \mathbf{u}_{k-n_u+1}) \right] \mathcal{U}^{-1}(\mathbf{u}) \right\| \quad (13)$$

where  $d_2$  is the largest absolute eigenvalue of the Hessian of  $\Phi_p(\cdot)$ .

**Proof.** By Taylor series expansion of  $\Phi_p(\mathbf{u}_{k-j})$  at  $\mathbf{u}$  and upper bounding of the norm of the Hessian of  $\Phi_p$  [Marchetti, 2009].  $\square$

Note that  $d_2$  represents an upper bound on the curvature of  $\Phi_p(\cdot)$ .

*Gradient Error due to Measurement Noise.* For relating the error norm  $\|\epsilon^n(\mathbf{u})\|$  to the location of the new operating point, the concepts of affine subspaces and distance between complement affine subspaces will be used (see Appendix A for a brief review of these concepts).

The largest possible value of  $\|\epsilon^n(\mathbf{u})\|$ , noted  $\|\epsilon^n(\mathbf{u})\|_{\max}$ , is computed in the next proposition.

*Proposition 2.* Given the  $n_u$  past operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$  and the interval  $\delta$  for the noisy function  $\psi(\cdot)$ , the largest possible value of  $\|\epsilon^n(\mathbf{u})\|$  is

$$\|\epsilon^n(\mathbf{u})\|_{\max} = \delta / l_{\min}(\mathbf{u}) \quad (14)$$

where  $l_{\min}(\mathbf{u})$  is the shortest distance between all possible pairs of complement affine subspaces that can be generated from  $\mathcal{S} = \{\mathbf{u}, \mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}\}$ .

**Proof.** The proof proceeds in two parts: (i) the largest error occurs when the error  $v$  is either  $\delta/2$  for some of the operating points and  $-\delta/2$  for the other points, with each set of points defining an affine subspace; and (ii) the error vector  $\epsilon^n(\mathbf{u})$  is normal to both affine subspaces, which results in the largest possible error norm given by (14) [Marchetti, 2009].  $\square$

## 5. UPPER BOUND ON GRADIENT ERROR

A bound on the condition number of the matrix  $\mathcal{U}(\mathbf{u})$  was proposed in Brdyś and Tatjewski [1994, 2005]. This bound ensures that the new operating point does not introduce large errors in the gradient estimates due to ill-conditioning of  $\mathcal{U}(\mathbf{u})$ . However, the bound is not directly related to the errors resulting from truncation and measurement noise. This section introduces a consistent, although possibly conservative, upper bound on the gradient error norm.

Consider the desired upper bound  $\mathcal{E}^U$  on the gradient error norm:

$$\|\epsilon(\mathbf{u})\| \leq \|\epsilon^t(\mathbf{u})\| + \|\epsilon^n(\mathbf{u})\| \leq \mathcal{E}^U \quad (15)$$

Given the  $n_u$  past operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$ , the following theorem provides a sufficient condition for the location of  $\mathbf{u}$  so as to satisfy (15).

*Theorem 1.* (Sufficient condition for gradient accuracy). The gradient error norm  $\|\epsilon(\mathbf{u})\|$  does not exceed the desired upper bound  $\mathcal{E}^U$  by choosing  $\mathbf{u}$  that satisfies

$$\mathcal{E}(\mathbf{u}) := \mathcal{E}^t(\mathbf{u}) + \|\epsilon^n(\mathbf{u})\|_{\max} \leq \mathcal{E}^U, \quad (16)$$

with  $\mathcal{E}^t(\mathbf{u})$  and  $\|\epsilon^n(\mathbf{u})\|_{\max}$  given by (13) and (14), respectively.

**Proof.** The proof follows from (15), inequality (12) and the fact that  $\|\epsilon^n(\mathbf{u})\| \leq \delta / l_{\min}(\mathbf{u})$  from (14).  $\square$

For given values of  $\delta$  and  $d_2$ , there is a minimal value that  $\mathcal{E}(\mathbf{u})$  can take. Hence,  $\mathcal{E}^U$  should be selected larger than this minimal value for the constraint (16) to be feasible.

*Example 1.* Consider the two-dimensional case ( $n_u = 2$ ) with  $\delta = 0.2$ ,  $d_2 = 2$  and the past operating points  $\mathbf{u}_k = [0 \ -0.5]^\top$  and  $\mathbf{u}_{k-1} = [0 \ 0.5]^\top$ . The upper bounds  $\mathcal{E}^t(\mathbf{u})$  and  $\|\epsilon^n(\mathbf{u})\|_{\max}$  can be evaluated in terms of the location of the new operating point  $\mathbf{u} = [u_1 \ u_2]^\top$ . Figures 2a-c show the contours of the error norms  $\mathcal{E}^t(\mathbf{u})$ ,  $\|\epsilon^n(\mathbf{u})\|_{\max}$  and  $\mathcal{E}(\mathbf{u})$ . It is seen that (i) both  $\mathcal{E}^t(\mathbf{u})$  and  $\|\epsilon^n(\mathbf{u})\|_{\max}$  increase as  $\mathcal{U}(\mathbf{u})$  becomes more ill-conditioned ( $\mathbf{u}$  aligned with  $\mathbf{u}_k$  and  $\mathbf{u}_{k-1}$ ), and (ii) the two regions generated by the constraint (16) are nonconvex.

*Convex Constraint.* The objective being to use the constraint (16) in the optimization problem (2), the fact that this constraint is nonconvex creates the possibility of multiple local solutions. Next, we introduce a tight relaxation that makes the constraint convex.

It can be seen that, for a given error level  $c$ , the expression  $\mathcal{E}^t(\mathbf{u}) = c$  generates two  $(n_u - 1)$ -dimensional spheres of radius  $r = \frac{c}{d_2}$ . The centers of these spheres are located symmetrically on each side of the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$  generated by the  $n_u$  past operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$ . Considering the new operating point  $\mathbf{u}$  located on the sphere, the center point is given by

$$\mathbf{u}_c^\top(\mathbf{u}) = \frac{1}{2} \left[ \mathbf{u}^\top \mathbf{u} - \mathbf{u}_k^\top \mathbf{u}_k \dots \dots \mathbf{u}^\top \mathbf{u} - \mathbf{u}_{k-n_u+1}^\top \mathbf{u}_{k-n_u+1} \right] \mathcal{U}^{-1}(\mathbf{u}). \quad (17)$$

It can be shown that  $\|\epsilon^n(\mathbf{u})\|_{\max}$  is convex on each side of the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$  (see also Figure 2b). Hence, non-convexity of the constraint (16) is due to the part of the aforementioned spheres that crosses the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$ . The distance (positive or negative) from the center point  $\mathbf{u}_c(\mathbf{u})$  to the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$  is given by:

$$l_C(\mathbf{u}) = \frac{b_k - \mathbf{n}_k^\top \mathbf{u}_c(\mathbf{u})}{\|\mathbf{n}_k\|}. \quad (18)$$

Given the  $n_u$  operating points  $\mathbf{u}_k, \dots, \mathbf{u}_{k-n_u+1}$ , the point  $\mathbf{u}_{m,k}$  can be obtained by projecting the center point  $\mathbf{u}_c(\mathbf{u})$  on the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$ :

$$\mathbf{u}_{m,k} = \mathbf{u}_c(\mathbf{u}) + \frac{l_C(\mathbf{u})}{\|\mathbf{n}_k\|} \mathbf{n}_k \quad (19)$$

It appears that  $\mathbf{u}_{m,k}$  is independent of  $\mathbf{u}$ . For a given upper bound  $\mathcal{E}^U$ , it is then possible to define convex feasible

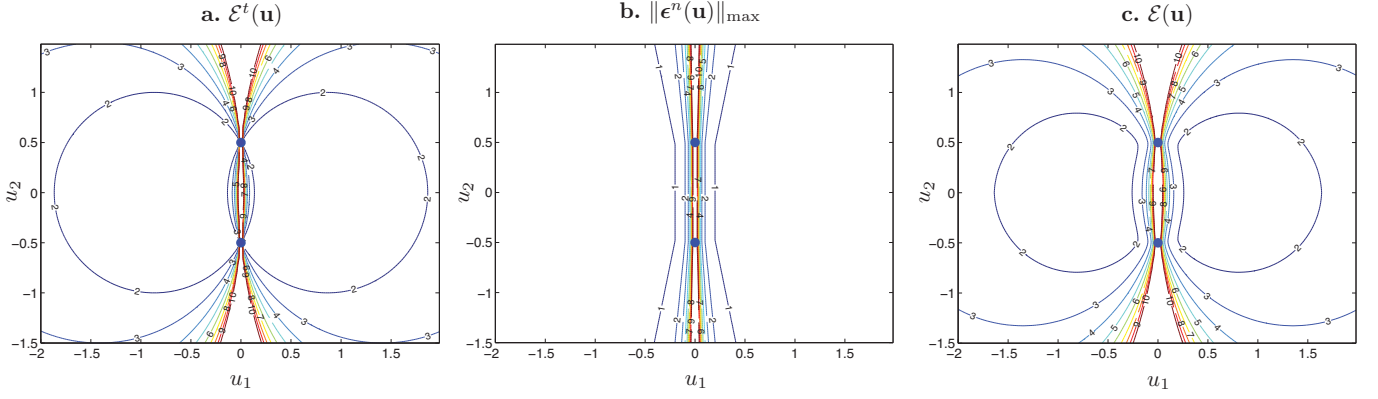


Fig. 2. Contour maps for the norm of the gradient error due to (a) truncation error, (b) measurement noise, and (c) total error.

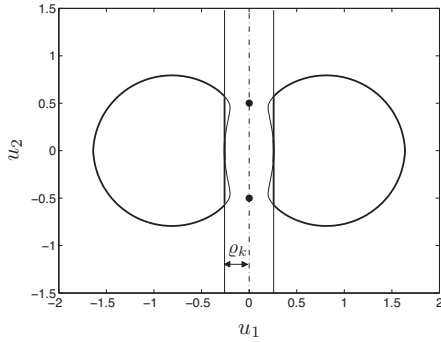


Fig. 3. Convex regions (in bold) corresponding to the constraint  $\mathcal{E}(\mathbf{u}) \leq 2$ .

regions by adding constraints expressing the minimal distance between the new operating point  $\mathbf{u}$  and the hyperplane, which eliminates the non-convex part of the regions generated by (16), as illustrated in Figure 3. The minimal point-to-hyperplane distance  $\varrho_k$  can be determined numerically by finding the smallest absolute value solution of the following equation:

$$\mathcal{E}\left(\mathbf{u}_{m,k} + \frac{\varrho_k}{\|\mathbf{n}_k\|} \mathbf{n}_k\right) = \mathcal{E}^U$$

## 6. DUAL MODIFIER-ADAPTATION SCHEME

The dual modifier-adaptation scheme proposed in this section uses the upper bound on the gradient error defined in Section 5 as a constraint in the optimization problem (2). On each side of the hyperplane  $\mathbf{n}_k^\top \mathbf{u} = b_k$  generated by the  $n_u$  past operating points, a modified model-based optimization problem is solved. The optimization problem corresponding to the half space  $\mathbf{n}_k^\top \mathbf{u} > b_k$  reads:

$$\begin{aligned} \mathbf{u}_{k+1}^+ &= \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^\top \mathbf{u} \\ \text{s.t. } \mathcal{E}(\mathbf{u}) &\leq \mathcal{E}^U, \quad \mathbf{n}_k^\top \mathbf{u} > b_k + \varrho_k \|\mathbf{n}_k\| \end{aligned} \quad (20)$$

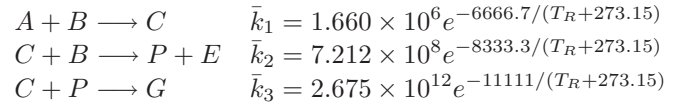
while, for the half space  $\mathbf{n}_k^\top \mathbf{u} < b_k$ , one has:

$$\begin{aligned} \mathbf{u}_{k+1}^- &= \arg \min_{\mathbf{u}} \Phi_m(\mathbf{u}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^\top \mathbf{u} \\ \text{s.t. } \mathcal{E}(\mathbf{u}) &\leq \mathcal{E}^U, \quad \mathbf{n}_k^\top \mathbf{u} < b_k - \varrho_k \|\mathbf{n}_k\| \end{aligned} \quad (21)$$

The modifiers  $\boldsymbol{\lambda}_k^\top$  are adapted as in (3). The next operating point is chosen as the value of  $\{\mathbf{u}_{k+1}^+, \mathbf{u}_{k+1}^-\}$  that minimizes the augmented cost function  $\Phi_m(\mathbf{u}, \boldsymbol{\theta})$ .

## 7. OPTIMIZATION OF REACTOR OPERATION

The reactor in the Williams-Otto plant (Williams and Otto [1960]), as modified by Roberts [1979], is used to illustrate the dual modifier-adaptation scheme. This reactor example has also been used to illustrate model adequacy and RTO performance (Forbes et al. [1994], Zhang and Forbes [2000]). It consists of an ideal CSTR in which the following reactions occur:



where the reactants  $A$  and  $B$  are fed with the mass flowrates  $F_A$  and  $F_B$ , respectively. The desired products are  $P$  and  $E$ .  $C$  is an intermediate product and  $G$  is an undesired product. The product stream has the mass flowrate  $F = F_A + F_B$ . Operation is isothermal at the temperature  $T_R$ . The reactor mass holdup is 2105 kg.

The objective is to maximize profit, which is expressed as the cost difference between the products and the reactants:

$$\phi(\mathbf{u}, \mathbf{y}) = 1143.38 X_P F + 25.92 X_E F - 76.23 F_A - 114.34 F_B$$

The flowrate of reactant  $A$  is fixed at 1.8275 kg/s. The flowrate of reactant  $B$  and the reactor temperature are the decision variables, thus  $\mathbf{u} = [F_B \ T_R]^\top$ .

In this example, the aforementioned reaction scheme corresponds to the simulated reality. However, since it is assumed that the reaction scheme is not well understood, the following two reactions have been proposed to model the system (Forbes et al. [1994]):



The material balance equations for the plant and the approximate model can be found in Zhang and Forbes [2000].

The inputs are scaled using the intervals [3, 6] for  $F_B$ , and [70, 100] for  $T_R$ . In this range, the maximal value of  $d_2$  obtained with the scaled inputs is  $d_2 = 1030$  for the model, whereas the (unknown) plant value is  $d_2 = 1221$ . The simulations are carried out assuming that the noise  $v$  has a Gaussian distribution with standard deviation  $\sigma_\phi = 0.5$ . The noise interval  $\delta = 3$  is chosen. The exponential filter (3) is implemented with  $d = 0.5$ .

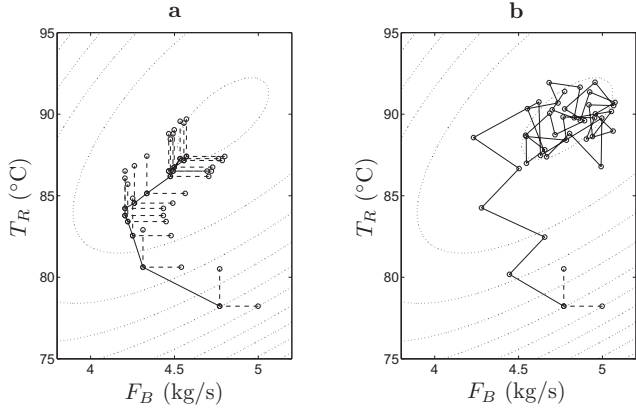


Fig. 4. Input trajectories for 45 operating points. The dotted lines represent the contours of the plant cost function. (a) Modifier adaptation using FFD. (b) Dual modifier adaptation with bound on gradient error.

*Modifier Adaptation using FFD.* First, modifier adaptation is applied using the FFD approach, which consists in perturbing the inputs one at the time from the current operating point with the fixed step size  $h$ . The gradient error norm, which is a function of  $h$ , is found to be minimal for  $h^* = 0.0763$  (scaled value). The corresponding gradient error norm  $\mathcal{E}^t(h^*) + \|\epsilon^n(h^*)\|_{\max}$  is 111.2 (Marchetti [2009]). Figure 4a shows a resulting input trajectory. The observed offset with respect to the plant optimum results mainly from the gradient error due to truncation.

*Dual Modifier Adaptation with Bound on Gradient Error.*

Dual modifier adaptation is applied with  $\mathcal{E}^U = 111.2$  (same value as above). The algorithm is initialized using FFD. Figure 4b shows a resulting input trajectory. Compared with modifier adaptation using FFD, significantly fewer operating points are required to approach the optimum.

Figure 5a shows the evolution of the plant profit and the gradient error norm for 20 noise realizations. At iteration 20, the flowrate  $F_A$  is increased from 1.8275 kg/s to 2.2 kg/s. Modifier adaptation tracks the change in the plant optimum. It can be seen in the upper plot of Figure 5a that the neighborhood of the new optimal profit is reached within 6 iterations for all 20 realizations. Also, the lower plot of Figure 5a shows that the gradient error norm is kept below  $\mathcal{E}^U$ . The observed peak in gradient error occurring at iterations 21 and 22 is due to the fact that, at these points, the gradient is inconsistent in that it is estimated using operating points with different values of  $F_A$ .

*Dual Modifier Adaptation with Bound on Condition Number.* For the sake of comparison, dual modifier adaptation is also applied with a lower bound on the inverse condition number of  $\mathcal{U}(\mathbf{u})$ , as proposed in Brdyś and Tatjewski [1994, 2005]. The results are shown in Figure 5b. A lower bound of 0.4 gives an adaptation that is similar to that using the gradient error bound in the first iterations. However, as soon as the neighborhood of the plant optimum is reached, the distance between the operating points decreases, and the gradient estimates become much less accurate. Furthermore, the feasible regions given by the condition number constraint decrease proportionally

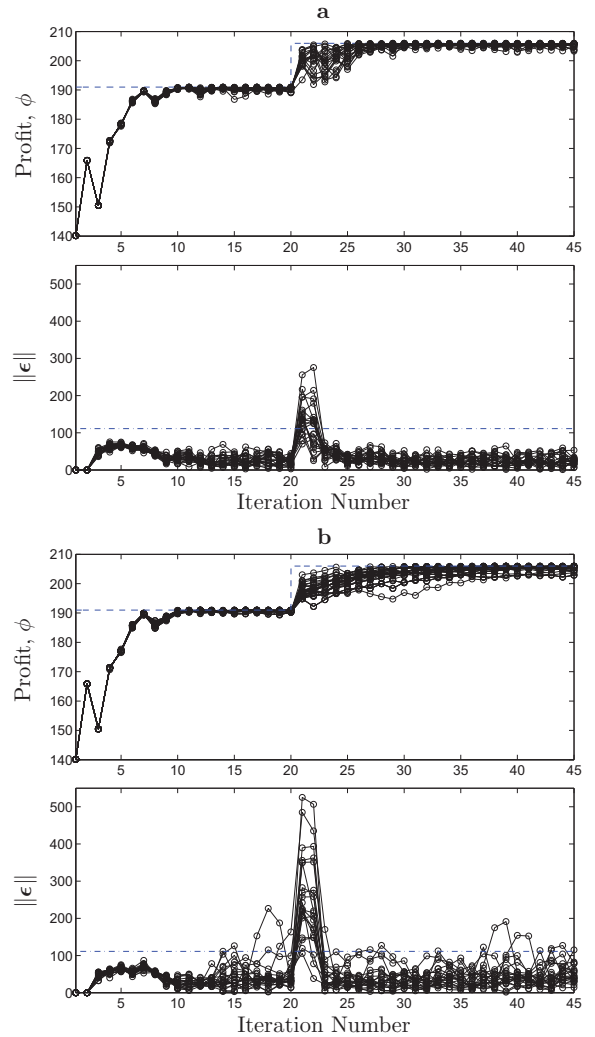


Fig. 5. Optimization for 20 noise realizations; there is a flowrate change at iteration 20. (a) Dual modifier adaptation with bound on gradient error norm. (b) Dual modifier adaptation with bound on the condition number. Dashed line: Optimal profit for the plant. Dash-dotted line:  $\mathcal{E}^U = 111.2$ .

to the distance between points. This appropriately prevents taking large steps in the wrong direction, but it also appears less suitable for tracking a changing optimum.

## 8. CONCLUSIONS

This study has demonstrated the potential of dual modifier adaptation, which pays attention to the accuracy with which the gradients are estimated. The results of the case study indicate that this approach, wherein the gradient error norm is bounded, produces more accurate gradient estimates than with simply bounding the condition number of  $\mathcal{U}(\mathbf{u})$ , i.e. a measure of the relative position of the successive inputs. In addition, the proposed scheme seems more capable of tracking a changing optimum. The performance depends on the amount of plant-model mismatch, the noise level, the estimated curvature of the cost function  $d_2$ , and the filter parameter  $d$ .

Future work will consider the extension of this approach to constrained optimization problems. In this case, mod-

ifier adaptation will require an estimate of the cost and constraint gradients of the plant to be available at each iteration. In order to be able to use the upper bound on the gradient error developed in this paper, a possible way is to associate the parameters  $\delta$  and  $\mathbf{d}_2$  to a Lagrangian function, which represents a linear combination of the cost and constraint functions.

## REFERENCES

- Brdyś, M. and Tatjewski, P. (1994). An algorithm for steady-state optimizing dual control of uncertain plants. In *Proc. 1st IFAC Workshop on New Trends in Design of Control Systems*, 249–254. Smolenice, Slovakia.
- Brdyś, M. and Tatjewski, P. (2005). *Iterative Algorithms for Multilayer Optimizing Control*. Imperial College Press, London UK.
- Chachuat, B., Srinivasan, B., and Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Comp. Chem. Eng. (in press)*.
- Forbes, J.F., Marlin, T.E., and MacGregor, J.F. (1994). Model adequacy requirements for optimizing plant operations. *Comp. Chem. Eng.*, 18(6), 497–510.
- Gao, W. and Engell, S. (2005). Iterative set-point optimization of batch chromatography. *Comp. Chem. Eng.*, 29, 1401–1409.
- Mansour, M. and Ellis, J.E. (2003). Comparison of methods for estimating real process derivatives in on-line optimization. *App. Math. Modelling*, 27, 275–291.
- Marchetti, A. (2009). *Modifier-Adaptation Methodology for Real-Time Optimization*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne (in preparation).
- Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.*, DOI: 10.1021/ie801352x.
- Marlin, T.E. and Hrymak, A.N. (1997). Real-time operations optimization of continuous processes. In *AIChE Symposium Series - CPC-V*, volume 93, 156–164.
- Roberts, P.D. (1979). An algorithm for steady-state system optimization and parameter estimation. *Int. J. Systems Sci.*, 10, 719–734.
- Williams, T.J. and Otto, R.E. (1960). A generalized chemical processing model for the investigation of computer control. *AIEE Trans.*, 79, 458.
- Zhang, Y. and Forbes, J.F. (2000). Extended design cost: A performance criterion for real-time optimization systems. *Comp. Chem. Eng.*, 24, 1829–1841.

## Appendix A. AFFINE SUBSPACES

In a  $n_u$ -dimensional space, a point is an affine subspace of dimension 0, a line is an affine subspace of dimension 1, and a plane is an affine subspace of dimension 2. An affine subspace of dimension  $(n_u - 1)$  is called an hyperplane.

*Hyperplane.* An hyperplane in  $n_u$ -dimensional space is given by

$$n_1 u_1 + n_2 u_2 + \dots + n_{n_u} u_{n_u} = b, \quad \text{or:} \quad \mathbf{n}^T \mathbf{u} = b \quad (\text{A.1})$$

and divides the space into two half-spaces:  $\mathbf{n}^T \mathbf{u} > b$ , and  $\mathbf{n}^T \mathbf{u} < b$ .

*Complement affine subspaces.* Given a set of  $(n_u + 1)$  points in a  $n_u$ -dimensional space,  $\mathcal{S} := \{\mathbf{u}_1, \dots, \mathbf{u}_{n_u+1}\}$ ,

a proper subset  $\mathcal{S}^A$ , i.e.  $\mathcal{S}^A \subsetneq \mathcal{S}$ , of  $n_u^A \in \{1, \dots, n_u\}$  points generates an affine subspace of dimension  $(n_u^A - 1)$ :

$$\mathbf{u} = \mathbf{u}_1 + \lambda_{1,2} \frac{\mathbf{u}_1 - \mathbf{u}_2}{\|\mathbf{u}_1 - \mathbf{u}_2\|} + \dots + \lambda_{1,n_u^A} \frac{\mathbf{u}_1 - \mathbf{u}_{n_u^A}}{\|\mathbf{u}_1 - \mathbf{u}_{n_u^A}\|} \quad (\text{A.2})$$

where the parameters  $\lambda_{1,2}, \dots, \lambda_{1,n_u^A}$  represent distances from the point  $\mathbf{u}_1$  in the directions  $\mathbf{u}_1 - \mathbf{u}_2, \dots, \mathbf{u}_1 - \mathbf{u}_{n_u^A}$ , respectively. The complement subset  $\mathcal{S}^C := \mathcal{S} \setminus \mathcal{S}^A$  of  $(n_u + 1 - n_u^A)$  points, generates the complement affine subspace of dimension  $(n_u - n_u^A)$ :

$$\mathbf{u} = \mathbf{u}_{n_u^A+1} + \lambda_{n_u^A+1,n_u^A+2} \frac{\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2}}{\|\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2}\|} + \dots \quad (\text{A.3})$$

$$\dots + \lambda_{n_u^A+1,n_u+1} \frac{\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1}}{\|\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1}\|}$$

*Distance between complement affine subspaces.*

*Definition 1.* (Distance between complement affine subspaces). Given a set of  $(n_u + 1)$  points in a  $n_u$ -dimensional space,  $\mathcal{S} := \{\mathbf{u}_1, \dots, \mathbf{u}_{n_u+1}\}$ , a proper subset of  $\mathcal{S}$ ,  $\mathcal{S}^A \subsetneq \mathcal{S}$  of  $n_u^A \in \{1, \dots, n_u\}$  points, and its complement  $\mathcal{S}^C := \mathcal{S} \setminus \mathcal{S}^A$  of  $(n_u + 1 - n_u^A)$  points, the distance between complement affine subspaces is defined as the (orthogonal) distance between the affine subspace of dimension  $(n_u^A - 1)$  generated by all the points in  $\mathcal{S}^A$ , and the affine subspace of dimension  $(n_u - n_u^A)$  generated by all the points in  $\mathcal{S}^C$ .

The total number of possible pairs of complement affine subspaces that can be generated from  $\mathcal{S}$  is  $n_b = 1 + \sum_{s=1}^{n_u-1} 2^s$ .

*Definition 2.* (Nearest complement affine subspaces). The shortest distance between complement affine subspaces is given by  $l_{\min} := \min\{l_1, l_2, \dots, l_{n_b}\}$ , where  $l_1, l_2, \dots, l_{n_b}$  are the distances between all possible pairs of complement affine subspaces that can be generated from  $\mathcal{S}$ .

In the 2-dimensional case ( $n_u = 2$ ), the number of distances to evaluate is  $n_b = 3$ , which corresponds to the 3 point-to-line distances. In the 3-dimensional case, there are  $n_b = 7$  distances to evaluate, which correspond to 4 point-to-plane distances, and 3 line-to-line distances.

In order to compute the distance between the complement affine subspaces (A.2) and (A.3), a vector  $\mathbf{n}$  that is normal to both subspaces is required:

$$\begin{bmatrix} \mathbf{u}_1 - \mathbf{u}_2 & \dots & \mathbf{u}_1 - \mathbf{u}_{n_u^A} & \mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2} & \dots \\ \mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1} \end{bmatrix}^T \mathbf{n} = \mathbf{0}, \quad \text{or,} \quad \mathbf{U} \mathbf{n} = \mathbf{0}. \quad (\text{A.4})$$

The matrix  $\mathbf{U} \in \mathbb{R}^{(n_u-1) \times n_u}$  is of rank  $(n_u - 1)$ . The vector  $\mathbf{n}$  can be obtained by singular-value decomposition of  $\mathbf{U}$ .

Given a point  $\mathbf{u}^a$  that belongs to the affine subspace (A.2), a point  $\mathbf{u}^b$  that belongs to the complement affine subspace (A.3), and a vector  $\mathbf{n}$  that is normal to both complement affine subspaces, the distance  $l_{AC}$  between the two complement affine subspaces is:

$$l_{AC} = \frac{|\mathbf{n}^T (\mathbf{u}^b - \mathbf{u}^a)|}{\|\mathbf{n}\|} \quad (\text{A.5})$$