Elsevier Editorial System(tm) for Journal of Process Control
Manuscript Draft

Manuscript Number:

Title: A Dual Modifier-Adaptation Approach for Real-Time Optimization

Article Type: Special Issue: ADCHEM 2009

Keywords: real-time optimization; modifier adaptation; dual modifier adaptation; gradient estimation; experimental gradient

Corresponding Author: Prof. Dominique Bonvin,

Corresponding Author's Institution: EPFL

First Author: Alejandro Gabriel Marchetti, Ph.D.

Order of Authors: Alejandro Gabriel Marchetti, Ph.D.; Benoît Chachuat, Professor; Dominique Bonvin, Professor

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 does not require the model parameters to be estimated on-line. Instead, it uses information regarding the constraints and selected gradients to improve the plant operation. 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 via numerical simulation for both an unconstrained and a constrained problem.

A Dual Modifier-Adaptation Approach for Real-Time Optimization

A. Marchetti [a,1], B. Chachuat [b], and D. Bonvin [a,2]

[a] *Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 9, CH-1015 Lausanne, Switzerland*

[b] *Chemical Engineering Department, McMaster University, 1280 Main Street West, Hamilton, ON L8P 1R2, 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 does not require the model parameters to be estimated on-line. Instead, it uses information regarding the constraints and selected gradients to improve the plant operation. 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 via numerical simulation for both an unconstrained and a constrained problem.

1 Introduction

Real-time optimization (RTO) is a technology that aims at improving steady-state operation of continuous plants [1]. The majority of RTO schemes available in the literature uses a model of the plant. Hence, reaching optimal performance in the presence of plant-model mismatch is a difficult task, which necessitates adaptation based on measured information. A three-way classification

[1] Present address: CIFASIS-CONICET, 27 de Febrero 210bis, S2000EZP Rosario, Argentina.
[2] Corresponding author. Email: dominique.bonvin@epfl.ch.

of RTO schemes has recently been proposed in [2]. One class includes the so-called modifier-adaptation approach [3], whereby appropriate terms are added to the optimization problem and updated so that the KKT conditions of the model match those of the plant. In this context, modifier adaptation requires experimental gradient information to be estimated on-line. This paper investigates the estimation of such gradients and their use in modifier-adaptation schemes.

A comparison of different approaches for on-line gradient estimation is given in [4]. Finite-difference techniques can be used to estimate the gradients experimentally. The most straightforward approach consists in perturbating 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 past operating points [5, 6]. The key issue therein is the ability to estimate the experimental gradients reliably while progressing with the optimization. Following the paradigm of dual control [7], this results in two conflicting objectives: the "primal objective" consists in improving the plant operation, while the "dual objective" aims at estimating accurate gradients. A way to accommodate these conflicting tasks is by adding a constraint in the optimization problem so as to ensure sufficient information in the measurements and guarantee gradient accuracy. For example, a constraint that prevents ill-conditioning in gradient computation has been proposed [5, 6]. The present paper goes further and investigates the two main sources of errors, namely the error introduced by the numerical approximation of derivatives 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, it has also been suggested to use the ill-conditioning measure not to constrain the optimization problem but rather to determine whether an additional input perturbation is needed [8]. Clearly, one such scheme could also be used in the context of the proposed dual-modifier approach.

The paper is organized as follows. Background material is presented in Section 2. The optimization problem is formulated and the necessary conditions of optimality are reviewed, then the modifier-adaptation scheme is presented. The method used to estimate the gradients from past operating points is introduced and analyzed in Section 3. Based on this analysis, a norm-based constraint is proposed, which is incorporated into the dual modifier-adaptation algorithm presented in Section 4 for unconstrained optimization problems, and in Section 5 for constrained optimization problems. The approach is illustrated in Section 6 for the unconstrained operation of the Williams-Otto reactor and for the optimization of a constrained numerical example. Finally,

Section 7 concludes the paper and presents directions for future work.

2 Preliminaries

2.1 Problem Formulation

The usual objective in RTO for continuous processes is to determine the operating point that optimizes some operating performance of the plant at steady state (e.g., minimization of operating cost or maximization of production rate), while satisfying a number of constraints (e.g., bounds on process variables or product specifications). The steady-state optimization problem for the plant can be formulated as follows:

$$\begin{aligned} \mathbf{u}_p^\star \in \arg\min_{\mathbf{u}} \quad & \Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \\ \text{s.t.} \quad & \mathbf{G}_p(\mathbf{u}) := \mathbf{g}(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq \mathbf{0}, \end{aligned} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the decision (or input) variables and $\mathbf{y}_p \in \mathbb{R}^{n_y}$ the measured (or output) variables; $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost function to be minimized; $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 1, \ldots, n_g$, is the set of inequality constraint functions, which includes input bounds. The notation $(\cdot)_p$ is used throughout for the variables associated with the plant.

This formulation assumes that $\phi(\mathbf{u}, \mathbf{y}_p)$ and $\mathbf{g}(\mathbf{u}, \mathbf{y}_p)$ are known functions of $\mathbf{u}$ and $\mathbf{y}_p$, i.e., they can be evaluated from the measurements. 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 $\mathbf{f}(\mathbf{u}, \mathbf{y}, \boldsymbol{\theta}) = \mathbf{0}$ 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 $\boldsymbol{\theta}$, i.e. $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$, the solution to the original problem (1) can be approached by solving the following NLP problem:

$$\begin{aligned} \mathbf{u}^\star \in \arg\min_{\mathbf{u}} \quad & \Phi(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{u}) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \leq \mathbf{0}. \end{aligned} \tag{2}$$

In the presence of plant-model mismatch, a model-based solution $\mathbf{u}^\star$ does not generally match the plant optimum $\mathbf{u}_p^\star$, so some adaptation is needed.

Assuming that the feasible set $U := \{\mathbf{u} : \mathbf{G}(\mathbf{u}) \leq \mathbf{0}\}$ is nonempty and compact for $\boldsymbol{\theta}$ given, and that $\Phi(\mathbf{u})$ is continuous on U, a minimizing solution $\mathbf{u}^\star$ of Problem (2) is guaranteed to exist (see, e.g., [9], Theorem 2.3.1). The set of active constraints at $\mathbf{u}^\star$ is denoted by $\mathcal{A} := \{i : G_i(\mathbf{u}^\star) = 0,\ i = 1, \ldots, n_g\}$.

2.2 Necessary Conditions of Optimality

Provided that a constraint qualification holds at the solution point $\mathbf{u}^\star$ and the functions Φ and $\mathbf{G}$ are differentiable at $\mathbf{u}^\star$, there exists unique Lagrange multipliers $\boldsymbol{\mu} \in \mathbb{R}^{n_g}$ such that the following first-order Karush-Kuhn-Tucker (KKT) conditions hold at $\mathbf{u}^\star$ [9]:

$$\begin{aligned} &\mathbf{G} \leq \mathbf{0}, \quad \boldsymbol{\mu}^\mathsf{T}\mathbf{G} = 0, \quad \boldsymbol{\mu} \geq \mathbf{0}, \\ &\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = \frac{\partial \Phi}{\partial \mathbf{u}} + \boldsymbol{\mu}^\mathsf{T}\frac{\partial \mathbf{G}}{\partial \mathbf{u}} = \mathbf{0}, \end{aligned} \tag{3}$$

with $\mathcal{L} := \Phi + \boldsymbol{\mu}^\mathsf{T}\mathbf{G}$ being the Lagrangian function.

The problem of approaching the optimization problem (1) for the plant by solving the model-based optimization (2) is analogous to the inside-out problem discussed by Biegler et al. [10]. In that work, the observation was made that a necessary condition for a model to be adequate for optimization is that it recognizes the plant optimum as a KKT point. In other words, a KKT point for the model ought to be a KKT point for the plant. The KKT conditions involve the values and gradients of the constraints as well as the gradient of the cost function. These KKT elements are denoted collectively by the vector $\boldsymbol{\mathcal{C}} = \left(G_1, \ldots, G_{n_g}, \frac{\partial G_1}{\partial \mathbf{u}}, \ldots, \frac{\partial G_{n_g}}{\partial \mathbf{u}}, \frac{\partial \Phi}{\partial \mathbf{u}}\right)^\mathsf{T} \in \mathbb{R}^{n_K}$, with $n_K = n_g + n_u(n_g + 1)$. Since the Lagrange multipliers $\boldsymbol{\mu}$ corresponding to inactive constraints are zero, KKT matching only requires to match the values and gradients of the active constraints. However, because neither the plant optimum nor its corresponding active set are known a priori, a sufficient condition for a model-based RTO scheme to reach the plant optimum upon convergence is that *all* the KKT elements of the model $\boldsymbol{\mathcal{C}}$ match those of the plant $\boldsymbol{\mathcal{C}}_p$ at every RTO iteration. The modifier-adaptation approach, which is described next, has been conceived for meeting this condition in spite of plant-model mismatch.

2.3 Modifier-Adaptation Approach

The idea behind modifier adaptation is to use measurements for correcting the predicted cost and constraints between successive RTO iterations in such a way that a KKT point for the model eventually coincides with the plant optimum [3]. At the kth step of modifier adaptation, the next input $\mathbf{u}_{k+1}$ is obtained as:

$$\begin{aligned} \mathbf{u}_{k+1} \in \arg\min_{\mathbf{u}} \; & \Phi_m(\mathbf{u}) := \Phi(\mathbf{u}) + \boldsymbol{\lambda}_k^{\Phi\mathsf{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}) := \mathbf{G}(\mathbf{u}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^{\mathbf{G}\mathsf{T}}(\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \end{aligned} \tag{4}$$

where $\boldsymbol{\varepsilon}_k \in \mathbb{R}^{n_g}$ are the constraint-value modifiers; $\boldsymbol{\lambda}_k^{\mathbf{G}} \in \mathbb{R}^{n_u \times n_g}$ are the constraint-gradient modifiers; and $\boldsymbol{\lambda}_k^{\Phi} \in \mathbb{R}^{n_u}$ are the cost-gradient modifiers. These modifiers represent the difference between the plant and predicted values of the KKT elements. They are adapted repeatedly by using (estimates of) the constraint values and the cost and constraint gradients of the plant at the current operating point $\mathbf{u}_k$:

$$\boldsymbol{\varepsilon}_k = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k), \tag{5}$$

$$\boldsymbol{\lambda}_k^{\mathbf{G}^\mathsf{T}} = \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_k), \tag{6}$$

$$\boldsymbol{\lambda}_k^{\Phi^\mathsf{T}} = \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k). \tag{7}$$

The modifiers can be denoted collectively by the n_K-dimensional vector $\boldsymbol{\Lambda}_k = \left(\varepsilon_{1,k}, \ldots, \varepsilon_{n_g,k}, \boldsymbol{\lambda}_k^{G_1\mathsf{T}}, \ldots, \boldsymbol{\lambda}_k^{G_{n_g}\mathsf{T}}, \boldsymbol{\lambda}^{\Phi\mathsf{T}}_k\right)^\mathsf{T} \in \mathbb{R}^{n_K}$. This way, (5)-(7) can be rewritten as $\boldsymbol{\Lambda}_k = \boldsymbol{\mathcal{C}}_p(\mathbf{u}_k) - \boldsymbol{\mathcal{C}}(\mathbf{u}_k)$. However, such a simple strategy may lead to excessive correction, thereby compromising the convergence of the algorithm, and it may also make modifier adaptation very sensitive to measurement noise. A better strategy consists in filtering the modifiers, using for example a first-order exponential filter:

$$\boldsymbol{\Lambda}_k = (\mathbf{I} - \mathsf{K})\boldsymbol{\Lambda}_{k-1} + \mathsf{K}\left[\boldsymbol{\mathcal{C}}_p(\mathbf{u}_k) - \boldsymbol{\mathcal{C}}(\mathbf{u}_k)\right], \tag{8}$$

where $\mathsf{K} \in \mathbb{R}^{n_K \times n_K}$ is a gain matrix.

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 optimization problem (4) satisfies the same first-order optimality conditions as the problem (1) [3]. The downside of modifier adaptation lies in the need to estimate the experimental gradients $\frac{\partial \Phi_p}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}$.

3 Experimental Gradient Computed from Past Operating Points

Gradients have numerical values that depend on the order of magnitude of the decision variables $\mathbf{u}$. It is assumed throughout this work that all the decision variables $\mathbf{u}$ are of the same order of magnitude, which usually necessitates some scaling. For example, if the decision variable u_i remains within the interval $[u_{i,\mathrm{a}}, u_{i,\mathrm{b}}]$, it can be scaled between 0 and 1 as $u_i^{\mathrm{scaled}} = (u_i - u_{i,\mathrm{a}})/(u_{i,\mathrm{b}} - u_{i,\mathrm{a}})$. For notational simplicity, the superscript indicating a scaled variable will be omitted thereafter.

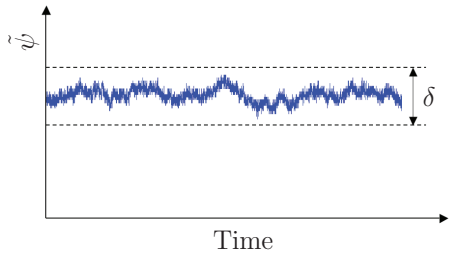

Fig. 1. Measured quantity at steady state with indication of the noise level δ.

3.1 Gradient Computation

A straightforward approach for estimating the experimental gradients consists in perturbing each input individually around the current operating point and estimating the corresponding gradient element. In the forward finite differencing (FFD) approach, an estimator of the partial derivative $\frac{\partial y_{p_i}}{\partial u_j}(\mathbf{u}_k)$, $i = 1, \ldots, n_y$, $j = 1, \ldots, n_u$, at the kth RTO iteration is obtained as

$$\hat{\gamma}_j(h) = \left[y_{p_i}(\mathbf{u}_k + h\mathbf{e}_j) - y_{p_i}(\mathbf{u}_k) \right] / h, \qquad h > 0, \tag{9}$$

where h is the step size and $\mathbf{e}_j$ the jth unit vector.

This approach requires n_u perturbations to be carried out at each RTO iteration. In addition, it is necessary to wait for a new steady-state after each perturbation. As an alternative approach, it is possible to estimate the gradients using several past operating points generated by the previous RTO iterations, as proposed for example in dual ISOPE algorithms [5, 6]. This approach reduces the total number of input changes that are required. Furthermore, unlike the FFD approach, the operating points used to estimate the gradients do not have a fixed spatial arrangement.

The analysis in this section is carried out for a general noisy function of the form

$$\tilde{\psi}(\mathbf{u}) = \psi(\mathbf{u}) + v, \tag{10}$$

where v denotes the measurement noise. $\tilde{\psi}(\mathbf{u})$ might represent the measured plant cost, a measured constraint value or simply an output variable. It is assumed that the noisy function $\tilde{\psi}(\mathbf{u})$ remains within an interval δ at steady-state operation, as illustrated in Figure 1. The interval δ can be selected by considering some confidence level for a given statistical description of v.

Consider the kth iteration and the n_u most recent operating points, $\mathbf{u}_{k-j}$, $j = 0, \ldots, n_u - 1$. The objective is to evaluate the gradient $\frac{\partial \psi}{\partial \mathbf{u}}(\mathbf{u})$ as a function of the location of the next operating point that will generically be labeled $\mathbf{u}$. Let us consider a first-order approximation of $\psi(\mathbf{u}_{k-j})$ in the neighborhood of the

new operating point $\mathbf{u}$:

$$\psi(\mathbf{u}_{k-j}) = \psi(\mathbf{u}) + \frac{\partial \psi}{\partial \mathbf{u}}(\mathbf{u})[\mathbf{u}_{k-j} - \mathbf{u}] + O\left(\|\mathbf{u}_{k-j} - \mathbf{u}\|^2\right), \tag{11}$$

for each $j = 0, \ldots, n_u - 1$. Written for the noisy functions $\tilde{\psi}(\mathbf{u}_{k-j})$ and $\tilde{\psi}(\mathbf{u})$, and neglecting the higher-order terms, (11) becomes:

$$\tilde{\psi}(\mathbf{u}_{k-j}) = \tilde{\psi}(\mathbf{u}) + \hat{\boldsymbol{\gamma}}(\mathbf{u})^{\mathsf{T}}[\mathbf{u}_{k-j} - \mathbf{u}], \qquad j = 0, \ldots, n_u - 1, \tag{12}$$

where $\hat{\boldsymbol{\gamma}}(\mathbf{u})$ is an estimate of the gradient $\boldsymbol{\gamma}(\mathbf{u})^{\mathsf{T}} := \frac{\partial \psi}{\partial \mathbf{u}}(\mathbf{u})$. This estimate can be computed from the n_u most recent operating points $\mathbf{u}_k, \ldots, \mathbf{u}_{k-n_u+1}$ and the corresponding noisy values $\tilde{\psi}(\mathbf{u}_k), \ldots, \tilde{\psi}(\mathbf{u}_{k-n_u+1})$ by writing (12) in the following matrix form [6]:

$$\hat{\boldsymbol{\gamma}}(\mathbf{u})^{\mathsf{T}} = \mathcal{Y}(\mathbf{u})\, \mathcal{U}^{-1}(\mathbf{u}), \tag{13}$$

with

$$\mathcal{U}(\mathbf{u}) := [\, \mathbf{u} - \mathbf{u}_k \ \ldots \ \mathbf{u} - \mathbf{u}_{k-n_u+1} \,] \in \mathbb{R}^{n_u \times n_u} \tag{14}$$

$$\mathcal{Y}(\mathbf{u}) := [\, \tilde{\psi}(\mathbf{u}) - \tilde{\psi}(\mathbf{u}_k) \ \ldots \ \tilde{\psi}(\mathbf{u}) - \tilde{\psi}(\mathbf{u}_{k-n_u+1}) \,] \in \mathbb{R}^{1 \times n_u}. \tag{15}$$

It is assumed in this analysis that the $n_u + 1$ operating points in $\mathcal{U}(\mathbf{u})$ are such that $\mathcal{U}(\mathbf{u})$ is invertible. The gradient estimation error is defined as

$$\boldsymbol{\epsilon}(\mathbf{u})^{\mathsf{T}} := \hat{\boldsymbol{\gamma}}(\mathbf{u})^{\mathsf{T}} - \frac{\partial \psi}{\partial \mathbf{u}}(\mathbf{u}),$$

which, from (13) and using $\tilde{\psi}(\mathbf{u}_{k-j}) = \psi(\mathbf{u}_{k-j}) + v_{k-j}$ and $\tilde{\psi}(\mathbf{u}) = \psi(\mathbf{u}) + v$, can be split as

$$\boldsymbol{\epsilon}(\mathbf{u}) = \boldsymbol{\epsilon}^t(\mathbf{u}) + \boldsymbol{\epsilon}^n(\mathbf{u}), \tag{16}$$

where $\boldsymbol{\epsilon}^t$ and $\boldsymbol{\epsilon}^n$ represent the errors due to truncation and measurement noise, respectively,

$$\boldsymbol{\epsilon}^t(\mathbf{u})^{\mathsf{T}} = \ [\, \psi(\mathbf{u}) - \psi(\mathbf{u}_k) \quad \ldots \quad \psi(\mathbf{u}) - \psi(\mathbf{u}_{k-n_u+1}) \,]\, \mathcal{U}^{-1}(\mathbf{u}) - \frac{\partial \psi}{\partial \mathbf{u}}(\mathbf{u}) \tag{17}$$

$$\boldsymbol{\epsilon}^n(\mathbf{u})^{\mathsf{T}} = \ [\, v - v_k \quad \ldots \quad v - v_{k-n_u+1} \,]\, \mathcal{U}^{-1}(\mathbf{u}). \tag{18}$$

Next, we investigate these two components of the gradient error.

3.2 Gradient Error due to Truncation

An upper bound on the norm of the truncation error is given in the next proposition.

Proposition 1 *Let $\psi(\mathbf{u})$ be twice continuously differentiable with respect to $\mathbf{u}$. Then, given the n_u most recent operating points $\mathbf{u}_k, \ldots, \mathbf{u}_{k-n_u+1}$, one has*

$$\|\boldsymbol{\epsilon}^t(\mathbf{u})\| \leq \mathcal{E}^t(\mathbf{u}), \tag{19}$$

with

$$\begin{aligned}\mathcal{E}^t(\mathbf{u}) := \frac{\sigma_{max}}{2}\Big\| \Big[\; (\mathbf{u}-\mathbf{u}_k)^\mathsf{T}(\mathbf{u}-\mathbf{u}_k) \;\; \ldots \\ \ldots \;\; (\mathbf{u}-\mathbf{u}_{k-n_u+1})^\mathsf{T}(\mathbf{u}-\mathbf{u}_{k-n_u+1}) \; \Big] \; \mathcal{U}^{-1}(\mathbf{u})\Big\|,\end{aligned} \tag{20}$$

where σ_{max} is the largest absolute eigenvalue of the Hessian of $\psi(\cdot)$.

Sketch of the proof. The proof proceeds by Taylor series expansion of $\psi(\mathbf{u}_{k-j})$ at $\mathbf{u}$ and upper bounding of the norm of the Hessian of ψ. See [11] for details. □

Note that σ_{max} represents an upper bound on the curvature of $\psi(\cdot)$.

Remark 1 *If the points are taken as $\mathbf{u}_{k-j} = \mathbf{u} + h\mathbf{e}_{j+1}$, $j = 0, \ldots, n_u - 1$, there results the FFD arrangement with $h > 0$ being the step size and $\mathbf{e}_{j+1}$ the $(j+1)$st unit vector. In this particular case, it can be shown that* (19) *reduces to*

$$\|\boldsymbol{\epsilon}^t\| \leq \frac{\sigma_{max}}{2}\sqrt{n_u}\; h, \tag{21}$$

which is independent of $\mathbf{u}$. Notice that (21) *is the same expression as that reported in [12] for the FFD approach.*

Remark 2 *For a given value c, the contour level $\mathcal{E}^t(\mathbf{u}) = c$ corresponds to the envelopes of two intersecting hyperspheres of radius $r = \frac{c}{\sigma_{max}}$, the centers of which are symmetrically located on each side of the hyperplane generated by the n_u past operating points [11]. The center point corresponding to a given operating point $\mathbf{u}$ is given by*

$$\mathbf{u}_c^\mathsf{T}(\mathbf{u}) = \frac{1}{2}\Big[\; \mathbf{u}^\mathsf{T}\mathbf{u} - \mathbf{u}_k^\mathsf{T}\mathbf{u}_k \;\; \cdots \;\; \mathbf{u}^\mathsf{T}\mathbf{u} - \mathbf{u}_{k-n_u+1}^\mathsf{T}\mathbf{u}_{k-n_u+1} \; \Big] \; \mathcal{U}^{-1}(\mathbf{u}), \tag{22}$$

and it can be shown that

$$\mathcal{E}^t(\mathbf{u}) = \sigma_{max}\Big\|\mathbf{u} - \mathbf{u}_c(\mathbf{u})\Big\| = \sigma_{max}\, r(\mathbf{u}), \tag{23}$$

where $r(\mathbf{u})$ is the radius of the hypersphere as a function of $\mathbf{u}$. The situation in the two-input case is depicted in Figure 2. The most recent operating points are $\mathbf{u}_k$ and $\mathbf{u}_{k-1}$. The new operating point $\mathbf{u}$ generates the center point $\mathbf{u}_{c1}(\mathbf{u})$ of radius $r(\mathbf{u})$. The symmetric center point $\mathbf{u}_{c2}(\mathbf{u})$ is also represented. The corresponding upper bound on the norm of the gradient error due to truncation is $\mathcal{E}^t(\mathbf{u}) = \sigma_{max}\, r(\mathbf{u})$.

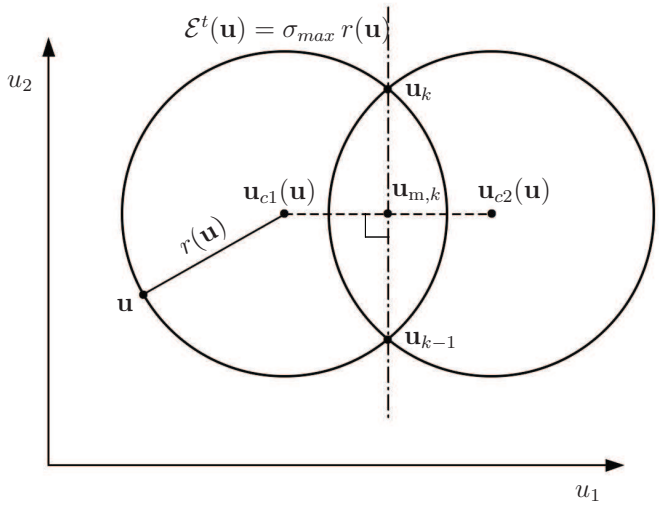

Fig. 2. Representation of the geometrical properties of $\mathcal{E}^t(\mathbf{u})$ in the two-input case.

3.3 Gradient Error due to Measurement Noise

For relating the error norm $\|\boldsymbol{\epsilon}^n(\mathbf{u})\|$ 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 $\|\boldsymbol{\epsilon}^n(\mathbf{u})\|$, noted $\mathcal{E}^n(\mathbf{u})$, is computed in the next proposition.

Proposition 2 *Given the n_u most recent operating points $\mathbf{u}_k, \ldots, \mathbf{u}_{k-n_u+1}$ and the interval δ for the noisy function $\tilde{\psi}(\cdot)$, one has:*

$$\|\boldsymbol{\epsilon}^n(\mathbf{u})\| \leq \mathcal{E}^n(\mathbf{u}) := \frac{\delta}{l_{\min}(\mathbf{u})}, \tag{24}$$

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

Sketch of the 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 $\boldsymbol{\epsilon}^n(\mathbf{u})$ is normal to both affine subspaces, which results in the largest possible error norm given by (24). See [11] for details. □

Remark 3 *In order to evaluate $l_{min}(\mathbf{u})$, it is necessary to evaluate all the distances between complement affine subspaces. Let us consider the distance between the new operating point $\mathbf{u}$ and the hyperplane generated by the most recent operating points. In this case, we have $\mathcal{S}^A := \{\mathbf{u}_k, \mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k-n_u+1}\}$*

and $\mathcal{S}^C := \{\mathbf{u}\}$. *The matrix* $\mathsf{U} \in \mathbb{R}^{(n_u-1)\times n_u}$ *defined in* (A.4) *is given by:*

$$\mathsf{U}_k := [\ \mathbf{u}_k - \mathbf{u}_{k-1}\ \ \mathbf{u}_k - \mathbf{u}_{k-2}\ \ \ldots\ \ \mathbf{u}_k - \mathbf{u}_{k-n_u+1}\ ]^{\mathsf{T}}.$$

Denoting by $\mathbf{n}_k$ *the vector normal to the hyperplane generated by the most recent operating points, we have that* $\mathsf{U}_k\mathbf{n}_k = \mathbf{0}$ *and, from* (A.1) *and* (A.5)*, the hyperplane is given by* $\mathbf{n}_k^{\mathsf{T}}\mathbf{u} = b_k$*, with* $b_k = \frac{\mathbf{n}_k\mathbf{u}_k}{\|\mathbf{n}_k\|}$*. Notice that, since* U_k *does not depend on* $\mathbf{u}$*, the direction of* $\mathbf{n}_k$ *is independent of* $\mathbf{u}$*. This is not the case for the normal directions between all the other complement affine subspaces, which depend on the position of* $\mathbf{u}$*.*

3.4 Upper Bound on Gradient Error

The estimated gradient (13) requires the matrix $\mathcal{U}(\mathbf{u})$ to be non-singular. In dual ISOPE algorithms [5, 6], good conditioning is achieved by introducing a lower bound on the inverse condition number of $\mathcal{U}(\mathbf{u})$:

$$\kappa^{-1}(\mathbf{u}) := \frac{\underline{\sigma}(\mathcal{U}(\mathbf{u}))}{\overline{\sigma}(\mathcal{U}(\mathbf{u}))} \geq \varphi, \tag{25}$$

where $\kappa(\mathbf{u})$ is the condition number of $\mathcal{U}(\mathbf{u})$, with $\overline{\sigma}$ and $\underline{\sigma}$ denoting the maximum and minimum singular values, respectively. 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.

In this section, a consistent, although possibly conservative, upper bound on the gradient error norm is introduced. Let $\mathcal{E}^U$ denote the desired upper bound on the gradient error norm:

$$\|\boldsymbol{\epsilon}(\mathbf{u})\| \leq \mathcal{E}^U. \tag{26}$$

The following theorem provides a sufficient condition for the location of $\mathbf{u}$ so as to satisfy (26) given the n_u past operating points $\mathbf{u}_k, \ldots, \mathbf{u}_{k-n_u+1}$.

Theorem 1 *For given values of* δ*,* d_2 *and* $\mathcal{E}^U$*, the gradient error norm* $\|\boldsymbol{\epsilon}(\mathbf{u})\|$ *does not exceed the desired upper bound* $\mathcal{E}^U$ *if* $\mathbf{u}$ *is chosen so that*

$$\mathcal{E}(\mathbf{u}) := \mathcal{E}^t(\mathbf{u}) + \mathcal{E}^n(\mathbf{u}) \leq \mathcal{E}^U, \tag{27}$$

with $\mathcal{E}^t(\mathbf{u})$ *and* $\mathcal{E}^n(\mathbf{u})$ *given by* (20) *and* (24)*, respectively.*

Proof. It follows from (16), (19) and (24) that:

$$\|\boldsymbol{\epsilon}(\mathbf{u})\| \leq \|\boldsymbol{\epsilon}^t(\mathbf{u})\| + \|\boldsymbol{\epsilon}^n(\mathbf{u})\| \leq \mathcal{E}(\mathbf{u}). \tag{28}$$

The proof follows from inequalities (28) and (27). □

Note that, for given values of δ 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 (27) to be feasible.

The following remark considers the feasible regions generated by $\mathcal{E}^n(\mathbf{u})$.

Remark 4 *If $\psi(\mathbf{u})$ is linear, $\sigma_{max} = 0$ (there is no truncation error), and (27) reduces to $\delta/l_{\min}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}}$. Equivalently, the following lower bound $l^{\mathrm{L}}_{\min}$ can be defined for $l_{\min}(\mathbf{u})$:*

$$l^{\mathrm{L}}_{\min} := \frac{\delta}{\mathcal{E}^{\mathrm{U}}} \leq l_{\min}(\mathbf{u}). \tag{29}$$

The constraint (29) can be also seen as the combination of the n_b constraints $l^{\mathrm{L}}_{\min} \leq l_i(\mathbf{u})$, for $i = 1, \ldots, n_b$, where n_b is the number of complement affine subspaces. Let us consider the distance between $\mathbf{u}$ and the hyperplane $\mathbf{n}_k^{\mathsf{T}}\mathbf{u} = b_k$. If $\boldsymbol{\epsilon}^{n\mathsf{T}}(\mathbf{u}-\mathbf{u}_k) > 0$, the constraint (29) corresponding to this pair of complement affine subspaces gives

$$l^{\mathrm{L}}_{\min} \leq \frac{\boldsymbol{\epsilon}^{n\mathsf{T}}}{\|\boldsymbol{\epsilon}^n\|}(\mathbf{u} - \mathbf{u}_k) = \frac{\mathbf{n}_k^{\mathsf{T}}}{\|\mathbf{n}_k\|}(\mathbf{u} - \mathbf{u}_k),$$

which can be written as

$$\mathbf{n}_k^{\mathsf{T}}\mathbf{u} \geq b_k + l^{\mathrm{L}}_{\min}\|\mathbf{n}_k\|. \tag{30}$$

On the other hand, if $\boldsymbol{\epsilon}^{n\mathsf{T}}(\mathbf{u} - \mathbf{u}_k) < 0$, we obtain

$$\mathbf{n}_k^{\mathsf{T}}\mathbf{u} \leq b_k - l^{\mathrm{L}}_{\min}\|\mathbf{n}_k\|. \tag{31}$$

Hence, this point-to-hyperplane constraint generates two feasible regions, one on each side of the hyperplane $\mathbf{n}_k^{\mathsf{T}}\mathbf{u} = b_k$. For all other complement affine subspaces, the direction of a vector that is normal to both affine subspaces will vary with the position of $\mathbf{u}$.

Example 1 *For the purpose of illustration, consider the two-input case ($n_u = 2$) with $\delta = 0.2$ and $\mathcal{E}^{\mathrm{U}} = 0.5$. Figure 3 uses the most recent operating points $\mathbf{u}_k = [0 \quad -0.5]^{\mathsf{T}}$ and $\mathbf{u}_{k-1} = [0 \quad 0.5]^{\mathsf{T}}$. The constraints (29), which can be evaluated in terms of the position of the new operating point $\mathbf{u} = [u_1 \quad u_2]^{\mathsf{T}}$, consist of three point-to-line distances. For the case of the distance between $\mathbf{u}$ and the line generated by $\mathbf{u}_k$ and $\mathbf{u}_{k-1}$, denoted l_1 in this example, the feasible regions generated by (30) and (31) are given in Figure 3a. The feasible regions corresponding to the two remaining point-to-line distances are shown in Figures 3b and 3c. The combination of these constraints is given in Figure 3d. It is seen that (29) generates two convex feasible regions, one on each side of the hyperplane $\mathbf{n}_k^{\mathsf{T}}\mathbf{u} = b_k$.*

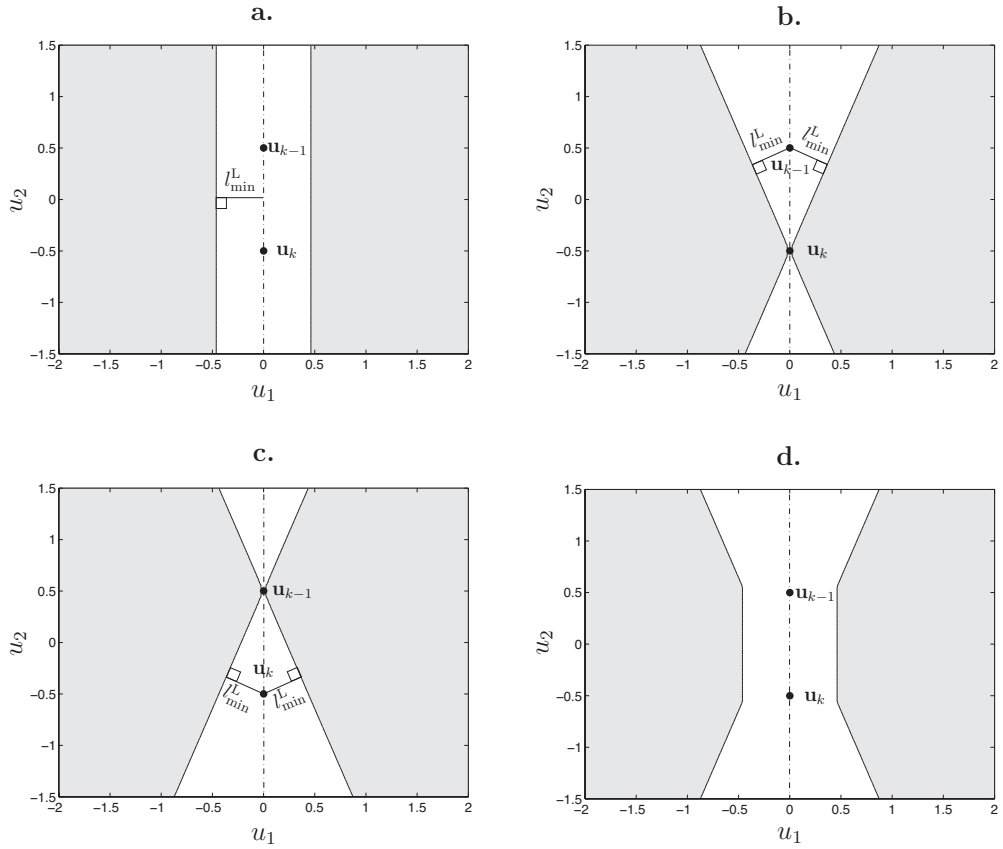

Fig. 3. Evaluation of the feasible regions given by constraint (29) in Example 1, as combination of the constraints $l^{\mathrm{L}}_{\mathrm{min}} \leq l_i(\mathbf{u})$. Shaded area: feasible regions; **Plot a**: feasible regions given by the lower bound on the distance between $\mathbf{u} = [u_1 \;\; u_2]^{\mathsf{T}}$ and the line generated by $\mathbf{u}_k$ and $\mathbf{u}_{k-1}$ ($l^{\mathrm{L}}_{\mathrm{min}} \leq l_1(\mathbf{u})$); **Plot b**: feasible regions given by the lower bound on the distance between $\mathbf{u}_{k-1}$ and the line generated by $\mathbf{u}$ and $\mathbf{u}_k$ ($l^{\mathrm{L}}_{\mathrm{min}} \leq l_2(\mathbf{u})$); **Plot c**: feasible regions given by the lower bound on the distance between $\mathbf{u}_k$ and the line generated by $\mathbf{u}$ and $\mathbf{u}_{k-1}$ ($l^{\mathrm{L}}_{\mathrm{min}} \leq l_3(\mathbf{u})$); **Plot d**: feasible regions given by constraint (29) ($l^{\mathrm{L}}_{\mathrm{min}} \leq l_{\mathrm{min}}(\mathbf{u})$).

Example 2 *Consider the two-input case ($n_u = 2$) with $\delta = 0.2$ and $\sigma_{max} = 2$. Figures 4a, 4c and 4e on the left side use the most recent operating points $\mathbf{u}_k = [0 \;\; -0.5]^{\mathsf{T}}$ and $\mathbf{u}_{k-1} = [0 \;\; 0.5]^{\mathsf{T}}$, while figures 4b, 4d and 4f on the right side, use the operating points $\mathbf{u}_k = [0 \;\; -0.1]^{\mathsf{T}}$ and $\mathbf{u}_{k-1} = [0 \;\; 0.1]^{\mathsf{T}}$ that are closer to each other. The upper bounds $\mathcal{E}^t(\mathbf{u})$ and $\mathcal{E}^n(\mathbf{u})$ are evaluated in terms of the position of the new operating point $\mathbf{u} = [u_1 \;\; u_2]^{\mathsf{T}}$. Figures 4a and 4b show the contours of $\mathcal{E}^t(\mathbf{u})$, whereas the contours of $\mathcal{E}^n(\mathbf{u})$ are shown in Figures 4c and 4d, and the contours of $\mathcal{E}(\mathbf{u})$ are shown in Figures 4e and 4f. It is seen that both $\mathcal{E}^t(\mathbf{u})$ and $\mathcal{E}^n(\mathbf{u})$ increase as $\mathcal{U}(\mathbf{u})$ becomes more ill-conditioned ($\mathbf{u}$ aligned with $\mathbf{u}_k$ and $\mathbf{u}_{k-1}$). Moreover, the two regions generated by the constraint (27) may become nonconvex depending on the distance between the past operating points.*

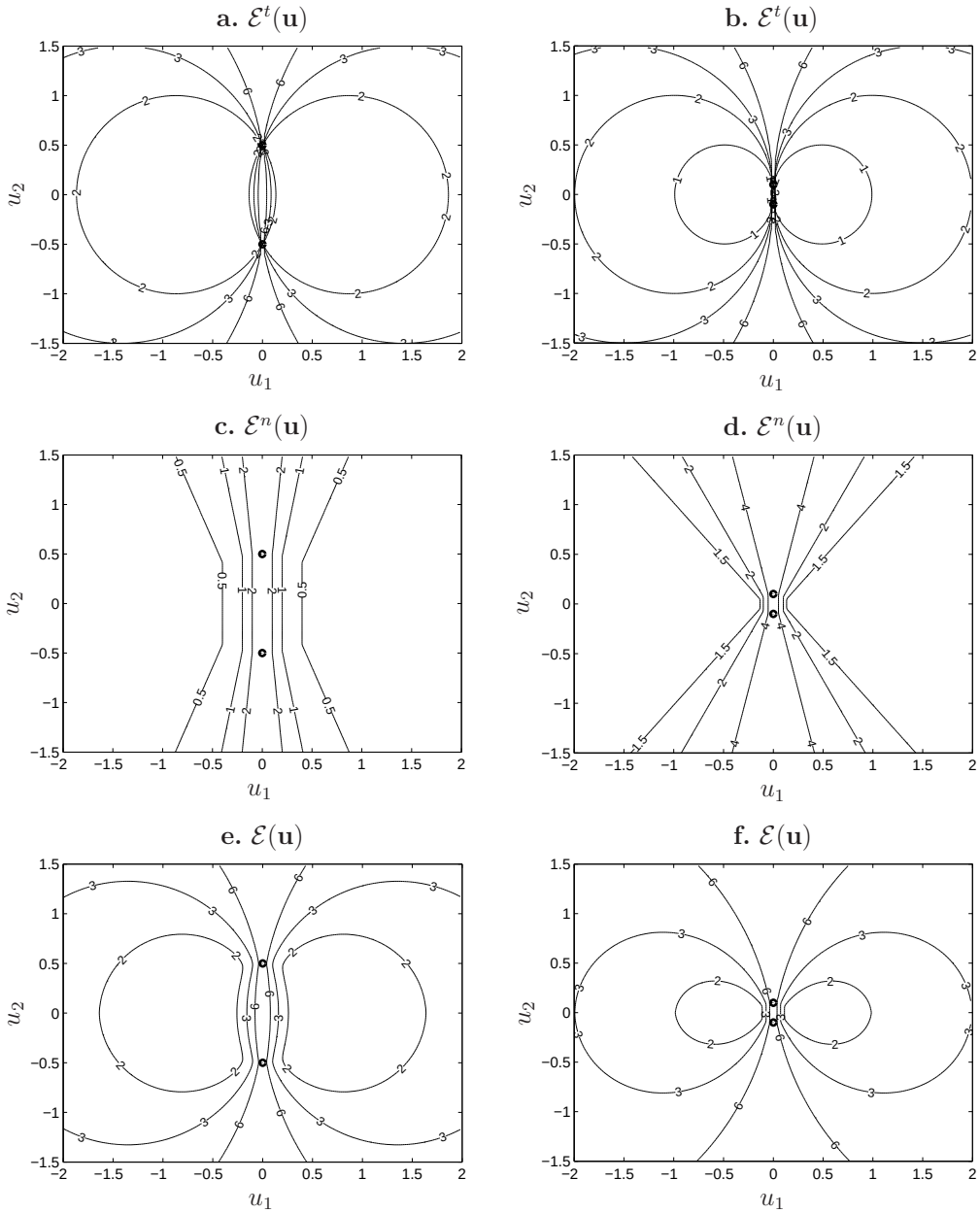

Fig. 4. Contour maps of an upper bound on the gradient error due to truncation (**a**,**b**), measurement noise (**c**,**d**), and both (**e**,**f**) for two cases of most recent points (more distant on the left, and closer on the right).

Convexification of the Gradient Error Constraint. If the optimization problem (4) is convex, addition of the nonconvex constraint (27) would open the possibility of multiple local solutions. Hence, we introduce a tight relaxation that makes this constraint convex.

Since the worst-case measurement error $\delta/l_{\min}(\mathbf{u})$ is convex on each side of the hyperplane $\mathbf{n}_k^\mathsf{T}\mathbf{u} = b_k$ (see Example 1), it is clear that the non-convexity of the regions generated by the constraint (27) is due to the part of the hyperspheres that crosses the hyperplane $\mathbf{n}_k^\mathsf{T}\mathbf{u} = b_k$. The distance (positive or negative)

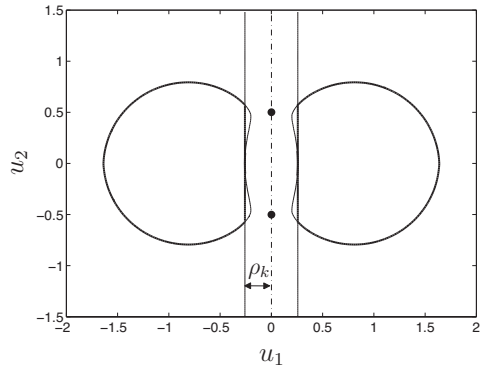

Fig. 5. Convex regions (in bold) corresponding to the constraint $\mathcal{E}(\mathbf{u}) \leq 2$.

from the center point $\mathbf{u}_c(\mathbf{u})$ to the hyperplane $\mathbf{n}_k^{\mathsf{T}}\mathbf{u} = b_k$ is given by:

$$l_C(\mathbf{u}) = \frac{b_k - \mathbf{n}_k^{\mathsf{T}}\mathbf{u}_c(\mathbf{u})}{\|\mathbf{n}_k\|}. \tag{32}$$

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

$$\mathbf{u}_{\mathrm{m},k} = \mathbf{u}_c(\mathbf{u}) + \frac{l_C(\mathbf{u})}{\|\mathbf{n}_k\|}\mathbf{n}_k. \tag{33}$$

It can be verified that $\mathbf{u}_{\mathrm{m},k}$ is independent of $\mathbf{u}$ (see the location of $\mathbf{u}_{\mathrm{m},k}$ in Figure 2 in the two-input case). For a given upper bound $\mathcal{E}^{\mathrm{U}}$, it is possible to define convex feasibility regions by adding constraints that express the distance between the new point and the hyperplane. This way, the non-convex part of the regions generated by (27) is eliminated, as illustrated in Figure 5. The minimal point-to-hyperplane distance ρ_k can be determined numerically by finding the smallest absolute value solution to the following equation

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

4 Dual Modifier Adaptation for Unconstrained Optimization

In unconstrained optimization problems, only the cost gradient needs to be estimated at each iteration. The function $\psi(\mathbf{u})$ introduced in (10) is therefore the plant cost itself, i.e. $\psi(\mathbf{u}) = \Phi_p(\mathbf{u})$. In the presence of noisy measurements, an estimate $\tilde{\psi}(\mathbf{u})$ of the cost is obtained as:

$$\tilde{\psi}(\mathbf{u}) = \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) = \Phi_p(\mathbf{u}) + v, \tag{34}$$

where $\boldsymbol{\nu}$ is the output measurement noise vector, and v represents the resulting noise in the cost function. Notice that v generally has a nonzero mean when the function $\phi(\mathbf{u}, \mathbf{y})$ is nonlinear in $\mathbf{y}$, even in the case that $\boldsymbol{\nu}$ itself has zero mean.

4.1 Loss in Cost due to Cost Gradient Error

The ability to bound the gradient error is not important per se but as a means to limit the loss in performance. In this section, a local analysis of the effect of the gradient error in terms of performance is carried out. This analysis assumes that the optimum $\mathbf{u}_{k+1}$ of the modified optimization problem is located in the neighborhood of the plant optimum $\mathbf{u}_p^\star$, which is assumed to be a strict local minimum. The loss in cost at the operating point $\mathbf{u}_{k+1}$ is given by:

$$\Delta\Phi_p(\mathbf{u}_{k+1}) := \Phi_p(\mathbf{u}_{k+1}) - \Phi_p(\mathbf{u}_p^\star). \tag{35}$$

With $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_p^\star) = \mathbf{0}$, the Taylor expansion of $\Delta\Phi_p(\mathbf{u}_{k+1})$ around $\mathbf{u}_p^\star$ gives:

$$\Delta\Phi_p(\mathbf{u}_{k+1}) = \frac{1}{2}(\mathbf{u}_{k+1} - \mathbf{u}_p^\star)^\mathsf{T} \frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2}(\mathbf{u}_p^\star)(\mathbf{u}_{k+1} - \mathbf{u}_p^\star) + O\left(\|\mathbf{u}_{k+1} - \mathbf{u}_p^\star\|^3\right) \tag{36}$$

The error between the predicted and plant cost gradients is given by:

$$\boldsymbol{\epsilon}_m^\Phi(\mathbf{u}_{k+1})^\mathsf{T} := \frac{\partial \Phi_m}{\partial \mathbf{u}}(\mathbf{u}_{k+1}) - \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_{k+1}), \tag{37}$$

In this equation, $\frac{\partial \Phi_m}{\partial \mathbf{u}}(\mathbf{u}_{k+1}) = \mathbf{0}$ because $\mathbf{u}_{k+1}$ is an optimum of the model-based optimization problem. On the other hand, $\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_{k+1})$ can be written using Taylor expansion around $\mathbf{u}_p^\star$ to give:

$$\boldsymbol{\epsilon}_m^\Phi(\mathbf{u}_{k+1})^\mathsf{T} = -(\mathbf{u}_{k+1} - \mathbf{u}_p^\star)^\mathsf{T} \frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2}(\mathbf{u}_p^\star) + O\left(\|\mathbf{u}_{k+1} - \mathbf{u}_p^\star\|^2\right). \tag{38}$$

Combining (36) and (38) gives:

$$\Delta\Phi_p(\mathbf{u}_{k+1}) = \frac{1}{2}\boldsymbol{\epsilon}_m^\Phi(\mathbf{u}_{k+1})^\mathsf{T}\left[\frac{\partial^2 \Phi_p}{\partial \mathbf{u}^2}(\mathbf{u}_p^\star)\right]^{-1}\boldsymbol{\epsilon}_m^\Phi(\mathbf{u}_{k+1}) + O\left(\|\mathbf{u}_{k+1} - \mathbf{u}_p^\star\|^3\right), \tag{39}$$

which establishes the local relation between the predicted cost gradient error $\boldsymbol{\epsilon}_m^\Phi(\mathbf{u}_{k+1})$ and the loss in cost $\Delta\Phi_p(\mathbf{u}_{k+1})$ in unconstrained optimization problems.

4.2 Dual Modifier-Adaptation Algorithm

The dual modifier-adaptation scheme proposed in this section uses the upper bound on the gradient error defined in Section 3 as a constraint in the optimization problem. On each side of the hyperplane generated by the most recent operating points, $\mathbf{n}_k^\mathsf{T}\mathbf{u} = b_k$, a modified optimization problem is solved. The optimization problem corresponding to the half space $\mathbf{n}_k^\mathsf{T}\mathbf{u} \geq b_k$ reads:

$$\begin{aligned} \mathbf{u}_{k+1}^{+} = \arg\min_{\mathbf{u}} \; & \Phi_m(\mathbf{u}) = \Phi(\mathbf{u}) + {\boldsymbol{\lambda}_k^{\Phi}}^\mathsf{T}\mathbf{u} \\ \text{s.t.} \quad & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}} \\ & \mathbf{n}_k^\mathsf{T}\mathbf{u} \geq b_k + \rho_k\|\mathbf{n}_k\|, \end{aligned} \tag{40}$$

and for the half space $\mathbf{n}_k^\mathsf{T}\mathbf{u} \leq b_k$:

$$\begin{aligned} \mathbf{u}_{k+1}^{-} = \arg\min_{\mathbf{u}} \; & \Phi_m(\mathbf{u}) = \Phi(\mathbf{u}) + {\boldsymbol{\lambda}_k^{\Phi}}^\mathsf{T}\mathbf{u} \\ \text{s.t.} \quad & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}} \\ & \mathbf{n}_k^\mathsf{T}\mathbf{u} \leq b_k - \rho_k\|\mathbf{n}_k\|. \end{aligned} \tag{41}$$

The modifiers $\boldsymbol{\lambda}_k^{\Phi}$ are adapted according to (8). The next operating point is chosen as the value of $\{\mathbf{u}_{k+1}^{+}, \mathbf{u}_{k+1}^{-}\}$ that minimizes the modified cost function $\Phi_m(\mathbf{u})$.

Initialization. In order to initialize the dual modifier-adaptation scheme, it is necessary to have $(n_u + 1)$ operating points satisfying the constraint $\mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}}$ so as to to compute a first estimate of the gradient and evaluate the gradient modifiers. One possibility is to generate the initial $(n_u + 1)$ operating points by considering deviations from the initial point along the Cartesian axes, as in the FFD scheme (9). This technique proposed for the dual ISOPE algorithm [5, 6] is retained here. Notice that an optimized initial phase has also been suggested for the dual ISOPE algorithm [6].

5 Dual Modifier Adaptation for Constrained Optimization

In constrained optimization problems, the modifier-adaptation approach requires an estimate of the cost and constraint gradients at each iteration. In order to implement dual modifier adaptation with a bound on the gradient error norm, one has to decide which gradient error to consider. The cost and constraint functions are evaluated from noisy measurements with different noise levels; furthermore, the curvature of all these functions also differ. Implementation of the upper bound on the gradient error norm introduced in

Section 3.4 requires the selection of the parameters δ and σ_{max} corresponding to the function of interest $\psi(\mathbf{u})$ in (10). In the case of an unconstrained optimization problem, this function was selected as the cost function. In the case of constrained optimization problems, either one of two possible strategies can be considered:

5.1 Two Strategies

Strategy 1. Determine values of δ and σ_{max} for the cost function and each of the constraints individually, and select the largest values of δ and σ_{max}. This strategy guarantees that the upper bound on the gradient error given by $\mathcal{E}(\mathbf{u})$ is valid for the cost function and for each constraint individually. However, it introduces extra conservatism, leading to the implementation of smaller than necessary regions for positioning the new operating point.

Strategy 2. Select the function $\psi(\mathbf{u})$ as a linear combination of the cost and constraint functions:

$$\begin{aligned}\tilde{\psi}(\mathbf{u}) &= \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) + \mathbf{c}^{\mathsf{T}}\mathbf{g}(\mathbf{u}, \mathbf{y}_p(\mathbf{u}) + \boldsymbol{\nu}) \\ &= \Phi_p(\mathbf{u}) + \mathbf{c}^{\mathsf{T}}\mathbf{G}_p(\mathbf{u}) + v. \end{aligned} \tag{42}$$

In particular, one natural choice is to select the weights on the constraint functions in (42) as the Lagrange multipliers, that is, $\mathbf{c} = \boldsymbol{\mu}$, in which case, the function $\psi(\mathbf{u})$ corresponds to the Lagrangian function. Interestingly, since the Lagrange multipliers associated with inactive constraints are zero, this choice eliminates from $\psi(\mathbf{u})$ the inactive constraints. It is clear that, if a constraint does not become active in modifier adaptation, the error with which its gradient is estimated will not influence the optimization behavior. However, the difficulty with such a choice is that the active constraints are not known a priori. One way around this difficulty is to update the values of $\mathbf{c}$ as $\mathbf{c}_k = \boldsymbol{\mu}_k$. Notice that this will modify the noise level v at each RTO iteration, hence, the values of δ and σ_{max} could also be updated at each RTO step to compute the upper bound $\mathcal{E}(\mathbf{u})$. The difficulty in doing so is that, if δ_k and $\sigma_{max,k}$ are allowed to vary, there is a risk that the constraint (27) becomes infeasible since the past operating points were placed using different values of δ_k and $\sigma_{max,k}$. Indeed, infeasibility can occur if the values of δ_k or $\sigma_{max,k}$ decrease from one RTO iteration to the next. Ways to deal with these infeasibilities will be the subject of future research. If the problem becomes infeasible, it is possible to either (i) reinitialize the algorithm with a FFD gradient estimation, which requires generating n_u new operating points, or (ii) increase the value of the upper bound $\mathcal{E}^{\mathrm{U}}$.

Once the parameters δ and σ_{max} are selected, the following modified optimization problems including constraints are solved on each side of the hyperplane generated by the n_u most recent operating points, $\mathbf{n}_k^\mathsf{T}\mathbf{u} = b_k$. The optimization problem corresponding to the half space $\mathbf{n}_k^\mathsf{T}\mathbf{u} \geq b_k$ is:

$$\begin{aligned} \mathbf{u}_{k+1}^{+} = \arg\min_{\mathbf{u}} \; & \Phi_m(\mathbf{u}) = \Phi(\mathbf{u}) + \boldsymbol{\lambda}_k^{\Phi\mathsf{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}) = \mathbf{G}(\mathbf{u}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^{\mathbf{G}\mathsf{T}}(\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\ & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}} \\ & \mathbf{n}_k^\mathsf{T}\mathbf{u} \geq b_k + \rho_k \|\mathbf{n}_k\|, \end{aligned} \tag{43}$$

whereas, for the other half space, $\mathbf{n}_k^\mathsf{T}\mathbf{u} \leq b_k$,

$$\begin{aligned} \mathbf{u}_{k+1}^{-} = \arg\min_{\mathbf{u}} \; & \Phi_m(\mathbf{u}) = \Phi(\mathbf{u}) + \boldsymbol{\lambda}_k^{\Phi\mathsf{T}}\mathbf{u} \\ \text{s.t.} \quad & \mathbf{G}_m(\mathbf{u}) = \mathbf{G}(\mathbf{u}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^{\mathbf{G}\mathsf{T}}(\mathbf{u} - \mathbf{u}_k) \leq \mathbf{0} \\ & \mathcal{E}(\mathbf{u}) \leq \mathcal{E}^{\mathrm{U}} \\ & \mathbf{n}_k^\mathsf{T}\mathbf{u} \leq b_k - \rho_k \|\mathbf{n}_k\|. \end{aligned} \tag{44}$$

The modifiers $\boldsymbol{\varepsilon}_k$, $\boldsymbol{\lambda}_k^{\mathbf{G}}$ and $\boldsymbol{\lambda}_k^{\Phi}$ are adapted as in (8). The next operating point is chosen as the value of $\{\mathbf{u}_{k+1}^{+}, \mathbf{u}_{k+1}^{-}\}$ that minimizes the augmented cost function $\Phi_m(\mathbf{u})$.

It might happen that one of the optimization problems (43) and (44) be infeasible. However, if the plant is not subjected to sudden perturbation nor change in the operating conditions that might drastically shift the constraints, it will seldom be the case that both problems (43) and (44) become infeasible simultaneously. Indeed, when facing constraints, adaptation can always return the same way it came. Therefore, given the most recent operating points $\mathbf{u}_k, \mathbf{u}_{k-1}, \ldots, \mathbf{u}_{k-n_u+1}$, a new operating point $\mathbf{u}_{k+1}$ satisfying $\mathcal{E}(\mathbf{u}_{k+1}) \leq \mathcal{E}^{\mathrm{U}}$ can always be found as $\mathbf{u}_{k+1} = \mathbf{u}_{k-n_u}$.

Also, since most numerical optimization solvers require a feasible initial guess, a procedure is required to find a feasible initial guess prior to each optimization and deal with infeasibility of the problems (43) and (44). This infeasibility issue, which is also inherent to the dual ISOPE approach, has not been addressed in the literature.

In summary, when constraints are included in the optimization problem, two main issues arise regarding the implementation of dual modifier adaptation with a bound on the gradient error. The first issue is tied to the strategy used to select the parameters δ and σ_{max}. The second issue regards the way to deal with possible infeasibility in optimization problems (43) and (44).

6 Illustrative Examples

The dual modifier-adaptation approach is now illustrated through two simulated examples. The first example considers the RTO of the unconstrained CSTR reactor in the Williams-Otto plant, while the second example is a constrained optimization problem.

6.1 Williams-Otto Reactor

6.1.1 Reactor and model

The reactor in the Williams-Otto plant [13], as modified by Roberts [14], is considered. This reactor example has also been used to illustrate model adequacy and RTO performance [15, 16]. It consists of an ideal CSTR in which the following reactions occur:

$$\begin{aligned} A + B &\longrightarrow C & k_1 &= 1.660 \times 10^6 e^{-6666.7/(T_R+273.15)} \\ C + B &\longrightarrow P + E & k_2 &= 7.212 \times 10^8 e^{-8333.3/(T_R+273.15)} \\ C + P &\longrightarrow G & k_3 &= 2.675 \times 10^{12} e^{-11111/(T_R+273.15)} \end{aligned}$$

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:

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

where X_P and X_E represent the concentrations of the products P and E. 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 $\mathbf{u} = [F_B \;\; T_R]^\mathsf{T}$.

In this example, the aforementioned reaction scheme corresponds to the simulated reality. However, since it is assumed that the reaction system is not well understood, the following two reactions are used in the model of the reactor [15]:

$$\begin{aligned} A + 2B &\longrightarrow P + E & k_1 &= 2.189 \times 10^8 e^{-8077.6/(T_R+273.15)} \\ A + B + P &\longrightarrow G & k_2 &= 4.310 \times 10^{13} e^{-12438/(T_R+273.15)}. \end{aligned}$$

The material balance equations for the plant and the approximate model can be found in [16].

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 σ_{max} obtained with the scaled inputs is 1030 for the model and 1221 for the (unknown) plant. 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 (8) is implemented for the cost gradient modifiers with $\mathsf{K} = 0.5\,\mathbf{I}_2$.

6.1.2 Modifier Adaptation using FFD

Modifier adaptation is first 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. In this two-dimensional case, the shortest distance between complement affine subspaces for the FFD arrangement is $l^{\min} = \frac{1}{\sqrt{2}}h$. Hence, from (21) and (24), one can write:

$$\mathcal{E}(h) = \frac{\sqrt{2}}{2}\ h\ \sigma_{max} + \sqrt{2}\frac{\delta}{h}. \tag{45}$$

The step size that minimizes $\mathcal{E}(h)$ is $h^\star = \sqrt{\frac{2\ \delta}{\sigma_{max}}} = 0.0763$ (scaled value), for which $\mathcal{E}(h^\star) = 111.2$.

Figure 6a shows an input trajectory. The observed offset with respect to the plant optimum results mainly from the gradient error due to truncation.

6.1.3 Dual Modifier Adaptation with Bound on Gradient Error

Dual modifier adaptation is now applied with $\mathcal{E}^U = 111.2$ (same value as above). The algorithm is initialized using FFD with $h^\star = 0.0763$. Figure 6b shows a realization of the input trajectory. Compared with modifier adaptation using FFD, significantly fewer iterations are required to approach the optimum.

Figure 7a shows the evolution of the plant profit and the gradient error norm for 20 noise realizations. To simulate a process change, the flowrate F_A is increased from 1.8275 kg/s to 2.2 kg/s at iteration 20. Modifier adaptation tracks the change in plant optimum. It can be seen in the upper plot of Figure 7a that the neighborhood of the new optimal profit is reached within 6 iterations for all 20 realizations. The lower plot of Figure 7a 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,

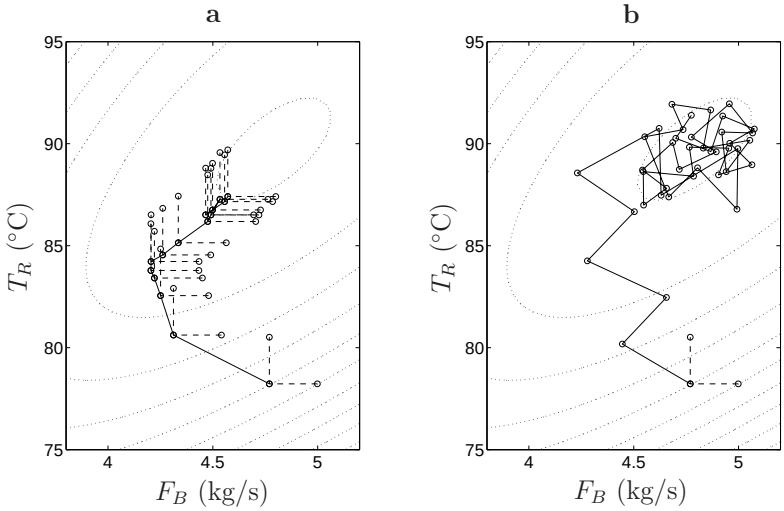

Fig. 6. Input trajectory with 45 operating points. The dotted lines represent the contours of the plant cost function. (**a**) Modifier adaptation using FFD. (**b**) Dual modifier adaptation with bound on gradient error.

the computed gradient is inconsistent because it is estimated using operating points corresponding to different values of F_A.

6.1.4 Dual Modifier Adaptation with Bound on Condition Number

For the sake of comparison, dual modifier adaptation is applied with a lower bound on the inverse condition number of $\mathcal{U}(\mathbf{u})$, given by (25), as proposed for the dual ISOPE approach [5, 6]. The results are shown in Figure 7b. The lower bound of $\varphi = 0.4$ 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 region given by the condition number constraint decreases proportionally 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.

6.2 Constrained Optimization Problem

Consider the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{u}\geq\mathbf{0}} \; & \Phi(\mathbf{u},\boldsymbol{\theta}) := (u_1-\theta_1)^2 + 4(u_2-2.5)^2 \\ \text{s.t.} \quad & G := u_1^2 + \theta_2 u_1 + \theta_3 u_2 + \theta_4 \leq 0, \end{aligned} \tag{46}$$

comprising two decision variables $\mathbf{u} = [u_1 \; u_2]^\mathsf{T}$, four model parameters $\boldsymbol{\theta} = [\theta_1 \; \theta_2 \; \theta_3 \; \theta_4]^\mathsf{T}$, and a single inequality constraint G. The parameter values $\boldsymbol{\theta}$

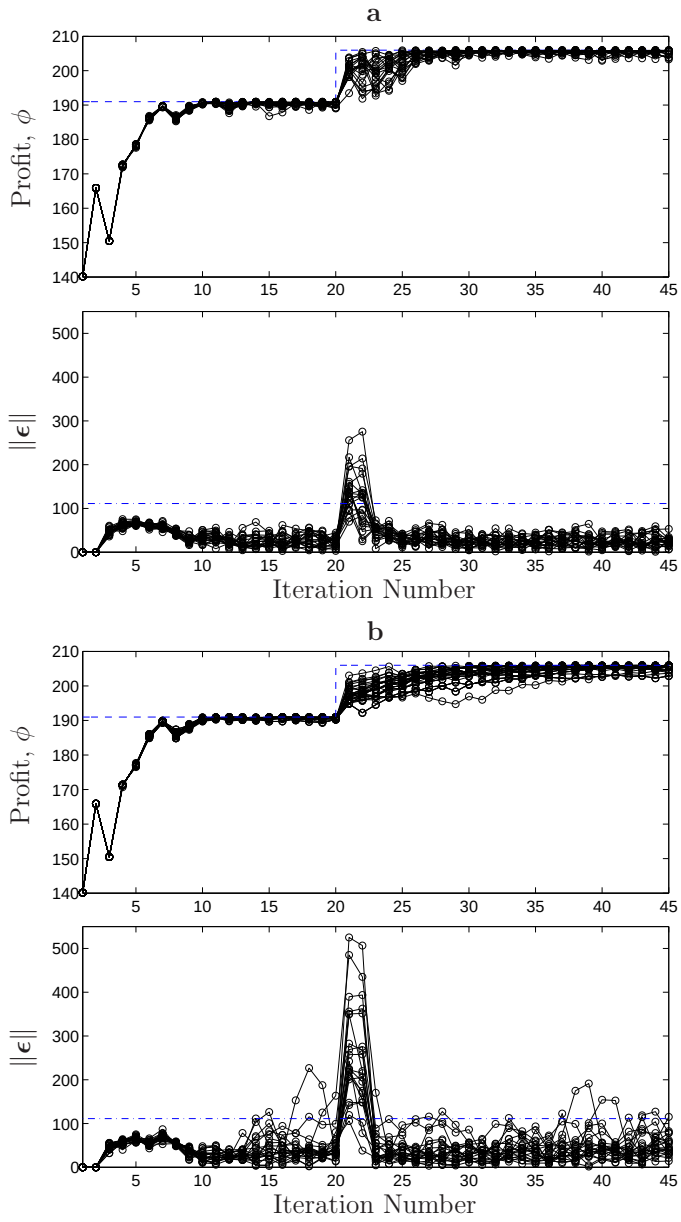

Fig. 7. Optimization for 20 noise realizations. A perturbation in the form of a flowrate change is introduced at iteration 20. (**a**) Dual modifier adaptation with bound on gradient error norm. (**b**) Dual modifier adaptation with bound on the condition number. Dashed line: Optimal profit for the plant. Dash-dotted line: $\mathcal{E}^U = 111.2$.

for the plant (simulated reality) and the model are reported in Table 1. This problem has been used to illustrate the modifier-adaptation approach in [3].

Because of measurement noise, both the cost function and the constraint are assumed to be Gaussian random variables with standard deviations $\sigma_\phi = \sigma_G = 0.02$. Hence, the noise v corresponding to the Lagrangian function has a Gaussian distribution with standard deviation $\sigma_\mathcal{L} = \sqrt{\sigma_\phi^2 + \mu^2\sigma_G^2}$. We select

Table 1
Values of the uncertain parameters $\boldsymbol{\theta}$ in Problem (46) for the plant and the model.

	θ_1	θ_2	θ_3	θ_4
Plant	3.5	-5	1.6	2.65
Model	2.0	-3	2	-0.75

$\mu = 4$, which is the value of the Lagrange multiplier at the model optimum (the unknown value of μ at the plant optimum is 1.68). With this choice, the value of σ_{max} corresponding to the Lagrangian function is 10. The noise interval $\delta = 6\,\sigma_{\mathcal{L}}$ is chosen. The exponential filter (8) is implemented with $\mathsf{K} = 0.8\,\mathbf{I}_5$.

Dual modifier adaptation is applied with $\mathcal{E}^U = 5.5$. The algorithm is initialized using FFD with $h = 0.16$, which is obtained as the smallest value solution to (45) with $\mathcal{E}(h) = \mathcal{E}^U$. Figure 8 shows a realization of the input trajectory. The algorithm converges to a region of input space that contains the plant optimum (point P). To avoid constraint violation due to measurement noise and perturbation of the inputs, a backoff from the constraint could be implemented.

We observed in the unconstrained example that, in spite of the perturbations around the plant optimum seen in Figure 6b, the profit is not strongly affected as shown in Figure 7a. On the other hand, in constrained optimization problems with active constraints, the perturbations required to estimate the gradients accurately will always produce movements away from the constraints, thereby resulting in a larger performance loss. This can be observed in Figure 8 upon "convergence" to a region that contains the plant optimum.

7 Conclusions

This paper has developed a rigorous upper bound on the gradient error norm that can be used in RTO schemes for positioning the next operating point with respect to past operation. This constraint takes into account the effect of truncation errors and measurement noise. The evaluation of the constraint requires the selection of two parameters corresponding to the function(s) for which the gradient is being estimated: δ, which is representative of the level of measurement noise, and σ_{max}, which is an upper bound on the curvature of $\psi(\mathbf{u})$. The proposed approach is based on error norms and thus is fairly conservative.

The Williams-Otto reactor example has demonstrated the potential of dual modifier adaptation, which pays attention to the accuracy with which the gra-

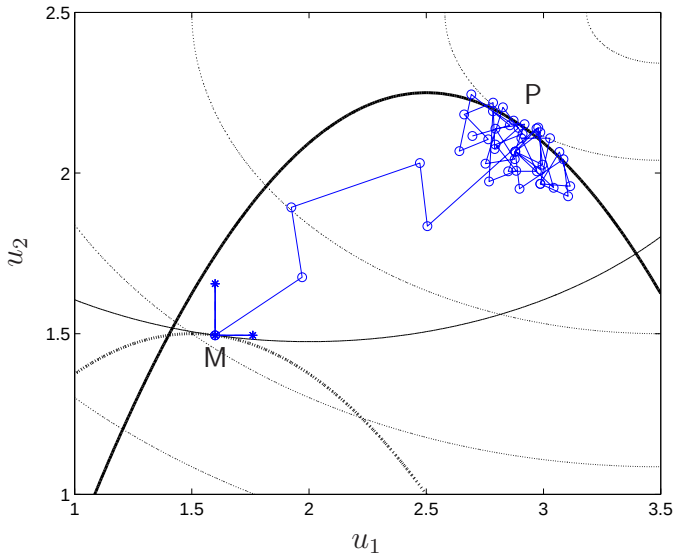

Fig. 8. Modifier adaptation applied to Problem (46). Dotted lines: contours of the cost function for the plant; Thick solid line: constraint for the plant; Thin solid line: contour of the cost function for the model; Thick dash-dotted line: constraint predicted by the model; Point P: plant optimum; Point M: model optimum.

dients are estimated. In this application, dual modifier adaptation performed better than modifier adaptation using the FFD approach. Also, the proposed dual modifier adaptation with a bound on the gradient error produces more accurate gradient estimates than with simply bounding the condition number of $\mathcal{U}(\mathbf{u})$. 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 curvature of the cost or Lagrangian function, and the gain matrix K used to filter the modifiers.

The gradients are estimated by numerical approximation using $(n_u + 1)$ operating points. Hence, reasonable gradient estimates can be obtained provided the frequency of the disturbances affecting the plant is sufficiently low with respect to the time required for the plant to reach steady state $(n_u + 1)$ times. This clearly limits the applicability of this approach, in particular for systems with a large number of inputs. Finally, the upper bound developed, albeit comprehensive and consistent, suffers from its complexity. For instance, the evaluation of the upper bound on the gradient error due to measurement noise requires the evaluation of a number of distances between complement affine subspaces, which quickly increases with the number of inputs. Clearly, faster and more accurate ways of estimating the gradients are desirable, and they represent the subject of future research. This is critical in order for modifier adaptation to shift from an ad hoc operation improvement strategy to a rigorous methodology amenable to industrial-scale applications.

References

[1] T. E. Marlin, A. N. Hrymak, Real-time operations optimization of continuous processes, in: AIChE Symposium Series - CPC-V, Vol. 93, 1997, pp. 156–164.

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

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

[4] M. Mansour, J. E. Ellis, Comparison of methods for estimating real process derivatives in on-line optimization, App. Math. Modelling 27 (2003) 275–291.

[5] M. Brdyś, P. Tatjewski, An algorithm for steady-state optimizing dual control of uncertain plants, in: Proc. 1st IFAC Workshop on New Trends in Design of Control Systems, Smolenice, Slovakia, 1994, pp. 249–254.

[6] M. Brdyś, P. Tatjewski, Iterative Algorithms for Multilayer Optimizing Control, Imperial College Press, London UK, 2005.

[7] B. Wittenmark, Adaptive dual control methods: an overview, in: 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing, 1995, pp. 67–72.

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

[9] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, Nonlinear Programming: Theory and Algorithms, 3rd Edition, John Wiley and Sons, New Jersey, 2006.

[10] L. T. Biegler, I. E. Grossmann, A. W. Westerberg, A note on approximation techniques used for process optimization, Comp. Chem. Eng. 9 (2) (1985) 201–206.

[11] A. Marchetti, Modifier-adaptation methodology for real-time optimization, Ph.D. thesis, Ecole Polytechnique Fédérale de Lausanne (2009).

[12] R. C. M. Brekelmans, L. T. Driessen, H. L. M. Hamers, D. den Hertog, Gradient estimation schemes for noisy functions, J. Opt. Th. Appl. 126 (3) (2005) 529–551.

[13] T. J. Williams, R. E. Otto, A generalized chemical processing model for the investigation of computer control, AIEE Trans. 79 (1960) 458.

[14] P. D. Roberts, An algorithm for steady-state system optimization and parameter estimation, J. System Science 10 (1979) 719–734.

[15] J. F. Forbes, T. E. Marlin, J. F. MacGregor, Model adequacy requirements for optimizing plant operations, Comp. Chem. Eng. 18 (6) (1994) 497–510.

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

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

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

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

Complement affine subspaces. Given a set of (n_u+1) points in a n_u-dimensional space, $\mathcal{S} := \{\mathbf{u}_1, \ldots, \mathbf{u}_{n_u+1}\}$, a proper subset $\mathcal{S}^A$, i.e. $\mathcal{S}^A \subsetneq \mathcal{S}$, of $n_u^A \in \{1, \ldots, n_u\}$ points generates an affine subspace of dimension $(n_u^A - 1)$:

$$\mathbf{u} = \mathbf{u}_1 + \lambda_{1,2}\frac{\mathbf{u}_1 - \mathbf{u}_2}{\|\mathbf{u}_1 - \mathbf{u}_2\|} + \cdots + \lambda_{1,n_u^A}\frac{\mathbf{u}_1 - \mathbf{u}_{n_u^A}}{\|\mathbf{u}_1 - \mathbf{u}_{n_u^A}\|} \tag{A.2}$$

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

$$\begin{aligned}\mathbf{u} = \mathbf{u}_{n_u^A+1} + \lambda_{n_u^A+1,n_u^A+2}\frac{\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2}}{\|\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2}\|} + \ldots \\ \cdots + \lambda_{n_u^A+1,n+1}\frac{\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1}}{\|\mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1}\|}\end{aligned} \tag{A.3}$$

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, $\mathcal{S} := \{\mathbf{u}_1, \ldots, \mathbf{u}_{n_u+1}\}$, a proper subset of $\mathcal{S}$, $\mathcal{S}^A \subsetneq \mathcal{S}$ of $n_u^A \in \{1, \ldots, n_u\}$ points, and its complement $\mathcal{S}^C := \mathcal{S} \backslash \mathcal{S}^A$ 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 $l_{min} := min\{l_1, l_2, \ldots, l_{n_b}\}$,*

where $l_1, l_2, \ldots, l_{n_b}$ 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:

$$\begin{aligned}[\mathbf{u}_1 - \mathbf{u}_2 \ \ldots \ \mathbf{u}_1 - \mathbf{u}_{n_u^A} \ \ \mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u^A+2} \ \cdots \\ \mathbf{u}_{n_u^A+1} - \mathbf{u}_{n_u+1} \ ]^\mathsf{T}\mathbf{n} = \mathbf{0}, \qquad \text{or,} \qquad \mathsf{U}\mathbf{n} = \mathbf{0}.\end{aligned} \tag{A.4}$$

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

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