Computers and Chemical Engineering 164 (2022) 107867

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

On improving the efficiency of modifier adaptation via directional information

D. Rodrigues [a,*], A.G. Marchetti [b], D. Bonvin [c]

[a] Centro de Química Estrutural, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal
[b] CIFASIS (CONICET - Universidad Nacional de Rosario), Rosario S2000EZP, Argentina
[c] Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

ARTICLE INFO

Article history:
Received 28 August 2021
Revised 18 March 2022
Accepted 29 May 2022
Available online 31 May 2022

Keywords:
Real-time optimization
Plant-model mismatch
Modifier adaptation
Active set
Dominant gradients

ABSTRACT

In real-time optimization, the solution quality depends on the model ability to predict the *plant* Karush–Kuhn–Tucker (KKT) conditions. In the case of non-parametric plant-model mismatch, one can add input-affine modifiers to the model cost and constraints as is done in modifier adaptation (MA). These modifiers require estimating the plant cost and constraint gradients. This paper discusses two ways of reducing the number of input directions, thereby improving the efficiency of MA in practice. The first approach capitalizes on the knowledge of the *active set* to reduce the number of KKT conditions. The second approach determines the *dominant gradients* using sensitivity analysis. This way, MA reaches near plant optimality efficiently by adapting the first-order modifiers only along the dominant input directions. These approaches allow generating several alternative MA schemes, which are analyzed in terms of the number of degrees of freedom and compared in a simulated study of the Williams–Otto plant.

1. Introduction

Real-time optimization (RTO) uses process measurements to improve the performance of optimization methods in industrial processes. Hence, RTO avoids relying exclusively on a (possibly inaccurate) process model. The RTO methods proposed in the literature can be divided into two classes depending on whether or not a process model is used in real time (Srinivasan et al., 2003). If a model is used, it can be updated iteratively based on real-time measurements and then used for numerical optimization. Since an optimization problem is solved at each iteration, such an approach is labeled *explicit RTO*. In contrast, optimality can also be enforced in real time via feedback control. Such an approach, which does not require solving an optimization problem in real time, is labeled *implicit RTO*. This paper is concerned with the model-update step in explicit RTO. The two main structuring decisions deal with the choices of (i) plant information that can be collected and is relevant to optimality (what to measure or estimate from measurements?), and (ii) the degrees of freedom (dof) available in the model (what to adapt?). A classification of explicit RTO schemes based on these two issues is given in Table 1 and discussed next.

- *Fit plant outputs.* The "two-step approach" (TS; Chen and Joseph, 1987; Marlin and Hrymak, 1997) advocates measuring plant outputs (such as temperatures and concentrations in a reactor) and adapting model parameters (such as kinetic parameters) so as to force the model outputs to fit the plant outputs. However, this approach has the major drawback that model adaptation does not actively seek to match the plant optimality conditions, in this case the first-order Karush–Kuhn–Tucker (KKT) conditions that include constrained values as well as cost and constraint gradients (Bazaraa et al., 2006). As a result, TS often results in sub-optimal performance in the presence of structural plant-model mismatch.
- *Fit plant outputs and selected KKT conditions.* The fact that TS typically does not converge to plant optimality motivated the development of modified TS approaches that attempt to match selected KKT conditions in addition to the outputs.
  - A first scheme is "Integrated System Optimization and Parameter Estimation" (ISOPE; Roberts, 1995; Brdyś and Tatjewski, 2005) that was initially developed for problems without uncertainty in the constraints, for example in the absence of process-dependent constraints. ISOPE uses an estimate of the plant cost gradient to modify the cost function of the optimization problem and enforce KKT matching between the modified model and the plant.

* Corresponding author.
E-mail address: dfmr@tecnico.ulisboa.pt (D. Rodrigues).

https://doi.org/10.1016/j.compchemeng.2022.107867
0098-1354/

Table 1
Classification of explicit RTO schemes based on the type of model update: vertical, the plant information that the model has to fit; horizontal, the model handles, that is, the available degrees of freedom for model adaptation. The various methods are the two-step approach (TS), simultaneous model identification and optimization (SMIO), integrated system optimization and parameter estimation (ISOPE), parameter adaptation (PA), and modifier adaptation (MA).

		Model handles: available degrees of freedom		
		Model parameters (n_θ parameters)	Modifiers (n_m modifiers)	Model parameters + Selected modifiers
Plant information to be fitted by model	Outputs (n_y outputs)	TS $n_\theta \rightarrow n_y$	-	-
	Outputs + Selected KKT conditions	SMIO $n_\theta \rightarrow (n_y + n_{grad})$	-	ISOPE $(n_\theta + n_u) \rightarrow (n_y + n_u)$
	All active KKT conditions (n_a conditions)	PA $n_\theta \rightarrow n_a$	MA $n_m \rightarrow n_a$	-

Dimensions: n_u is the number of inputs, n_y the number of outputs, n_θ the number of model parameters, n_m the number of modifiers, n_{grad} the number of gradients in the KKT conditions, and n_a the number of *active* KKT conditions. Note that n_m, n_{grad} and n_a depend on the formulation of the optimization problem (either standard, output, or Lagrangian) and on whether the active set is known or not (details are given in Section 2.2). Reading example for TS: $n_\theta \rightarrow n_y$ means that n_θ model parameters are available to fit n_y plant outputs.

  - Another scheme is "Simultaneous Model Identification and Optimization" (SMIO; Mandur and Budman, 2015) that adapts the model parameters in two steps, first to match the outputs as well as possible, followed by another limited correction to try to get closer to the cost and constraint gradients. As there are often too few degrees of freedom to match both the outputs and the gradients, the resulting parameter values represent a compromise between the two objectives.
- *Fit all active KKT conditions.* For plant optimality, one would like to monitor all plant KKT conditions and track the active ones. However, since the active KKT conditions are typically unknown, one needs to consider all KKT conditions. Also, usually, there is no one-to-one correspondence between model parameters and KKT conditions, which means that a given model parameter can affect none, one, or several KKT conditions.
  - If all n_a active plant KKT conditions can be matched by adjusting the n_θ model parameters, then RTO is implemented iteratively by adapting the model parameters to match the measured (or estimated from measurements) plant KKT conditions locally at each iteration. This scheme is labeled here "parameter adaptation" (PA). However, this situation is rarely encountered in practice as n_a is often much larger than n_θ.
  - In "modifier adaptation" (MA; Marchetti et al., 2016), model parameters are kept fixed, but input-affine modifier terms are added to the model cost and constraints so that the KKT conditions of the modified model match those of the plant locally. Each modifier is used to fit a single KKT element without affecting the other elements. Since there are as many modifiers as there are KKT elements, MA can achieve full KKT matching upon convergence (Marchetti et al., 2009).

This paper investigates the use of modifiers to correct the model and drive the plant to, or near to, optimality. Over the years, several alternative MA formulations have been proposed. The initial (standard) formulation adds input-affine modifier terms to both the cost and constraint functions (Marchetti et al., 2009). It has also been suggested to put the modifiers on the output functions instead (MAy; Papasavvas et al., 2019). Another option is to add the modifiers to the Lagrangian function (MA-$\mathcal{L}$; Marchetti et al., 2016). Although these formulations are not equivalent, they all lead to KKT matching upon convergence. Note that these three formulations are fairly general and do not require prior knowledge of the active set.

This paper also discusses the fact that, for good performance, it is often not necessary to consider all active KKT conditions since only a few of them are dominant. Tracking all KKT conditions requires estimating all gradient elements at each RTO iteration, which can be experimentally expensive. This calls for welcome simplifications, in particular when (i) the active set is known, and (ii) the dominant input directions can be determined. For the latter case, Costello et al. (2016) proposed a method to improve the efficiency of gradient estimation by estimating plant derivatives only along a few dominant input directions. The computation of these directions relies on computing the local sensitivity of the Lagrangian gradient with respect to perturbations of the uncertain parameters. The resulting scheme has been labeled "directional modifier adaptation" (DMA). Along the same lines, Singhal et al. (2018) proposed a global sensitivity analysis to rank the input and parameter directions according to their effect in the whole parameter space. The scheme has been labeled "active directional modifier adaptation" (ADMA). Despite this relevant work, the reduction in the number of input directions and KKT conditions remains an important challenge.

The contribution of this paper is threefold: (i) use the knowledge of the active set to reduce the number of KKT conditions and propose novel known-active-set schemes since, although the corresponding KKT conditions are known (see for instance Marchetti et al., 2020), these schemes have not been proposed in the literature; (ii) generalize the use of dominant gradients obtained via global sensitivity analysis and propose novel ADMA schemes, some of them in combination with the knowledge of the active set; and (iii) compare the various schemes in terms of both the number of degrees of freedom and performance in a relevant simulated study, which allows the reader to perceive the differences between the various methods as well as their advantages and disadvantages with respect to each other.

The paper is organized as follows. Section 2 formulates the static RTO problem, proposes three sets of KKT conditions, and briefly reviews the corresponding MA schemes. Section 3 proposes to use directional information to increase the efficiency of MA schemes. The developed concepts are illustrated in Section 4 via a simulated study of the Williams–Otto plant, while Section 5 concludes the paper. There is also an appendix dealing with parametric sensitivity analysis and the concept of active subspaces.

2. Preliminaries

2.1. Formulation of the optimization problem

The optimization problem for the plant reads

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

$$\text{s.t.} \quad G_{j,p}(\mathbf{u}) := g_j\big(\mathbf{u}, \mathbf{y}_p(\mathbf{u})\big) \le 0, \quad j = 1, \dots, n_g, \tag{1b}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the input vector, $\mathbf{y}_p \in \mathbb{R}^{n_y}$ the output vector, $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ the cost to be minimized, and $g_j : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ the j^{th} constraint.

The main challenge in solving this optimization problem stems from the fact that the static map $\mathbf{y}_p(\mathbf{u})$, which relates the inputs to the plant outputs, is unknown. However, an approximate process model is assumed to be available in the form of the parametrized input-output map $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$, where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are the model parameters. Using this model, Problem (1) can be approximated by the model-based optimization problem

$$\min_{\mathbf{u}} \quad \Phi(\mathbf{u}, \boldsymbol{\theta}) := \phi\big(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\big) \tag{2a}$$

$$\text{s.t.} \quad G_j(\mathbf{u}, \boldsymbol{\theta}) := g_j\big(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})\big) \le 0, \quad j = 1, \dots, n_g. \tag{2b}$$

The optimal input vector $\mathbf{u}^\star$ is found by solving Problem (2). In the presence of plant-model mismatch, the model optimum $\mathbf{u}^\star$ differs from the plant optimum $\mathbf{u}_p^\star$. Hence, the aim of RTO is to find $\mathbf{u}_p^\star$ by iteratively adapting and solving Problem (2).

2.2. KKT conditions

Local minima of Problems (1) and (2) are characterized by the corresponding necessary conditions of optimality. There are several different but equivalent formulations as shown next. In this section, it is assumed that linear independence constraint qualification is satisfied.

2.2.1. Standard formulation

For $\mathbf{u}^\star$ to be a local minimum of Problem (2), there must exist a vector $\boldsymbol{\nu}^\star = [\nu_1^\star, \dots, \nu_{n_g}^\star]^\mathsf{T}$ such that the following KKT conditions hold (Bazaraa et al., 2006):[1]

$$\frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}^\star, \boldsymbol{\theta}) + \sum_{j=1}^{n_g} \nu_j^\star \frac{\partial G_j}{\partial \mathbf{u}}(\mathbf{u}^\star, \boldsymbol{\theta}) = \mathbf{0} \tag{3a}$$

$$G_j(\mathbf{u}^\star, \boldsymbol{\theta}) \le 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{3b}$$

$$\nu_j^\star G_j(\mathbf{u}^\star, \boldsymbol{\theta}) = 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{3c}$$

$$\nu_j^\star \ge 0, \qquad \forall j \in \{1, \dots, n_g\}. \tag{3d}$$

Note that it is possible to express this set of KKT conditions differently as shown next.

2.2.2. Output formulation

The cost $\Phi(\mathbf{u}, \boldsymbol{\theta})$ and the constraints $\mathbf{G}(\mathbf{u}, \boldsymbol{\theta})$ are functions of the outputs $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta})$ as per Eq. (2). It follows that the KKT conditions (3) can be expressed in terms of the outputs $\mathbf{y}$ and the output gradients $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}$ as follows (see Papasavvas et al., 2019 for more details):

$$\begin{aligned}\frac{\partial \phi}{\partial \mathbf{u}}\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) + \frac{\partial \phi}{\partial \mathbf{y}}\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}^\star, \boldsymbol{\theta}) \\ + \sum_{j=1}^{n_g} \nu_j^\star \bigg[\frac{\partial g_j}{\partial \mathbf{u}}\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) \\ + \frac{\partial g_j}{\partial \mathbf{y}}\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}^\star, \boldsymbol{\theta}) \bigg] = \mathbf{0}\end{aligned} \tag{4a}$$

$$g_j\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) \le 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{4b}$$

$$\nu_j^\star g_j\big(\mathbf{u}^\star, \mathbf{y}(\mathbf{u}^\star, \boldsymbol{\theta})\big) = 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{4c}$$

$$\nu_j^\star \ge 0, \qquad \forall j \in \{1, \dots, n_g\}. \tag{4d}$$

Although this output formulation encompasses more KKT elements than the standard formulation (3), it is of practical interest since it involves the directly accessible outputs $\mathbf{y}$ and their gradients $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}$.

2.2.3. Lagrangian formulation

One can consider the Lagrangian function,

$$\mathcal{L}(\mathbf{z}, \boldsymbol{\theta}) := \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\nu}^\mathsf{T} \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}), \tag{5}$$

where $\mathbf{G} \in \mathbb{R}^{n_g}$ is the vector of model constraints, $\boldsymbol{\nu} \in \mathbb{R}^{n_g}$ the vector of Lagrange multipliers, and $\mathbf{z} := \begin{bmatrix} \boldsymbol{\nu} \\ \mathbf{u} \end{bmatrix} \in \mathbb{R}^{n_g + n_u}$. With this formulation, the KKT conditions read (see Marchetti et al., 2016 for more details):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{z}^\star, \boldsymbol{\theta}) = \mathbf{0} \tag{6a}$$

$$G_j(\mathbf{u}^\star, \boldsymbol{\theta}) \le 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{6b}$$

$$\nu_j^\star G_j(\mathbf{u}^\star, \boldsymbol{\theta}) = 0, \qquad \forall j \in \{1, \dots, n_g\} \tag{6c}$$

$$\nu_j^\star \ge 0, \qquad \forall j \in \{1, \dots, n_g\}. \tag{6d}$$

For the plant optimization problem, the Lagrangian function is $\mathcal{L}_p(\mathbf{z}) := \Phi_p(\mathbf{u}) + \boldsymbol{\nu}^\mathsf{T} \mathbf{G}_p(\mathbf{u})$. Note that the optimal plant Lagrange multipliers $\boldsymbol{\nu}_p^\star$ are typically unknown. If needed, for instance to compute the plant Lagrangian, they can be approximated by the model value $\boldsymbol{\nu}^\star$.

Corresponding KKT elements. The KKT conditions involve the constrained values as well as the cost and constraint gradients. These conditions include $n_g + n_u$ equalities and $2n_g$ inequalities to determine $n_g + n_u$ unknowns, namely, $\mathbf{z}^\star = \begin{bmatrix} \boldsymbol{\nu}^\star \\ \mathbf{u}^\star \end{bmatrix}$, that is, the identity of the active constraints ($\boldsymbol{\nu}^\star$) and the optimal inputs $\mathbf{u}^\star$. The quantities in the KKT conditions, for example $\mathbf{G}$, $\frac{\partial \Phi}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{G}}{\partial \mathbf{u}}$ in Eq. (3), $\mathbf{y}$ and $\frac{\partial \mathbf{y}}{\partial \mathbf{u}}$ in Eq. (4), or $\mathbf{G}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{u}}$ in Eq. (6), are called KKT elements, the number of which varies with the formulation of the KKT conditions. For example, the standard formulation (3) contains $n_K := n_g + n_u(n_g + 1)$ KKT elements, the output formulation (4) contains $n_{K_y} := n_y(n_u + 1)$ elements, while the Lagrangian formulation (6) contains $n_g + n_u$ KKT elements.

We can group these elements in a so-called KKT vector as follows:

$$\begin{aligned}\mathcal{K}(\mathbf{u}, \boldsymbol{\theta}) := \bigg[\mathbf{G}^\mathsf{T}(\mathbf{u}, \boldsymbol{\theta}), \ \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}), \frac{\partial G_1}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}), \\ \dots, \frac{\partial G_{n_g}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) \bigg]^\mathsf{T} \in \mathbb{R}^{n_K}\end{aligned} \tag{7}$$

[1] The partial derivative of a scalar with respect to a vector is considered as a row vector throughout the paper.

Table 2
Alternative MA schemes based on the way the model is modified to match all (**schemes in bold**) or only a subset (*schemes in italics*) of the KKT conditions. The classification is organized according to whether or not the active set is known.

			Scheme	# Modifiers n_m available dof	# Active KKT n_a used dof	# Residual dof n_{res}
Plant information to be matched by the model	Active set unknown	**All KKT conditions**	**MA**	$n_g + n_u(n_g+1)$	$n_g + n_u$	$n_u n_g$
			MAy	$n_y(n_u+1)$	$n_g + n_u$	$n_y(n_u+1) - n_g - n_u$
			MA-$\mathcal{L}$	$n_g + n_u$	$n_g + n_u$	0
		All constraints and dominant gradients	*ADMA*	$n_g + n_u^d(n_g+1)$	$n_g + n_u^d$	$n_u^d n_g$
			ADMAy	$n_y(n_u^d+1)$	$n_g + n_u^d$	$n_y(n_u^d+1) - n_g - n_u^d$
			ADMA-$\mathcal{L}$	$n_g + n_u^d$	$n_g + n_u^d$	0
		All constraints	*CA*	n_g	n_g	0
			CAy	n_y	n_g	$n_y - n_g$
	Active set known	**All active KKT conditions** (active constraints and reduced gradients)	**MA-kAS**	$n_g + n_r(n_g+1)$	$n_g^a + n_r$	$n_r n_g + n_g - n_g^a$
			MAy-kAS	$n_y(n_r+1)$	$n_g^a + n_r$	$n_y(n_r+1) - n_g^a - n_r$
			MA-$\mathcal{L}$-kAS	$n_g + n_r$	$n_g^a + n_r$	$n_g - n_g^a$
		Active constraints and dominant reduced gradients	*ADMA-kAS*	$n_g + n_r^d(n_g+1)$	$n_g^a + n_r^d$	$n_r^d n_g + n_g - n_g^a$
			ADMAy-kAS	$n_y(n_r^d+1)$	$n_g^a + n_r^d$	$n_y(n_r^d+1) - n_g^a - n_r^d$
			ADMA-$\mathcal{L}$-kAS	$n_g + n_r^d$	$n_g^a + n_r^d$	$n_g - n_g^a$
		Active constraints	*CA-kAS*	n_g	n_g^a	$n_g - n_g^a$
			CAy-kAS	n_y	n_g^a	$n_y - n_g^a$

The first class includes the standard formulation (MA), the output formulation (MAy), the Lagrangian formulation (MA-$\mathcal{L}$), directional schemes (active directional modifier adaptation ADMA, its output formulation ADMAy and its Lagrangian formulation ADMA-$\mathcal{L}$), and constraint-based schemes (constraint adaptation CA and its output formulation CAy). The second class encompasses the corresponding schemes that can be used when the active set is known. The number of residual dof is the difference between the number of modifiers (or available dof) and the number of active KKT conditions (or used dof). Dimensions: n_u is the number of inputs, n_y the number of outputs, n_g the number of constraints, n_g^a the number of active constraints, n_u^d the number of dominant input directions, $n_r = n_u - n_g^a$ the number of reduced input directions, n_r^d the number of dominant reduced input directions, n_m the number of modifiers, n_a the number of *active* KKT conditions, and n_{res} the number of residual degrees of freedom.

for the standard formulation (3),

$$\mathcal{K}^y(\mathbf{u},\boldsymbol{\theta}) := \left[\mathbf{y}^\mathsf{T}(\mathbf{u},\boldsymbol{\theta}),\ \frac{\partial y_1}{\partial \mathbf{u}}(\mathbf{u},\boldsymbol{\theta}),\ \ldots,\ \frac{\partial y_{n_y}}{\partial \mathbf{u}}(\mathbf{u},\boldsymbol{\theta})\right]^\mathsf{T} \in \mathbb{R}^{n_{K_y}} \tag{8}$$

for the output formulation (4), or

$$\mathcal{K}^{\mathcal{L}}(\mathbf{z},\boldsymbol{\theta}) := \left[\mathbf{G}^\mathsf{T}(\mathbf{u},\boldsymbol{\theta}),\ \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{z},\boldsymbol{\theta})\right]^\mathsf{T} \in \mathbb{R}^{n_g+n_u} \tag{9}$$

for the Lagrangian formulation (6).

2.3. Corresponding modifier-adaptation schemes

This section briefly reviews the schemes MA, MAy and MA-$\mathcal{L}$ associated with the three KKT formulations presented in the previous section. These schemes are listed in Table 2, which also contains extensions that will be introduced in Section 3.

2.3.1. Standard formulation – MA

Modifier adaptation introduces input-affine corrections to the cost and constraint functions of the nominal model. At the kth RTO iteration, the inputs $\mathbf{u}_k$ are applied to the plant, and the modifiers are computed as follows:

$$\boldsymbol{\varepsilon}_k^{\mathbf{G}} := \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}) \tag{10a}$$

$$(\boldsymbol{\lambda}_k^{\Phi})^\mathsf{T} := \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta}) \tag{10b}$$

$$(\boldsymbol{\lambda}_k^{\mathbf{G}})^\mathsf{T} := \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta}). \tag{10c}$$

Here, $\boldsymbol{\varepsilon}_k^{\mathbf{G}} \in \mathbb{R}^{n_g}$ are zeroth-order modifiers for the constraints, and $\boldsymbol{\lambda}_k^{\Phi} \in \mathbb{R}^{n_u}$ and $\boldsymbol{\lambda}_k^{\mathbf{G}} \in \mathbb{R}^{n_u \times n_g}$ are first-order modifiers for the cost and constraint gradients, respectively. The next inputs $\mathbf{u}_{k+1}$ are computed by solving the following modified optimization problem:

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \quad \Phi_m(\mathbf{u},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + (\boldsymbol{\lambda}_k^{\Phi})^\mathsf{T}\mathbf{u} \tag{11a}$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u},\boldsymbol{\theta}) := \mathbf{G}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathbf{G}} + (\boldsymbol{\lambda}_k^{\mathbf{G}})^\mathsf{T}(\mathbf{u}-\mathbf{u}_k) \le \mathbf{0}. \tag{11b}$$

The major advantage of MA is that it guarantees meeting the plant KKT conditions upon convergence. The satisfaction of the KKT conditions for the plant optimization Problem (1), upon convergence of standard MA, is proven in Marchetti et al. (2009). The reader is referred to the overview by Marchetti et al. (2016) for more details on MA, including the use of filtering.

2.3.2. Output formulation – MAy

In this formulation, input-affine corrections are introduced to the output functions. At the kth RTO iteration, the inputs $\mathbf{u}_k$ are applied to the plant, and the modifiers are computed as:

$$\boldsymbol{\varepsilon}_k^{\mathbf{y}} := \mathbf{y}_p(\mathbf{u}_k) - \mathbf{y}(\mathbf{u}_k,\boldsymbol{\theta}) \tag{12a}$$

$$(\boldsymbol{\lambda}_k^{\mathbf{y}})^\mathsf{T} := \frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta}). \tag{12b}$$

Here, $\boldsymbol{\varepsilon}_k^{\mathbf{y}} \in \mathbb{R}^{n_y}$ are zeroth-order modifiers for the outputs, and $\boldsymbol{\lambda}_k^{\mathbf{y}} \in \mathbb{R}^{n_u \times n_y}$ are first-order modifiers for the output gradients. The next inputs $\mathbf{u}_{k+1}$ are computed by solving the following modified optimization problem:

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \ \Phi_m(\mathbf{u},\boldsymbol{\theta}) := \phi\big(\mathbf{u}, \mathbf{y}_m(\mathbf{u},\boldsymbol{\theta})\big) \tag{13a}$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u},\boldsymbol{\theta}) := \mathbf{g}\big(\mathbf{u}, \mathbf{y}_m(\mathbf{u},\boldsymbol{\theta})\big) \le \mathbf{0}, \tag{13b}$$

where

$$\mathbf{y}_m(\mathbf{u},\boldsymbol{\theta}) := \mathbf{y}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathbf{y}} + (\boldsymbol{\lambda}_k^{\mathbf{y}})^\mathsf{T}(\mathbf{u}-\mathbf{u}_k). \tag{14}$$

Satisfaction of the plant KKT conditions upon convergence is proven in Papasavvas et al. (2019).

2.3.3. Lagrangian formulation – MA-$\mathcal{L}$

With this formulation, input-affine corrections are introduced to the Lagrangian function (Marchetti et al., 2016). At the kth RTO iteration, the inputs $\mathbf{u}_k$ are applied to the plant, and the modifiers are computed as:

$$\boldsymbol{\varepsilon}_k^{\mathbf{G}} := \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}) \tag{15a}$$

$$(\boldsymbol{\lambda}_k^{\mathcal{L}})^\mathsf{T} := \frac{\partial \mathcal{L}_p}{\partial \mathbf{u}}(\mathbf{z}_k) - \frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{z}_k,\boldsymbol{\theta}). \tag{15b}$$

Here, $\boldsymbol{\varepsilon}_k^{\mathbf{G}} \in \mathbb{R}^{n_g}$ are zeroth-order modifiers for the constraints, and $\boldsymbol{\lambda}_k^{\mathcal{L}} \in \mathbb{R}^{n_u}$ are first-order modifiers for the Lagrangian gradient. The next inputs $\mathbf{u}_{k+1}$ and the Lagrange multipliers $\boldsymbol{\nu}_{k+1}$ are computed by solving the following modified optimization problem:

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \quad \Phi_m(\mathbf{u},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + (\boldsymbol{\lambda}_k^{\mathcal{L}})^{\mathrm{T}}\mathbf{u} \tag{16a}$$

$$\text{s.t.} \quad \mathbf{G}_m(\mathbf{u},\boldsymbol{\theta}) := \mathbf{G}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathbf{G}} \leq \mathbf{0}. \tag{16b}$$

The fact that MA-$\mathcal{L}$ reaches plant optimality is detailed in Appendix B, which also illustrates the difference in convergence path between MA and MA-$\mathcal{L}$.

3. Efficient directional modifier adaptation

In modifier adaptation, the model modification includes two aspects, namely, a local correction and an iterative process:

- Locally, at the current operating point $\mathbf{u}_k$, one can correct the model so that its constrained values and its cost and constraint gradients match those of the plant, the plant values being measured or estimated from measurements.
- Since the model corrections are made locally at $\mathbf{u}_k$, and not at the (unknown) plant optimum $\mathbf{u}_p^\star$, one needs to iterate to reach plant optimality.

When directional information is available, the number of input directions along which gradient elements need to be estimated can be reduced. Directional information is typically available when the active set or the dominant gradients are known. These two cases are discussed next.

3.1. Knowledge of active set

Knowledge of the active set leads to a reduced set of KKT conditions, and in particular to fewer gradients, as shown next.

3.1.1. Reduced set of KKT conditions

Since $\nu_j^\star = 0$ for each inactive constraint $G_j(\mathbf{u}^\star,\boldsymbol{\theta}) < 0$, one can consider only the n_g^a active constraints in Eq. (3) to obtain

$$\frac{\partial \Phi}{\partial \mathbf{u}}\left(\mathbf{u}^\star,\boldsymbol{\theta}\right) + \sum_{j=1}^{n_g^a} \nu_j^{a\star} \frac{\partial G_j^a}{\partial \mathbf{u}}\left(\mathbf{u}^\star,\boldsymbol{\theta}\right) = \mathbf{0} \tag{17a}$$

$$G_j^a(\mathbf{u}^\star,\boldsymbol{\theta}) = 0, \qquad \forall j \in \{1,\ldots,n_g^a\} \tag{17b}$$

$$\nu_j^{a\star} \geq 0, \qquad \forall j \in \left\{1,\ldots,n_g^a\right\}, \tag{17c}$$

where the superscript $(\cdot)^a$ relates to the active constraints.

Eq. (17a) shows that the cost gradient must equal a linear combination of the gradients of the active constraints. In other words, the projection of the cost gradient on the space tangent to the active constraints (called the reduced cost gradient) must vanish. This space is obtained by writing

$$\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}^\star,\boldsymbol{\theta})\,\mathbf{N}^\star = \mathbf{0}, \tag{18}$$

where $\mathbf{N}^\star \in \mathbb{R}^{n_u \times n_r}$ is a null-space matrix with orthonormal columns that satisfies Eq. (18) and $n_r = n_u - n_g^a$. The n_r columns of $\mathbf{N}^\star$ represent a basis of the space that is orthogonal to the gradients of the active constraints.[2] Post-multiplying Eq. (17a) by $\mathbf{N}^\star$ and using Eq. (18) gives:

$$\nabla_r \Phi(\mathbf{u}^\star,\boldsymbol{\theta}) := \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}^\star,\boldsymbol{\theta})\,\mathbf{N}^\star = \mathbf{0}, \tag{19}$$

where $\nabla_r \Phi(\mathbf{u}^\star,\boldsymbol{\theta}) \in \mathbb{R}^{1\times n_r}$ is the reduced cost gradient of optimization Problem (2).[3]

Remark 1. With $\mathcal{L}^a(\mathbf{z},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + (\boldsymbol{\nu}^a)^{\mathrm{T}}\mathbf{G}^a(\mathbf{u},\boldsymbol{\theta})$, Eq. (17a) can be written

$$\frac{\partial \mathcal{L}^a}{\partial \mathbf{u}}\left(\mathbf{z}^\star,\boldsymbol{\theta}\right) = \frac{\partial \Phi}{\partial \mathbf{u}}\left(\mathbf{u}^\star,\boldsymbol{\theta}\right) + (\boldsymbol{\nu}^{a\star})^T \frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}\left(\mathbf{u}^\star,\boldsymbol{\theta}\right) = \mathbf{0}, \tag{20}$$

which, upon post-multiplying by $\mathbf{N}^\star$ and with the definition of the reduced Lagrangian gradient $\nabla_r \mathcal{L}^a(\mathbf{z}^\star,\boldsymbol{\theta}) := \frac{\partial \mathcal{L}^a}{\partial \mathbf{u}}(\mathbf{z}^\star,\boldsymbol{\theta})\,\mathbf{N}^\star$, gives:

$$\nabla_r \mathcal{L}^a(\mathbf{z}^\star,\boldsymbol{\theta}) = \nabla_r \Phi(\mathbf{u}^\star,\boldsymbol{\theta}) = \mathbf{0}. \tag{21}$$

This shows that, with this reduced set of KKT conditions, the reduced cost gradient is also the reduced Lagrangian gradient.

In summary, if the set of active constraints is known, the KKT conditions correspond to the following $n_g^a + n_r$ equalities related to the active constraints and reduced cost gradients:

$$\mathbf{G}^a(\mathbf{u}^\star,\boldsymbol{\theta}) = \mathbf{0} \tag{22a}$$

$$\nabla_r \Phi(\mathbf{u}^\star,\boldsymbol{\theta}) = \mathbf{0}. \tag{22b}$$

Note that this reduced set of KKT conditions does not involve any Lagrange multipliers. More details about this reduced set of KKT conditions can be found in Marchetti et al. (2020).

KKT elements. In this formulation, the KKT elements can be grouped in the KKT vector

$$\mathcal{K}^a(\mathbf{u},\boldsymbol{\theta}) := \left[\mathbf{G}^{a\mathrm{T}}(\mathbf{u},\boldsymbol{\theta}), \quad \nabla_r \Phi(\mathbf{u},\boldsymbol{\theta})\right]^{\mathrm{T}} \in \mathbb{R}^{n_u}, \tag{23}$$

with $n_u = n_g^a + n_r$.

3.1.2. Known-active-set schemes (X-kAS schemes)

If the active set at the plant optimum is known, there are fewer (in fact $n_u = n_g^a + n_r$) KKT elements owing to the use of reduced gradients as per Section 3.1.1. However, due to plant-model mismatch, it will be necessary to use additional degrees of freedom to ensure both feasibility prior to convergence and optimality upon convergence. We will illustrate the known-active-set formulation via the MA-kAS scheme, although it is also possible to formulate other X-kAS schemes. Note that all the X-kAS schemes are novel and do not exist in the published literature.

The reduced cost and constraint gradients at $\mathbf{u}_k$ read:

$$\nabla_r \Phi(\mathbf{u}_k,\boldsymbol{\theta}) = \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta})\,\mathbf{N}_k, \tag{24a}$$

$$\nabla_r \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}) = \frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta})\,\mathbf{N}_k, \tag{24b}$$

where $\mathbf{N}_k \in \mathbb{R}^{n_u \times n_r}$ is a null-space matrix with orthonormal columns that satisfies

$$\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta})\,\mathbf{N}_k = \mathbf{0}. \tag{25}$$

The use of reduced gradients leads to fewer modifiers, namely:

$$\boldsymbol{\varepsilon}_k^{\mathbf{G}} := \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}), \tag{26a}$$

$$(\boldsymbol{\lambda}_k^{\Phi,r})^{\mathrm{T}} := \nabla_r \Phi_p(\mathbf{u}_k) - \nabla_r \Phi(\mathbf{u}_k,\boldsymbol{\theta}), \tag{26b}$$

$$(\boldsymbol{\lambda}_k^{\mathbf{G},r})^{\mathrm{T}} := \nabla_r \mathbf{G}_p(\mathbf{u}_k) - \nabla_r \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}), \tag{26c}$$

where $\boldsymbol{\lambda}_k^{\Phi,r} \in \mathbb{R}^{n_r}$ and $\boldsymbol{\lambda}_k^{\mathbf{G},r} \in \mathbb{R}^{n_r \times n_g}$ are the first-order modifiers for the reduced cost and constraint gradients, respectively. The reduced cost and constraint gradients of the plant, $\nabla_r \Phi_p(\mathbf{u}_k) = \frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k)\,\mathbf{N}_k$ and $\nabla_r \mathbf{G}_p(\mathbf{u}_k) = \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}(\mathbf{u}_k)\,\mathbf{N}_k$, can be estimated via

[2] $\mathbf{N}^\star$ can be computed as the n_r right singular vectors of $\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}$ that correspond to vanishing singular values.

[3] The reduced cost gradient represents the cost variation resulting from input variations in the space tangent to the active constraints.

finite-difference approximation by perturbing the inputs in the n_r-dimensional subspace at each RTO point $\mathbf{u}_k$.

Once the modifiers (26) have been evaluated, the next inputs $\mathbf{u}_{k+1}$ are computed by solving the following modified optimization problem:

$$\mathbf{u}_{k+1} = \arg\min_{\mathbf{u}} \Phi_m(\mathbf{u},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + \left(\boldsymbol{\lambda}_k^{\Phi,r}\right)^{\mathsf{T}} \mathbf{N}_k^{\mathsf{T}}\,\mathbf{u} \tag{27a}$$

$$\text{s.t.}\quad \mathbf{G}_m^a(\mathbf{u},\boldsymbol{\theta}) := \mathbf{G}^a(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathbf{G}^a} + \left(\boldsymbol{\lambda}_k^{\mathbf{G}^a,r}\right)^{\mathsf{T}} \mathbf{N}_k^{\mathsf{T}}\,(\mathbf{u}-\mathbf{u}_k) = \mathbf{0}, \tag{27b}$$

$$\mathbf{G}_m^i(\mathbf{u},\boldsymbol{\theta}) := \mathbf{G}^i(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathbf{G}^i} + \left(\boldsymbol{\lambda}_k^{\mathbf{G}^i,r}\right)^{\mathsf{T}} \mathbf{N}_k^{\mathsf{T}}\,(\mathbf{u}-\mathbf{u}_k) \le \mathbf{0}, \tag{27c}$$

where the superscripts $(\cdot)^{\mathbf{G}^a}$ and $(\cdot)^{\mathbf{G}^i}$ denote the active and inactive constraints, respectively. Note that the active inequality constraints have been replaced by equality constraints in Problem (27), while the inactive inequality constraints are kept to avoid constraint violations prior to convergence.

MA-kAS can reach plant optimality upon convergence if $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_\infty)\mathbf{N}_\infty = \mathbf{0}$ as detailed in Appendix C. This condition can be met in two different ways:

(i) If $\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta}) = \frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_k)$, that is, when there is no plant-model mismatch associated with the gradients of the active constraints, for instance, when the active constraints include only input-dependent constraints such as input bounds. In this case, the converged condition $\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_\infty,\boldsymbol{\theta})\mathbf{N}_\infty = \mathbf{0}$ obviously implies $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_\infty)\mathbf{N}_\infty = \mathbf{0}$.

(ii) If $\mathbf{N}_k$ is chosen to satisfy $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_k^h)\,\mathbf{N}_k = \mathbf{0}$, with $\mathbf{u}_k^h$ being the value of $\mathbf{u}_k$ available every $h \in \mathbb{N}$ iterations. This alternative choice of $\mathbf{N}_k$, which also leads to $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_\infty)\mathbf{N}_\infty = \mathbf{0}$, is experimentally more expensive than the choice of $\mathbf{N}_k$ in Eq. (25) as it involves estimating the gradients of the active plant constraints in all n_u directions at every h iterations. The computation of the modifiers $\boldsymbol{\lambda}_k^{\Phi,r}$ and $\boldsymbol{\lambda}_k^{\mathbf{G},r}$ in (26b) and (26c) requires estimating the gradients of the plant cost and constraints in n_r directions at each iteration regardless of the choice of $\mathbf{N}_k$. Thus, the choice of $\mathbf{N}_k$ in this Case (ii) requires $n_r + \frac{n_u - n_r}{h} + 1$ experimental runs on average at each iteration, which is more than $n_r + 1$ with the choice of $\mathbf{N}_k$ in Eq. (25) but less than $n_u + 1$ experimental runs for standard MA.

In practice, one often prefers the choice of $\mathbf{N}_k$ in Eq. (25), that is, the reduced input directions $\mathbf{N}_k$ are chosen orthogonal to the gradients of the active *model* constraints, since they can be computed at a lower experimental cost than the choice of $\mathbf{N}_k$ in Case (ii). For this reason, the choice of $\mathbf{N}_k$ in Eq. (25) is used in the remainder. Note, however, that this choice results in a (hopefully small) loss of optimality upon convergence when the condition associated with Case (i) does not apply.

Remark 2. The proposed MA-kAS formulation is in fact a directional MA approach, wherein, at each RTO iteration, the cost and constraint gradients are corrected along the n_r input directions $\mathbf{u}_r$ given by the columns of $\mathbf{N}_k$, that is, $\mathbf{u}_r := \mathbf{N}_k^{\mathsf{T}}\,\mathbf{u}$.

Remark 3. Problem (27) can be transformed into an output formulation similar to (13) or a Lagrangian formulation similar to (16), which results in MA schemes labeled MAy-kAS and MA-$\mathcal{L}$-kAS. As in the case of MA-kAS, the only changes in MAy-kAS and MA-$\mathcal{L}$-kAS with respect to MAy and MA-$\mathcal{L}$ are the replacement of the active inequality constraints by equality constraints and the use of reduced gradients instead of full gradients. It can be shown that MAy-kAS and MA-$\mathcal{L}$-kAS can reach plant optimality by following a reasoning similar to the one in Appendix C.

Remark 4. The first-order cost modifiers (26b) can be written as:

$$(\boldsymbol{\lambda}_k^{\Phi,r})^{\mathsf{T}} = \left(\frac{\partial \Phi_p}{\partial \mathbf{u}}(\mathbf{u}_k) - \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta})\right)\mathbf{N}_k = (\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}}\mathbf{N}_k, \tag{28}$$

which allows writing the modified cost in Eq. (27a) as:

$$\Phi_m(\mathbf{u},\boldsymbol{\theta}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + (\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}}\mathbf{N}_k\mathbf{N}_k^{\mathsf{T}}\,\mathbf{u}. \tag{29}$$

Since $\mathbf{N}_k\mathbf{N}_k^{\mathsf{T}} \in \mathbb{R}^{n_u \times n_u}$ is a projection matrix of rank n_r, Eq. (29) shows that, compared to full gradient modifiers, the first-order cost modifiers in MA-kAS correct the cost gradient only in the n_r reduced input directions $\mathbf{u}_r := \mathbf{N}_k^{\mathsf{T}}\,\mathbf{u}$. The same remark is valid for the first-order constraint modifiers $\boldsymbol{\lambda}_k^{\mathbf{G},r}$ in (26c). This has the advantage that the modifiers $\boldsymbol{\lambda}_k^{\Phi,r}$ and $\boldsymbol{\lambda}_k^{\mathbf{G},r}$ only need to be estimated in the reduced input directions. These reduced input directions are in fact the sensitivity-seeking directions introduced by François et al. (2005).

Implementation aspects (MA-kAS). We propose an algorithm that strives for KKT matching when the active set is known. The null-space matrix $\mathbf{N}_k$ that satisfies $\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_k,\boldsymbol{\theta})\,\mathbf{N}_k = \mathbf{0}$ is computed online at each RTO iteration. The reduced plant gradients at $\mathbf{u}_k$ are evaluated as:

$$\nabla_r \Phi_p(\mathbf{u}_k) := \left.\frac{\partial \Phi_p(\mathbf{u}_k + \mathbf{N}_k\,\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}, \tag{30a}$$

$$\nabla_r \mathbf{G}_p(\mathbf{u}_k) := \left.\frac{\partial \mathbf{G}_p(\mathbf{u}_k + \mathbf{N}_k\,\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}, \tag{30b}$$

with the perturbation vector $\mathbf{p} \in \mathbb{R}^{n_r}$. More details about the selection of an appropriate step size for $\mathbf{p}$ can be found in Costello et al. (2016). The MA-kAS scheme given in Algorithm 1 is novel, albeit based on the MA algorithm proposed by Marchetti et al. (2009) (see also Papasavvas et al., 2019; Marchetti et al., 2016). Note that it is possible to write similar versions for MAy-kAS and MA-$\mathcal{L}$-kAS, as mentioned in Remark 3.

3.2. Computation of dominant gradients

Here, we challenge the idea that all KKT conditions must be matched for "good performance". Indeed, it suffices to match the active constraints and the dominant elements of the cost and constraint gradients. However, the active constraints are often unknown, and the dominant gradients need to be determined, for which a few practical considerations are given next:

- For safety and quality issues, satisfying process constraints is key to process performance. Hence, it is essential to match at least the n_g^a active constraints. When the active set is unknown, it is proposed to match the values of all n_g constraints using bias terms. This will ensure that the (unknown) active constraints will be matched. Note that, if the constrained values can be measured, this step of matching the plant constraints is rather straightforward.
- In contrast, the estimation of plant gradients is often time consuming and thus experimentally expensive. Hence, one might be interested in matching these gradients only in the dominant input directions rather than in all input directions, which requires the estimation of fewer plant derivatives. To determine the n_u^d dominant input directions, we propose to perform a sensitivity analysis using the Lagrangian formulation. The active subspace approach proposed by Singhal et al. (2018) can be applied to the Lagrangian gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{z},\boldsymbol{\theta})$. Concretely, the approach described in Appendix A.3 can be applied, with the vector function $\mathbf{f}(\boldsymbol{\theta})$ in Algorithm 4 chosen as $\frac{\partial \mathcal{L}}{\partial \mathbf{u}}(\mathbf{z},\boldsymbol{\theta})$. The

Algorithm 1 Modifier adaptation with known active set (MA-kAS).

Step 0 (Initialization): Select the filter matrix $\mathbf{K} \in \mathbb{R}^{n_u \times n_u}$ (typically a diagonal matrix) with eigenvalues in the interval (0,1], and choose the feasible starting point $\mathbf{u}_0$.
for $k = 0 \rightarrow \infty$
Step 1 (Plant evaluation): Apply $\mathbf{u}_k$ to the plant, wait for steady state, collect the measurements $\mathbf{y}_p(\mathbf{u}_k)$, and evaluate $\Phi_p(\mathbf{u}_k)$ and $\mathbf{G}_p(\mathbf{u}_k)$.
Step 2 (Estimation of directional derivatives):
Use the model to compute the $(n_u \times n_r)$-dimensional null-space matrix with orthonormal columns $\mathbf{N}_k$ that satisfies $\frac{\partial \mathbf{G}^a}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta})\,\mathbf{N}_k = \mathbf{0}$, and estimate the plant directional derivatives as per Eq. (30).
Step 3 (Computation of modifiers):
Compute the modifiers at $\mathbf{u}_k$ as per Eq. (26).
Step 4 (Computation of next inputs): Solve the modified optimization Problem (27) to compute the optimal inputs $\mathbf{u}^\star_{k+1}$ and generate the filtered inputs

$$\mathbf{u}_{k+1} = (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}^\star_{k+1}. \tag{31}$$

end

Algorithm 2 Output active directional modifier adaptation (ADMAy).

Step 0 (Initialization): Use the model to compute the $(n_u \times n_u^d)$-dimensional matrix with orthonormal columns $\hat{\mathbf{W}}_1$ of dominant input directions as per Algorithm 4 in Appendix A.3.
Select the filter matrix $\mathbf{K} \in \mathbb{R}^{n_u \times n_u}$ (typically a diagonal matrix) with eigenvalues in the interval (0,1], and choose the feasible starting point $\mathbf{u}_0$.
for $k = 0 \rightarrow \infty$
Step 1 (Plant evaluation): Apply $\mathbf{u}_k$ to the plant, wait for steady state and collect the measurements $\mathbf{y}_p(\mathbf{u}_k)$.
Step 2 (Estimation of directional derivatives): Estimate the plant directional derivatives as per Eq. (32).
Step 3 (Computation of modifiers): At $\mathbf{u}_k$, estimate the full output gradients as:

$$\frac{\partial \mathbf{y}_p}{\partial \mathbf{u}}(\mathbf{u}_k) = \nabla_{\hat{W}_1}\mathbf{y}_p(\mathbf{u}_k)\,\hat{\mathbf{W}}_1^\top + \frac{\partial \mathbf{y}}{\partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\theta})\left(\mathbf{I}_{n_u} - \hat{\mathbf{W}}_1\hat{\mathbf{W}}_1^\top\right), \tag{33}$$

and compute the modifiers as per Eq. (12).
Step 4 (Computation of next inputs): Solve the modified optimization Problem (13) to compute the optimal inputs $\mathbf{u}^\star_{k+1}$ and generate the filtered inputs

$$\mathbf{u}_{k+1} = (\mathbf{I} - \mathbf{K})\mathbf{u}_k + \mathbf{K}\mathbf{u}^\star_{k+1}. \tag{34}$$

end

dominant input directions are computed as the $(n_u \times n_u^d)$-dimensional matrix $\hat{\mathbf{W}}_1$ that represents the n_u^d dominant eigenvectors of the matrix $\hat{\mathbf{G}}$ in Appendix A.3.

Hence, the "dominant" KKT conditions to be matched include n_g constraints as well as the cost and constraint gradients in n_u^d dominant input directions.

3.2.1. Active directional schemes (ADMA schemes)

All aforementioned schemes are capable of reaching plant optimality if the plant gradients are accurate. However, the experimental cost to estimate these gradients can be very high, in particular when there are many inputs since the number of gradient elements to be estimated is proportional to n_u (more precisely, this number is $n_u(n_g + 1)$ for MA, $n_y n_u$ for MAy, n_u for MA-$\mathcal{L}$, $n_r(n_g + 1)$ for MA-kAS, $n_y n_r$ for MAy-kAS, and n_r for MA-$\mathcal{L}$-kAS). Hence, it is of considerable practical interest to be able to reduce the number of gradient elements to estimate by considering only the dominant input directions. Using the concept of dominant input directions, the schemes MA, MAy, MA-$\mathcal{L}$, MA-kAS, MAy-kAS, and MA-$\mathcal{L}$-kAS can be turned into corresponding ADMA schemes. These formulations, together with some of their features, are listed in Table 2.

Implementation aspects (ADMAy). For the case of unknown active set, we propose to implement partial KKT matching, that is, the gradient-zeroing conditions are matched only along dominant input directions. These n_u^d directions are computed offline as the $(n_u \times n_u^d)$-dimensional matrix $\hat{\mathbf{W}}_1$ (see Algorithm 4 in Appendix A.3). Using the output formulation, the directional derivatives of the outputs at $\mathbf{u}_k$ in the n_u^d dominant directions are:

$$\nabla_{\hat{W}_1}\mathbf{y}_p(\mathbf{u}_k) := \left.\frac{\partial \mathbf{y}_p(\mathbf{u}_k + \hat{\mathbf{W}}_1\,\mathbf{p})}{\partial \mathbf{p}}\right|_{\mathbf{p}=\mathbf{0}}, \tag{32}$$

with the perturbation vector $\mathbf{p} \in \mathbb{R}^{n_u^d}$. More details about the selection of an appropriate step size for $\mathbf{p}$ can be found in Costello et al. (2016). The ADMAy scheme given in Algorithm 2 is novel, albeit based on the ADMA algorithm proposed by Singhal et al. (2018). Note that it is possible to write similar versions for ADMA-$\mathcal{L}$ and the known-active-set formulations ADMA-kAS, ADMAy-kAS, and ADMA-$\mathcal{L}$-kAS, as detailed in the next remark.

Remark 5. It is also possible to combine the last two schemes, that is, to reduce the n_r input directions to n_r^d directions, which would correspond to ADMAy-kAS in Table 2. This is of particular interest when $n_r^d \ll n_r$. Such a scheme can be obtained by simply projecting the columns of the matrix $\hat{\mathbf{W}}_1$ for the ADMAy scheme along the directions in the columns of the matrix $\mathbf{N}_k$ for the MA-kAS scheme, which results in the matrix $\hat{\mathbf{W}}_{1,k} = \mathrm{orth}\left(\mathbf{N}_k\mathbf{N}_k^\top\hat{\mathbf{W}}_1\right)$ for the ADMAy-kAS scheme, where orth returns a matrix with orthonormal columns that span the same column space.

4. Simulation study

4.1. Problem statement

The proposed methodology is applied to the process shown in Fig. 1, which is a modification of the Williams–Otto plant proposed by Williams and Otto (1960). Hence, the WO plant used in this section is adapted from Williams and Otto (1960), but all numerical values are not directly available in the original publication, which prompted us to define the model in some unambiguous way. In this section, and in Appendix D, we recall the problem formulation proposed by Singhal (2018). This formulation includes not only the Williams–Otto reactor that has been used as a simulation testbed

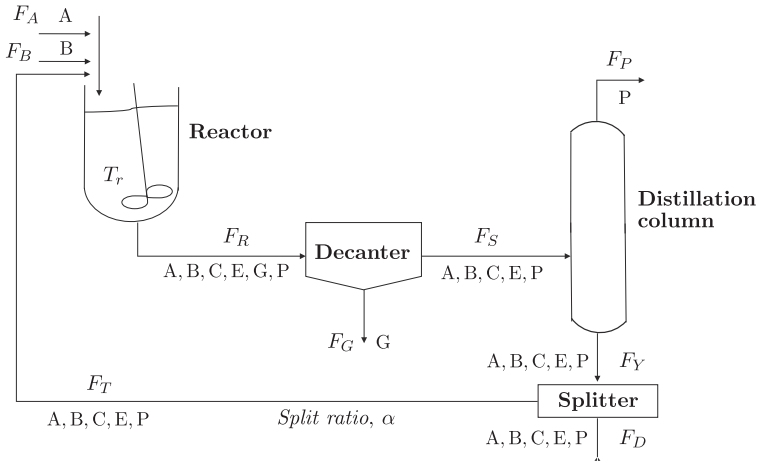

Fig. 1. Willliams–Otto process with a reactor, decanter, distillation column, splitter and recycle stream.

for several RTO methods in previous studies (Navia et al., 2013; Gao et al., 2015; Marchetti et al., 2016) but also other process units that make its structural plant-model mismatch more challenging. The WO reactor has two inputs, whereas the WO plant considered here has four inputs, which is a much more challenging process to optimize. Furthermore, as shown in this section, the WO plant exhibits a fair amount of directionality, whereas the WO reactor does not. In summary, our example describes each unit operation in a simpler and clearer way than the original WO plant, which facilitates its understanding and implementation by other researchers, while it remains a more challenging problem than the WO reactor.

4.1.1. Process description

In the process shown in Fig. 1, F_A and F_B are the fresh feeds of Species A and B, while F_i, $i = R, S, G, D, P, T, Y$, are the flowrates of the various streams. The feeds F_A and F_B are mixed with the recycle stream F_T in a CSTR, where the following three reactions take place:

$$\mathrm{A} + \mathrm{B} \xrightarrow{\mathrm{k}_1} \mathrm{C}, \qquad \mathrm{C} + \mathrm{B} \xrightarrow{\mathrm{k}_2} \mathrm{P} + \mathrm{E}, \qquad \mathrm{P} + \mathrm{C} \xrightarrow{\mathrm{k}_3} \mathrm{G}, \tag{35a}$$

$$\mathrm{k}_i = \mathrm{A}_i \exp\left(\frac{-\mathrm{E}_i}{T_r + 273.15}\right), \quad i = 1, 2, 3. \tag{35b}$$

Here, C is an intermediate, P is the main product, E is a side product, and G is an oily waste product. The side product E can be sold for its fuel value, while G must be disposed of at a cost. The decanter completely removes Species G from the reactor outlet stream into the waste product stream F_G. The decanter outlet stream is sent to a distillation unit that removes the product P at the top. Due to the formation of an azeotrope between P and E, some of the product P (in fact, the mass fraction β of the amount of E in the column feed) is retained in the bottoms. The fraction α of this bottom product is recycled to the reactor, while the rest is used as fuel. The reactor is simplified by assuming isothermal operation. The other units are also simplified to keep the example small and illustrate the proposed concepts with lesser complexity. As a result, the process is modeled without an energy balance. The material balance equations are given in Appendix D.

4.1.2. Available model

To introduce structural plant-model mismatch, the reaction system is modeled using only two reactions, thereby ignoring the intermediate C:

$$\mathrm{A} + 2\mathrm{B} \xrightarrow{\bar{\mathrm{k}}_1} \mathrm{P} + \mathrm{E}, \qquad \mathrm{A} + \mathrm{B} + \mathrm{P} \xrightarrow{\bar{\mathrm{k}}_2} \mathrm{G}, \tag{36a}$$

$$\bar{\mathrm{k}}_i = \bar{\mathrm{A}}_i \exp\left(\frac{-\bar{\mathrm{E}}_i}{T_r + 273.15}\right), \quad i = 1, 2. \tag{36b}$$

The corresponding model equations are given in Eq. (60). The five adjustable model parameters are the pre-exponential factors $\bar{\mathrm{A}}_1$ and $\bar{\mathrm{A}}_2$, the activation energies $\bar{\mathrm{E}}_1$ and $\bar{\mathrm{E}}_2$, and the bottoms fraction β. The parameter values and their uncertainty ranges are given in Table 4.

4.1.3. Formulation of the optimization problem

The objective is to maximize the return on investment (ROI) in terms of the net sales minus the costs for raw material, utility and waste disposal. It is also desired to keep the production rate of P below a threshold value because of limitation in downstream processing. The decision variables are the feed rates F_A and F_B, the reactor temperature T_r, and the split ratio α. These decision variables are bounded. The optimization problem can be formulated mathematically as:

$$\begin{aligned} \max_{F_A, F_B, T_r, \alpha} \ \mathrm{ROI} :&= 7358.4\,(P_P F_P + P_E F_D) \\ &\quad -8400\,(P_A F_A + P_B F_B + P_C F_G) - P_R F_R, \end{aligned} \tag{37a}$$

$$\text{s.t.} \quad F_P \ (\mathrm{kg/s}) \le 0.7, \tag{37b}$$

$$\begin{aligned} &1 \le F_A \ (\mathrm{kg/s}) \le 5, \quad 1 \le F_B \ (\mathrm{kg/s}) \le 4.5, \\ &70 \le T_r \ (^\circ\mathrm{C}) \ \le 100, \quad 0 \le \alpha \le 0.95. \end{aligned} \tag{37c}$$

The price values P_i, $i = A, B, E, G, P, R$, are given in Table 3. The optimal plant inputs are $\mathbf{u}_p^\star = [1.79,\ 4.04,\ 85.82,\ 0.89]^\mathsf{T}$ with a ROI value of 725.12 \$/s and the constraint on F_P active. Due to plant-model mismatch, the optimal inputs computed with the 2-reaction model are significantly different, namely, $\mathbf{u}^\star = [1.60,\ 3.52,\ 88.17,\ 0.88]^\mathsf{T}$, which gives a plant ROI of 639.04 \$/s. The objective is to combine the best available but *inaccurate* model with *noisy* output measurements to drive the plant cost from 639.04 to close to 725.12 \$/s, and this in as few iterations as possible.

4.2. Performance of selected directional MA schemes

Next, we present the application of selected RTO schemes to the Williams–Otto plant. It is assumed that the steady-state mass flow rates F_P and F_D are measured, from which all quantities in Problem (37) can be computed via simple mass balances. For all

Table 3
Williams–Otto plant: plant parameters and parameters that are common to both the plant and the model.

Plant parameter	Value	Common parameter	Value	Common parameter	Value
A_1 (s^{-1})	1.6599×10^6	V (kg)	2105	P_A (\$/kg)	0.0441
A_2 (s^{-1})	7.2117×10^8	M_A (kg/mol)	100	P_B (\$/kg)	0.0661
A_3 (s^{-1})	2.6745×10^{12}	M_B (kg/mol)	100	P_E (\$/kg)	0.015
E_1 (K)	6666.7	M_C (kg/mol)	200	P_G (\$/kg)	0.022
E_2 (K)	8333.3	M_P (kg/mol)	100	P_P (\$/kg)	0.6614
E_3 (K)	11111	M_E (kg/mol)	200	P_R (\$/kg)	4.8943
β (−)	0.1	M_G (kg/mol)	300	–	–

Table 4
Williams–Otto plant: model parameters.

Model parameter	Model parameter value $\boldsymbol{\theta}$	Uncertainty range	Probability distribution
$\bar{A}_1$ (s^{-1})	1.2661×10^8	$[8.5824 \times 10^7, 1.2874 \times 10^8]$	uniform
$\bar{A}_2$ (s^{-1})	1.2179×10^{13}	$[1.1324 \times 10^{13}, 1.6986 \times 10^{13}]$	uniform
$\bar{E}_1$ (K)	8014.0	[5966.88, 8950.32]	uniform
$\bar{E}_2$ (K)	12350.4	[11031.6, 16547.4]	uniform
β (-)	0.0882	[0.06, 0.09]	uniform

RTO schemes, two different conditions are tested: (i) no measurement noise, to assess the theoretical potential of each scheme; and (ii) Gaussian measurement noise with a standard deviation of 0.05% of the true output values. This is a plausible noise level by considering that the steady-state mass flow rates of the outlet streams P and D can be obtained by using redundant measurements of mass flow rates or precise measurements of the mass of P and D that is placed in storage tanks for further processing in a certain amount of time. A filter matrix $\mathbf{K}$ with all diagonal elements equal to 0.9 is used for all simulations without measurement noise, while diagonal elements equal to 0.5 are considered for all simulations with measurement noise. At the k^{th} RTO iteration, the plant gradients are estimated via finite-difference approximation from measurements at the nominal point $\mathbf{u}_k$ and neighboring points. This implies that the number of experimental runs at each iteration equals the number of neighboring points plus one for the nominal point. The step away from $\mathbf{u}_k$ to compute each gradient corresponds to 0.1% of the distance between the lower and upper bounds in the case of noise-free measurements, and to 2% of that distance in the presence of noisy measurements. The number of neighboring points equals the number of relevant input directions, namely, $n_u = 4$ for the basic formulations MA, MAy, and MA-$\mathcal{L}$; $n_r = 3$ for MA-kAS, MAy-kAS, MA-$\mathcal{L}$-kAS; $n_u^d = 2$ or $n_u^d = 1$ for ADMA, ADMAy and ADMA-$\mathcal{L}$; and $n_r^d = 2$ or $n_r^d = 1$ for ADMA-kAS, ADMAy-kAS, and ADMA-$\mathcal{L}$-kAS. The number of dominant input directions is chosen as 1 or 2 since the eigenvalues of $\hat{\mathbf{G}}$ are $\hat{\gamma}_1 = 1.4959 \times 10^{12}$, $\hat{\gamma}_2 = 0.3110 \times 10^{12}$, $\hat{\gamma}_3 = 0.0318 \times 10^{12}$, $\hat{\gamma}_4 = 0.0001 \times 10^{12}$, which means that $\hat{\gamma}_1$ is significantly larger than $\hat{\gamma}_2$, and $\hat{\gamma}_2$ is significantly larger than $\hat{\gamma}_3$. The dominant input directions are then the first column or the first two columns of $\hat{\mathbf{W}}_1$ in Appendix A.3, namely, $[-0.7275, 0.5728, -0.0259, 0.3767]^T$ and $[-0.6423, -0.7612, -0.0292, -0.0850]^T$.

One can consider 12 different MA schemes corresponding to the Cartesian product of the following options: MA or ADMA; unknown active set or known active set; standard formulation, output formulation, or Lagrangian formulation. Hence, the possible RTO schemes are the standard formulation MA, the output formulation MAy, and the Lagrangian formulation MA-$\mathcal{L}$, the known-active-set schemes MA-kAS, MAy-kAS, and MA-$\mathcal{L}$-kAS, the active directional schemes ADMA, ADMAy, and ADMA-$\mathcal{L}$, and the active directional, known-active-set formulations ADMA-kAS, ADMAy-kAS, and ADMA-$\mathcal{L}$-kAS. The results obtained using some of these schemes are discussed next.

Fig. 2 shows the results without measurement noise as a function of the number of RTO iterations. The figure shows the evolution of the cost (ROI) and of the constraint to be activated (F_P) and leads to the following conclusions:

- In the absence of noise, the RTO schemes MA, MAy, and MA-$\mathcal{L}$ converge to the optimal ROI value of 725.12 \$/s, but they all require the estimation of plant gradients along the 4 inputs. The MA schemes with known active set do not reach plant optimality since the reduced input directions are chosen orthogonal to the gradients of the active *model* constraints. MA-kAS reaches a ROI value of 721 \$/s with the plant gradients estimated in 3 input directions. With the ADMA-kAS schemes, the same value of 721 \$/s can be achieved with the estimation of plant gradients in only 2 input directions, while a value of 712 \$/s is reached with the estimation of plant gradients in 1 input direction. By using ADMA schemes with unknown active set, the results are worse than with the ADMA-kAS schemes for the same number of input directions (not shown). Indeed, one obtains a ROI value of 710 and 715 \$/s with the estimation of plant gradients in 1 and 2 input directions, respectively.

Fig. 3 presents the results with measurement noise as a function of the number of experimental runs corresponding to the first 20 RTO iterations. The following conclusions can be drawn:

- In the presence of noise, the various MA schemes reach cost values above 720 \$/s, while the ADMA-kAS schemes with fewer input directions are only marginally inferior with cost values between 710 and 720 \$/s. The fact that the ADMA-kAS schemes are only slightly affected by noise indicates that the dominant input directions are able to provide a large signal-to-noise ratio for the estimation of plant gradients.
- As shown by the performance in the first 10 to 15 experimental runs, both the schemes with known active set and the active directional schemes allow reaching near plant optimality faster than the basic schemes MA, MAy, and MA-$\mathcal{L}$. This shows the practical importance of estimating the plant gradients along a reduced number of input directions. Indeed, improvement speed is often more important than optimality upon convergence. In an industrial setting, desired (such as setpoint changes) and undesired disturbances are frequent, and the process is more often trying to adapt to new conditions rather than converging to some optimal operating point.
- The convergence of the Lagrangian schemes is comparable to that of the corresponding standard and output

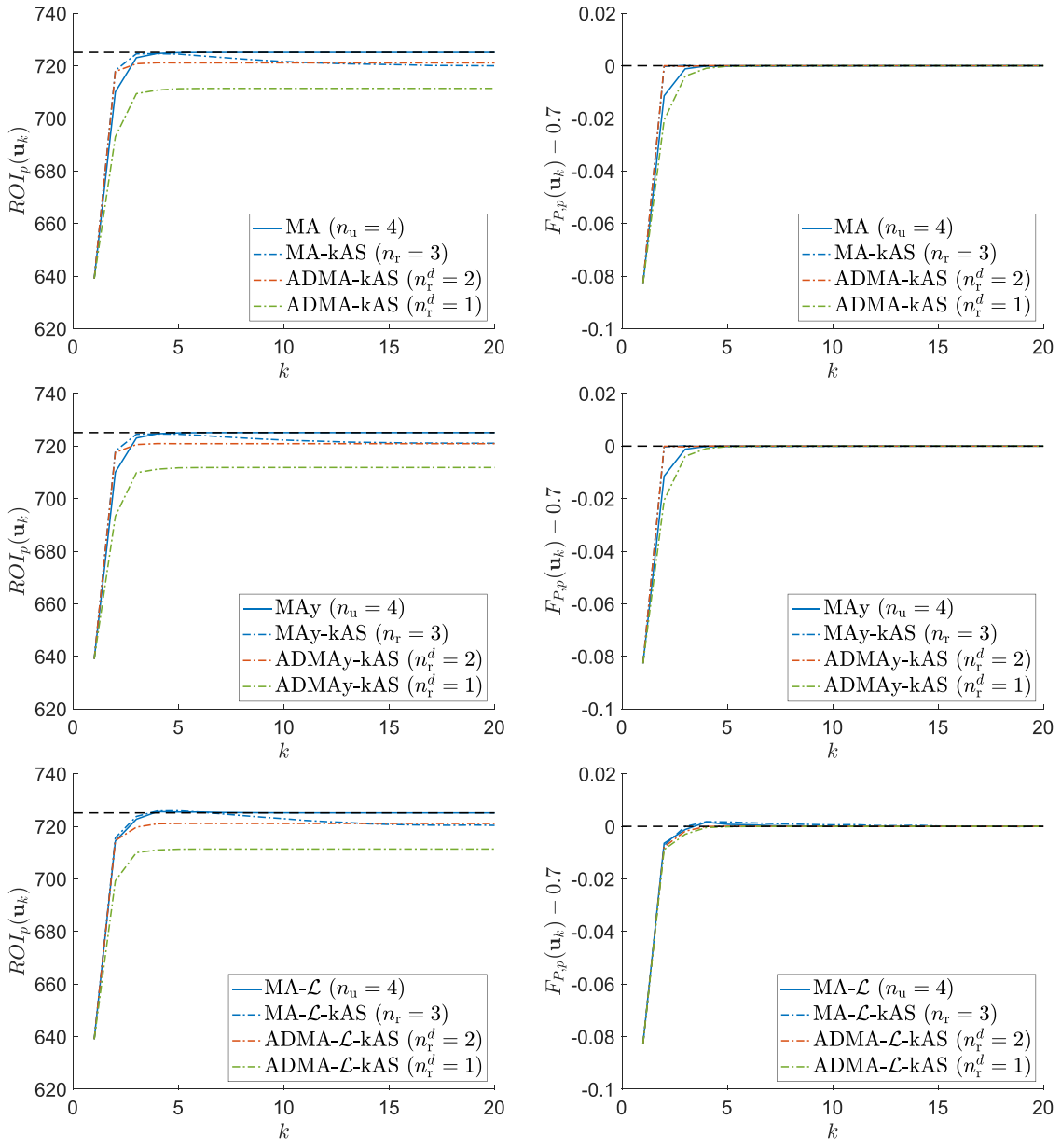

Fig. 2. Plant performance without measurement noise as a function of the number of RTO iterations. Cost (left) and constraint to be activated (right) for MA/MA-kAS/ADMA-kAS (top), MAy/MAy-kAS/ADMAy-kAS (middle), and MA-$\mathcal{L}$/MA-$\mathcal{L}$-kAS/ADMA-$\mathcal{L}$-kAS (bottom). Blue solid lines: basic schemes with $n_u = 4$. Blue dashed-dotted lines: known-active-set schemes with $n_r = 3$. Red dashed-dotted lines: active directional, known-active-set schemes with $n_r^d = 2$. Green dashed-dotted lines: active directional, known-active-set schemes with $n_r^d = 1$. Black dashed lines: values at the plant optimum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

schemes. However, the Lagrangian schemes exhibit an underdamped behavior and may even result in constraint violations prior to convergence, which supports Remark 6 in Appendix B.

In addition, a comparison of Figs. 2 and 3 allows drawing additional conclusions:

- The ADMA schemes with known active set provide good performance both in the absence and in the presence of noise. This indicates that the dominant input directions, which should not only be sensitive to the plant-model mismatch, but also orthogonal to the gradients of the active plant constraints, are well approximated by the model. Hence, although the model is not capable of predicting good optimal input values, it is useful to provide good dominant input directions for the ADMA schemes with known active set.
- In a practical context with measurement noise, the effect of noise may dominate the loss of optimality resulting from the reduced number of input directions in known-active-set or active directional formulations. In such cases, the method proposed in Papasavvas (2021) consisting in reducing the effects of noise without increasing the number of experimental runs could be used. It involves collecting more measurements once steady state has been detected and using appropriate statistical analysis to reject a large part of measurement noise. However, the collection of more measurements after each steady state would imply an increase of the time spent for each experimental run, which would contradict the purpose of reducing the time needed to reach near plant optimality. Hence, there is a clear tradeoff between the time needed to collect more measurements and reduce measurement noise and the time required by the RTO scheme to

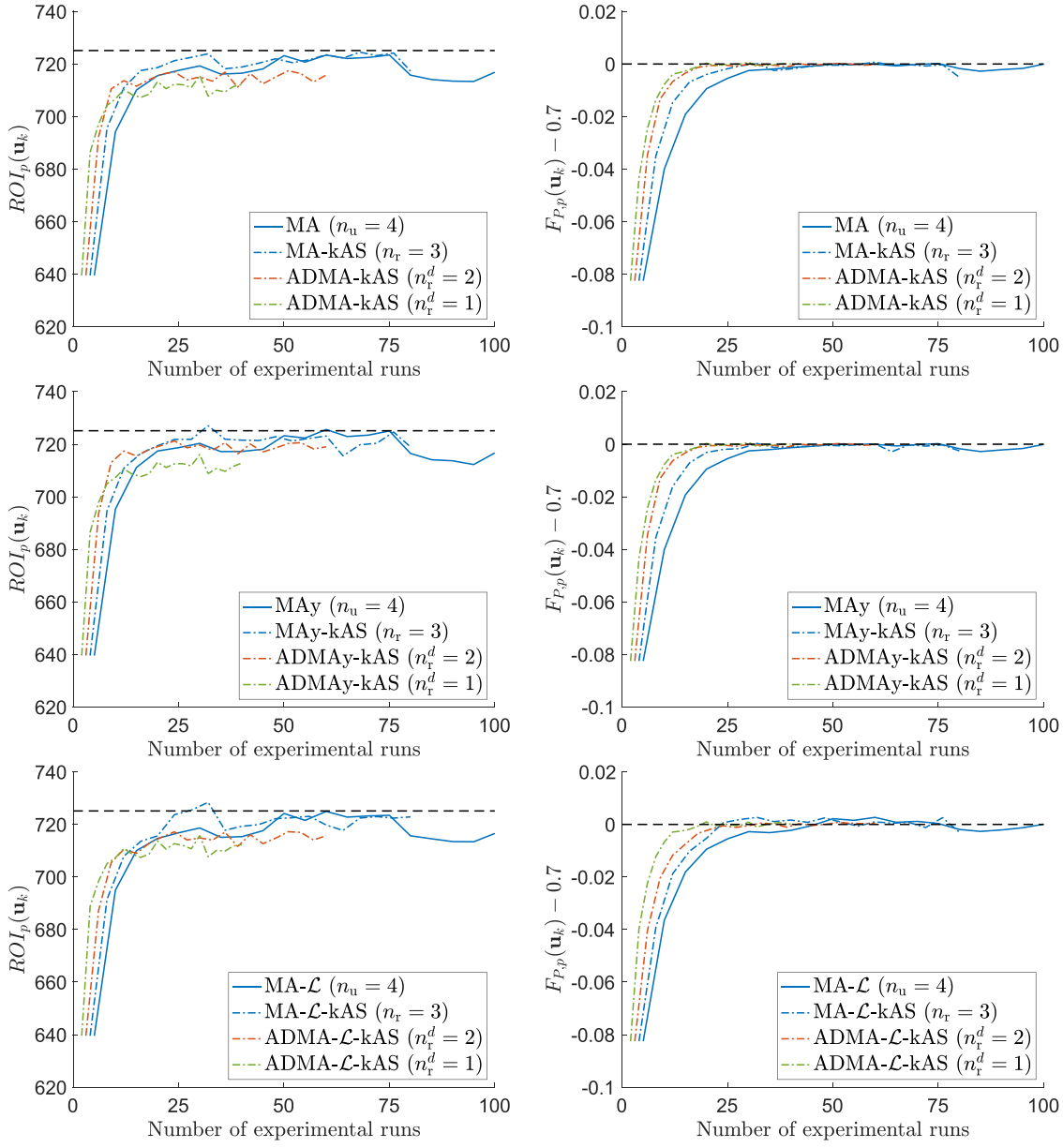

Fig. 3. Plant performance with measurement noise as a function of the number of experimental runs. Cost (left) and constraint to be activated (right) for MA/MA-kAS/ADMA-kAS (top), MAy/MAy-kAS/ADMAy-kAS (middle), and MA-$\mathcal{L}$/MA-$\mathcal{L}$-kAS/ADMA-$\mathcal{L}$-kAS (bottom). Blue solid lines: basic schemes with $n_u = 4$. Blue dashed-dotted lines: known-active-set schemes with $n_r = 3$. Red dashed-dotted lines: active directional, known-active-set schemes with $n_r^d = 2$. Green dashed-dotted lines: active directional, known-active-set schemes with $n_r^d = 1$. Black dashed lines: values at the plant optimum. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

reach plant optimality. The optimal value of this tradeoff depends on the problem at hand.

5. Conclusions

This paper has discussed practical aspects of modifier adaptation in the context of plant-model mismatch. In that context, it is important to have strong handles to modify the model and achieve KKT matching between the model and the plant. With respect to partial KKT matching, the most important KKT conditions are the active constraints and the Lagrangian gradient along the dominant input directions. Hence, one can reduce the number of input directions by considering that either the active set or the dominant input directions or both are known. With that information, one can propose additional MA formulations that are tailored to reducing the experimental effort that is needed to drive the plant near optimality. In this work, 4 MA or ADMA schemes were known a priori (MA, MAy, MA-$\mathcal{L}$, and ADMA), while 8 additional/novel ones have been proposed as per Table 2.

The investigation of the case study has been quite instructive. Note that this is not the Williams–Otto reactor that has been used extensively in the literature, but rather the Williams–Otto plant that consists of that reactor, a decanter, a distillation column and a splitter, with in addition a recycle stream. The steady-state model is kept purposely simple, but the real-time optimization task is challenging due to the significant amount of structural plant-model mismatch and the existence of four inputs. Furthermore, we found the performance comparison of the various MA schemes as a function of the number of experimental runs particularly interesting, as it really expresses the true experimental cost associated with a RTO scheme. The results of the simulation study showed that it is possible to achieve almost the same cost by using 1 or 2 input di-

rections instead of the original 4 directions, which allows reaching near plant optimality in the first 10 to 15 experimental runs. For all these reasons, we found our particular problem setting for the Williams–Otto plant very useful for comparing RTO algorithms and therefore we encourage other researchers to use the problem as a comparison case study for RTO. In particular, it could be instructive for the community to compare the performance of novel RTO schemes with the results in Figs. 2 and 3.

The proposed methodology was developed and used in the context of *explicit optimization*, that is, adaptation of a model in real time with subsequent use for numerical optimization. However, in the context of plant-model mismatch, there is a growing interest in *implicit optimization* schemes, whereby plant optimality is enforced via self-optimizing control. The major challenge there deals with the selection of both the controlled variables (typically dominant KKT conditions) and the manipulated variables (inputs that are most relevant for optimality). This selection task is typically done offline based on a model of the process. The techniques developed in this work regarding the selection of the dominant KKT conditions and the associated dominant input directions appear to be quite relevant to the field of implicit optimization as well.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

D. Rodrigues: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **A.G. Marchetti:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing. **D. Bonvin:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Acknowledgements

The authors would like to thank Martand Singhal and Timm Faulwasser, both formerly at EPFL, for discussions related to the topic of this paper, and Aris Papasavvas, University of Edinburgh, for constructive feedback on the paper.

Appendix A. Parametric sensitivity analysis

The effect of parametric perturbations on the KKT conditions is key to the concept of directionality developed in this paper. In this appendix, we recall the techniques developed by Singhal (2018) for parametric sensitivity analysis in the context of KKT matching for MA schemes to ensure that the paper is self-contained in its methodology.

We begin by formally defining the terms influential and less influential parameter directions. Consider the vector map $\mathbf{f}(\boldsymbol{\theta})$, with $\mathbf{f}:\mathbb{R}^{n_\theta}\to\mathbb{R}^{n_\mathrm{f}}$. This map could for example represent the effect of n_θ parameters on n_f active KKT conditions. In addition, consider the influential and less influential spaces $\mathcal{I}$ and $\mathcal{LI}$ that are orthogonal complements to each other such that they form a direct sum on $\mathbb{R}^{n_\theta}$, that is, $\mathcal{I}\oplus\mathcal{LI}=\mathbb{R}^{n_\theta}$.

Definition 1 (Influential and less influential parameter directions; Smith, 2014)**.** Any parameter direction $\boldsymbol{\theta}_1\in\mathcal{LI}\subset\mathbb{R}^{n_\theta}$ is said to be less influential for the vector function $\mathbf{f}$ if $\|\mathbf{f}(\boldsymbol{\theta}_1)-\mathbf{f}(\boldsymbol{\theta}_2)\|<\epsilon$ for all $\boldsymbol{\theta}_2\in\mathcal{LI}$, where ϵ is a small positive scalar. The orthogonal complement of $\mathcal{LI}$ is the subspace $\mathcal{I}$ of influential directions.

There exist several tools to determine influential parameter directions. For instance, Fisher information is the classical way of finding these influential parameter directions. Another notable technique is that of Sobol indices that quantify the influence of a parameter on the variance of the response (Sobol, 2001). Although the latter approach has the advantage that it does not require any linearization, the computation of sensitivity indices can be prohibitively expensive for large parameter dimensions. The alternative is to use linearization techniques to compute local sensitivities that are used to determine influential parameter directions (Abdel-Khalik et al., 2013; Constantine et al., 2014). The approximation to global sensitivities is then obtained by aggregating local sensitivities evaluated at random parameter values sampled from an admissible parameter set. In the following subsection, we recall one such approach known as active subspaces (Russi, 2010; Constantine, 2015).

A1. Local sensitivity analysis

Consider the n_f-dimensional vector $\mathbf{f}(\boldsymbol{\theta})$ and the $(n_\mathrm{f}\times n_\theta)$-dimensional local sensitivity matrix $\mathbf{S}$ evaluated at $\bar{\boldsymbol{\theta}}$:

$$\mathbf{S}:=\left(\frac{\partial\mathbf{f}}{\partial\boldsymbol{\theta}}\right)_{\bar{\boldsymbol{\theta}}}. \tag{38}$$

One can compute the domain (parameter) and image (KKT) directions at $\bar{\boldsymbol{\theta}}$ by singular value decomposition of $\mathbf{S}$ as follows:

$$\mathbf{S}=\mathbf{Q}\,\boldsymbol{\Sigma}\,\mathbf{R}^\mathrm{T}, \tag{39}$$

with the $(n_\mathrm{f}\times n_\mathrm{f})$-dimensional orthonormal matrix $\mathbf{Q}=[\mathbf{q}_1\ \dots\ \mathbf{q}_{n_\mathrm{f}}]$, the $(n_\theta\times n_\theta)$-dimensional orthonormal matrix $\mathbf{R}=[\mathbf{r}_1\ \dots\ \mathbf{r}_{n_\theta}]$ and the $(n_\mathrm{f}\times n_\theta)$-dimensional singular value matrix $\boldsymbol{\Sigma}$ with the singular values $\sigma_1\geq\ldots\geq\sigma_{n_\mathrm{S}}\geq 0$ on the main diagonal and zeros everywhere else, where $n_\mathrm{S}=\min(n_\mathrm{f},n_\theta)$.

The number of dominant directions, n_S^d, is determined as the first occurrence of $\sigma_{n_\mathrm{S}^d}\gg\sigma_{n_\mathrm{S}^d+1}$. It follows that the local sensitivity matrix $\mathbf{S}$ can be approximated as:

$$\mathbf{S}\approx\mathbf{S_1}=\mathbf{Q}_1\,\boldsymbol{\Sigma}_1\,\mathbf{R}_1^\mathrm{T}, \tag{40}$$

with $\mathbf{Q}_1=[\mathbf{q}_1\ \dots\ \mathbf{q}_{n_\mathrm{S}^d}]$, $\mathbf{R}_1=[\mathbf{r}_1\ \dots\ \mathbf{r}_{n_\mathrm{S}^d}]$ and $\boldsymbol{\Sigma}_1=\mathrm{diag}(\sigma_1,\ldots,\sigma_{n_\mathrm{S}^d})$.

The vectors in $\mathbf{Q}_1$ describe n_S^d dominant orthogonal directions in the space of KKT conditions, while the vectors in $\mathbf{R}_1$ describe n_S^d dominant orthogonal directions in the parameter space, the relative strength of each direction being given by the singular values in $\boldsymbol{\Sigma}_1$.

A2. Global sensitivity analysis

For the vector map $\mathbf{f}(\boldsymbol{\theta})$, with $\mathbf{f}:\mathbb{R}^{n_\theta}\to\mathbb{R}^{n_\mathrm{f}}$, we can compute a global sensitivity by considering the probability distribution of the parameters $\boldsymbol{\theta}$. We propose to determine both the domain and image active subspaces.

For the domain (parameter) space, the sensitivity matrix $\mathbf{F}\in\mathbb{R}^{n_\theta\times n_\theta}$ is considered,

$$\mathbf{F}=\int\nabla_\theta\mathbf{f}^\mathrm{T}(\boldsymbol{\theta})\,\nabla_\theta\mathbf{f}(\boldsymbol{\theta})\,\rho(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta}, \tag{41}$$

where $\nabla_\theta\mathbf{f}(\boldsymbol{\theta}):=\frac{\partial\mathbf{f}}{\partial\boldsymbol{\theta}}(\boldsymbol{\theta})$ is the Jacobian of the function $\mathbf{f}$ with respect to $\boldsymbol{\theta}$, and ρ is the probability density function of $\boldsymbol{\theta}$ over the admissible bounded set $\boldsymbol{\Theta}$ with $\rho=0$ for $\boldsymbol{\theta}\notin\boldsymbol{\Theta}$. Here, the parameter vector $\boldsymbol{\theta}$ is a scaled version of the original parameters. Without loss of generality, the admissible parameter set can be taken as $\boldsymbol{\Theta}=[-1,\,1]^{n_\theta}\subseteq\mathbb{R}^{n_\theta}$. Note that, since $\mathbf{F}$ is symmetric and positive semi-definite, it diagonalizes as

$$\mathbf{F}=\mathbf{V}\,\boldsymbol{\Phi}\,\mathbf{V}^\mathrm{T},\quad \boldsymbol{\Phi}=\mathrm{diag}(\phi_1,\ldots,\phi_{n_\theta}), \tag{42}$$

with $\phi_1 \geq \ldots \geq \phi_{n_\theta} \geq 0$; $\mathbf{V} \in \mathbb{R}^{n_\theta \times n_\theta}$ is an orthonormal matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_{n_\theta}$ are the orthonormal eigenvectors of $\mathbf{F}$.

For the image (KKT) space, the sensitivity matrix $\mathbf{G} \in \mathbb{R}^{n_f \times n_f}$ is considered,

$$\mathbf{G} = \int \nabla_\theta \mathbf{f}(\boldsymbol{\theta})\, \nabla_\theta \mathbf{f}^{\mathrm{T}}(\boldsymbol{\theta})\, \rho(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}. \tag{43}$$

Since $\mathbf{G}$ is symmetric and positive semi-definite, it diagonalizes as

$$\mathbf{G} = \mathbf{W}\boldsymbol{\Gamma}\mathbf{W}^T, \quad \boldsymbol{\Gamma} = \mathrm{diag}\big(\gamma_1, \ldots, \gamma_{n_f}\big), \tag{44}$$

with $\gamma_1 \geq \ldots \geq \gamma_{n_f} \geq 0$; $\mathbf{W} \in \mathbb{R}^{n_f \times n_f}$ is an orthonormal matrix whose columns $\mathbf{w}_1, \ldots, \mathbf{w}_{n_f}$ are the orthonormal eigenvectors of $\mathbf{G}$.

Since obtaining the sensitivity matrices $\mathbf{F}$ in (41) and $\mathbf{G}$ in (43) analytically may be prohibitive in many cases, Constantine (2015) proposed to compute these matrices and the active subspaces via sampling-based schemes. Concretely, the following two approaches can be highlighted to approximate $\mathbf{F}$ and $\mathbf{G}$:

(1) Monte–Carlo sampling, which relies on randomly picking N sample points $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ from the probability density function $\rho(\boldsymbol{\theta})$ and assigning equal weights to each sample. Then, approximations of (41) and (43) are computed as follows:

$$\hat{\mathbf{F}} = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \mathbf{f}^{\mathrm{T}}(\boldsymbol{\theta}_n)\, \nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n), \tag{45}$$

$$\hat{\mathbf{G}} = \frac{1}{N}\sum_{n=1}^{N} \nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n)\, \nabla_\theta \mathbf{f}^{\mathrm{T}}(\boldsymbol{\theta}_n). \tag{46}$$

(2) Gaussian quadrature, which relies on choosing N sample points $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_N$ and N weights $w_1, \ldots, w_N$ such that the integrals of the form $\int v(\boldsymbol{\theta})\, \rho(\boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}$ are computed exactly for multivariate polynomials $v(\boldsymbol{\theta})$ up to some degree d as $\sum_{n=1}^{N} w_n v(\boldsymbol{\theta}_n)$. This method may be more efficient in terms of number of sample points and does not require sampling random parameter values. Then, approximations of (41) and (43) are computed as follows:

$$\hat{\mathbf{F}} = \sum_{n=1}^{N} w_n \nabla_\theta \mathbf{f}^{\mathrm{T}}(\boldsymbol{\theta}_n)\, \nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n), \tag{47}$$

$$\hat{\mathbf{G}} = \sum_{n=1}^{N} w_n \nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n)\, \nabla_\theta \mathbf{f}^{\mathrm{T}}(\boldsymbol{\theta}_n). \tag{48}$$

The sampling-based schemes are described in Algorithms 3 and 4.

A3. Computation of active subspaces

The model can be so complex that the analytical expressions in the sensitivity matrices $\mathbf{F}$ and $\mathbf{G}$ are intractable. Sampling-based approaches can be used to approximate these matrices, which is also discussed in Singhal et al. (2018). It is recommended to scale the parameters $\boldsymbol{\theta}$ so that they lie between -1 and 1. The dominant and less dominant domain and image spaces are computed by constructing the approximations $\hat{\mathbf{F}}$ and $\hat{\mathbf{G}}$ as given in Algorithms 3 and 4, respectively.

Note: For a single realization ($N = 1$) and assuming $n_\theta \geq n_f$, the $(n_\theta \times n_\theta)$-dimensional matrix $\hat{\mathbf{F}}$ is of rank n_f. It is then full rank due to the summation over $N \geq n_\theta$ realizations.

Algorithm 3 Computation of influential parameter space (dominant domain space).

For the n_f-dimensional vector function $\mathbf{f}(\boldsymbol{\theta})$, do:

Step 1: Draw N independent samples $\boldsymbol{\theta}_n$, $n = 1, \ldots, N$, from $\boldsymbol{\Theta}$ using the probability density $\rho(\boldsymbol{\theta})$.
Step 2: For each sample $\boldsymbol{\theta}_n$, compute the $(n_f \times n_\theta)$-dimensional Jacobian matrix $\nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n)$.
Step 3: Compute the $(n_\theta \times n_\theta)$-dimensional matrix $\hat{\mathbf{F}}$ as in (45) or (47).
Step 4: Compute the $(n_\theta \times n_\theta)$-dimensional orthonormal matrix $\hat{\mathbf{V}}$ by eigenvalue decomposition of $\hat{\mathbf{F}}$:
$\hat{\mathbf{F}} = \hat{\mathbf{V}}\,\hat{\boldsymbol{\Phi}}\,\hat{\mathbf{V}}^{\mathrm{T}}$, $\hat{\mathbf{V}} = [\hat{\mathbf{v}}_1 \ldots \hat{\mathbf{v}}_{n_\theta}]$, and $\hat{\boldsymbol{\Phi}} = \mathrm{diag}(\hat{\phi}_1, \ldots, \hat{\phi}_{n_\theta})$.
Step 5: Select the influential and less influential parameter spaces by partitioning the matrix $\hat{\mathbf{V}}$ as follows:
$\hat{\mathbf{V}}_1 = [\hat{\mathbf{v}}_1 \ldots \hat{\mathbf{v}}_{n_\theta^d}]$, $\hat{\mathbf{V}}_2 = [\hat{\mathbf{v}}_{n_\theta^d+1} \ldots \hat{\mathbf{v}}_{n_\theta}]$,
$n_\theta^d : \hat{\phi}_{n_\theta^d} \gg \hat{\phi}_{n_\theta^d+1}$,
$\mathcal{I} = \mathrm{col}\big(\hat{\mathbf{V}}_1\big)$, $\mathcal{LI} = \mathrm{col}\big(\hat{\mathbf{V}}_2\big)$.

Algorithm 4 Computation of dominant KKT elements (dominant image space).

For the n_f-dimensional vector function $\mathbf{f}(\boldsymbol{\theta})$, do:

Step 1: Draw N independent samples $\boldsymbol{\theta}_n$, $n = 1, \ldots, N$, from $\boldsymbol{\Theta}$ using the probability density $\rho(\boldsymbol{\theta})$.
Step 2: For each sample $\boldsymbol{\theta}_n$, compute the $(n_f \times n_\theta)$-dimensional Jacobian matrix $\nabla_\theta \mathbf{f}(\boldsymbol{\theta}_n)$.
Step 3: Compute the $(n_f \times n_f)$-dimensional matrix $\hat{\mathbf{G}}$ as in (46) or (48).
Step 4: Compute the $(n_f \times n_f)$-dimensional orthonormal matrix $\hat{\mathbf{W}}$ by eigenvalue decomposition of $\hat{\mathbf{G}}$:
$\hat{\mathbf{G}} = \hat{\mathbf{W}}\,\hat{\boldsymbol{\Gamma}}\,\hat{\mathbf{W}}^{\mathrm{T}}$, $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1 \ldots \hat{\mathbf{w}}_{n_f}]$, and $\hat{\boldsymbol{\Gamma}} = \mathrm{diag}(\hat{\gamma}_1, \ldots, \hat{\gamma}_{n_f})$.
Step 5: Select the dominant and less dominant KKT spaces by partitioning the matrix $\hat{\mathbf{W}}$ as follows:
$\hat{\mathbf{W}}_1 = [\hat{\mathbf{w}}_1 \ldots \hat{\mathbf{w}}_{n_f^d}]$, $\hat{\mathbf{W}}_2 = [\hat{\mathbf{w}}_{n_f^d+1} \ldots \hat{\mathbf{w}}_{n_f}]$,
$n_f^d : \hat{\gamma}_{n_f^d} \gg \hat{\gamma}_{n_f^d+1}$,
$\mathcal{I} = \mathrm{col}\big(\hat{\mathbf{W}}_1\big)$, $\mathcal{LI} = \mathrm{col}\big(\hat{\mathbf{W}}_2\big)$.

Appendix B. Convergence of MA-$\mathcal{L}$ to plant optimality

The following theorem proves that MA-$\mathcal{L}$ reaches plant optimality upon convergence.

Theorem B.1 (KKT matching for MA-$\mathcal{L}$)**.** *Let MA-$\mathcal{L}$ converge, with $\mathbf{u}_\infty = \lim_{k\to\infty} \mathbf{u}_k$ being a KKT point for the modified Problem (16) and $\boldsymbol{\nu}_\infty = \lim_{k\to\infty} \boldsymbol{\nu}_k \in \mathbb{R}^{n_g}$ the vector of corresponding Lagrange multipliers. Then, $\mathbf{u}_\infty$ is also a KKT point for the plant Problem (1).*

Proof. Let us introduce the Lagrangian function of Problem (16), namely, $\mathcal{L}_m(\mathbf{z}, \boldsymbol{\theta}) := \Phi_m(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\nu}^{\mathrm{T}}\, \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta})$. It follows from the definitions of Φ_m and $\mathbf{G}_m$ and the use of Eq. (15) that:

$$\mathbf{G}_m(\mathbf{u}_k, \boldsymbol{\theta}) = \mathbf{G}_p(\mathbf{u}_k), \tag{49a}$$

$$\frac{\partial \mathcal{L}_m}{\partial \mathbf{u}}(\mathbf{z}_k, \boldsymbol{\theta}) = \frac{\partial \mathcal{L}_p}{\partial \mathbf{u}}(\mathbf{z}_k). \tag{49b}$$

Upon convergence, that is, for $k \to \infty$, the KKT conditions read:

$$\frac{\partial \mathcal{L}_m}{\partial \mathbf{u}}(\mathbf{z}_\infty, \boldsymbol{\theta}) = \mathbf{0}, \tag{50a}$$

$$\mathbf{G}_m(\mathbf{u}_\infty, \boldsymbol{\theta}) \leq \mathbf{0}, \tag{50b}$$

$$(\boldsymbol{\nu}_\infty)^\top \mathbf{G}_m(\mathbf{u}_\infty, \boldsymbol{\theta}) = 0, \tag{50c}$$

$$\boldsymbol{\nu}_\infty \geq \mathbf{0}, \tag{50d}$$

or, with Eq. (49),

$$\frac{\partial \mathcal{L}_p}{\partial \mathbf{u}}(\mathbf{z}_\infty) = \mathbf{0}, \tag{51a}$$

$$\mathbf{G}_p(\mathbf{u}_\infty) \leq \mathbf{0}, \tag{51b}$$

$$(\boldsymbol{\nu}_\infty)^\top \mathbf{G}_p(\mathbf{u}_\infty) = 0, \tag{51c}$$

$$\boldsymbol{\nu}_\infty \geq \mathbf{0}, \tag{51d}$$

which are the KKT conditions of Problem (1). □

Remark 6. Although MA-$\mathcal{L}$ uses a single gradient modifier $\boldsymbol{\lambda}_k^{\mathcal{L}}$, it may result in slower convergence to plant optimality and larger constraint violations prior to convergence (Marchetti et al., 2016).

Appendix C. Convergence of MA-kAS to plant optimality

The following theorem proves that MA-kAS can reach plant optimality upon convergence.

Theorem C.1 (KKT matching for MA-kAS)**.** *Let MA-kAS converge, with* $\mathbf{u}_\infty = \lim_{k\to\infty} \mathbf{u}_k$ *being a KKT point for the modified Problem (27) and* $\boldsymbol{\nu}_\infty = \lim_{k\to\infty} \boldsymbol{\nu}_k \in \mathbb{R}^{n_g}$ *the vector of corresponding Lagrange multipliers. Furthermore, let* $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_\infty)\mathbf{N}_\infty = \mathbf{0}$*, which holds for either Case (i) or Case (ii) in Section 3.1.2. Then,* $\mathbf{u}_\infty$ *is also a KKT point for the plant Problem (1).*

Proof. Consider the modified cost and constraint functions Φ_m and $\mathbf{G}_m$ of Problem (27) and the corresponding Lagrangian $\mathcal{L}_m(\mathbf{z}, \boldsymbol{\theta}) := \Phi_m(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\nu}^\top \mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta})$. Using Eqs. (24) and (26), one can write:

$$\begin{aligned}\mathbf{G}_m(\mathbf{u}, \boldsymbol{\theta}) = \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) &+ \Big(\mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta})\Big) \\ &+ \Big(\nabla_r \mathbf{G}_p(\mathbf{u}_k) - \nabla_r \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta})\Big)\mathbf{N}_k^\top (\mathbf{u} - \mathbf{u}_k),\end{aligned} \tag{52a}$$

$$\begin{aligned}\mathcal{L}_m(\mathbf{z}, \boldsymbol{\theta}) = \Phi(\mathbf{u}, \boldsymbol{\theta}) &+ \Big(\nabla_r \Phi_p(\mathbf{u}_k) - \nabla_r \Phi(\mathbf{u}_k, \boldsymbol{\theta})\Big)\mathbf{N}_k^\top \mathbf{u} \\ &+ \boldsymbol{\nu}^\top \Big[\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \Big(\mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta})\Big) \\ &+ \Big(\nabla_r \mathbf{G}_p(\mathbf{u}_k) - \nabla_r \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta})\Big)\mathbf{N}_k^\top (\mathbf{u} - \mathbf{u}_k)\Big],\end{aligned} \tag{52b}$$

$$\begin{aligned}\frac{\partial \mathcal{L}_m}{\partial \mathbf{u}}(\mathbf{z}, \boldsymbol{\theta}) = \frac{\partial \Phi}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) &+ \Big(\nabla_r \Phi_p(\mathbf{u}_k) - \nabla_r \Phi(\mathbf{u}_k, \boldsymbol{\theta})\Big)\mathbf{N}_k^\top \\ &+ \boldsymbol{\nu}^\top \left[\frac{\partial \mathbf{G}}{\partial \mathbf{u}}(\mathbf{u}, \boldsymbol{\theta}) + \Big(\nabla_r \mathbf{G}_p(\mathbf{u}_k) - \nabla_r \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta})\Big)\mathbf{N}_k^\top\right].\end{aligned} \tag{52c}$$

Post-multiplying Eq. (52c) with $\mathbf{N}_k$, considering that $\mathbf{N}_k^\top \mathbf{N}_k = \mathbf{I}_{n_r}$ and evaluating Eqs. (52a) and (52c) at $\mathbf{u}_k$ gives:

$$\mathbf{G}_m(\mathbf{u}_k, \boldsymbol{\theta}) = \mathbf{G}_p(\mathbf{u}_k), \tag{53a}$$

$$\begin{aligned}\nabla_r \mathcal{L}_m(\mathbf{z}_k, \boldsymbol{\theta}) = \frac{\partial \mathcal{L}_m}{\partial \mathbf{u}}(\mathbf{z}_k, \boldsymbol{\theta})\,\mathbf{N}_k = \nabla_r \Phi_p(\mathbf{u}_k) \\ + \boldsymbol{\nu}_k^\top \nabla_r \mathbf{G}_p(\mathbf{u}_k) = \nabla_r \mathcal{L}_p(\mathbf{z}_k).\end{aligned} \tag{53b}$$

From Eq. (22), the KKT conditions of Problem (27) upon convergence are:

$$\mathbf{G}_m^a(\mathbf{u}_\infty, \boldsymbol{\theta}) = \mathbf{0}, \tag{54a}$$

$$\nabla_r \mathcal{L}_m^a(\mathbf{z}_\infty, \boldsymbol{\theta}) = \frac{\partial \mathcal{L}_m^a}{\partial \mathbf{u}}(\mathbf{z}_\infty, \boldsymbol{\theta})\,\mathbf{N}_\infty = \mathbf{0}, \tag{54b}$$

or, with Eq. (53),

$$\mathbf{G}_p^a(\mathbf{u}_\infty) = \mathbf{0}, \tag{55a}$$

$$\nabla_r \mathcal{L}_p^a(\mathbf{z}_\infty, \boldsymbol{\theta}) = \mathbf{0}, \tag{55b}$$

which are the KKT conditions of Problem (1) when $\frac{\partial \mathbf{G}_p^a}{\partial \mathbf{u}}(\mathbf{u}_\infty)\mathbf{N}_\infty = \mathbf{0}$. □

Appendix D. Williams–Otto plant

D1. Plant equations

The processing units of the plant are described next. The steady-state material balances and the reaction equations around the reactor are given as:

$$F_A + F_{T,\mathrm{A}} - F_{R,\mathrm{A}} - V\,r_1 = 0, \tag{56a}$$

$$F_B + F_{T,\mathrm{B}} - F_{R,\mathrm{B}} - V\,r_1 - V\,r_2 = 0, \tag{56b}$$

$$F_{T,\mathrm{C}} - F_{R,\mathrm{C}} + \frac{M_\mathrm{C}}{M_\mathrm{A}} V\,r_1 - \frac{M_\mathrm{C}}{M_\mathrm{B}} V\,r_2 - V\,r_3 = 0, \tag{56c}$$

$$F_{T,\mathrm{E}} - F_{R,\mathrm{E}} + \frac{M_\mathrm{E}}{M_\mathrm{B}} V\,r_2 = 0, \tag{56d}$$

$$F_{T,\mathrm{G}} - F_{R,\mathrm{G}} + \frac{M_\mathrm{G}}{M_\mathrm{C}} V\,r_3 = 0, \tag{56e}$$

$$F_{T,\mathrm{P}} - F_{R,\mathrm{P}} + \frac{M_\mathrm{P}}{M_\mathrm{B}} V\,r_2 - \frac{M_\mathrm{P}}{M_\mathrm{C}} V\,r_3 = 0, \tag{56f}$$

$$F_R - (F_{R,\mathrm{A}} + F_{R,\mathrm{B}} + F_{R,\mathrm{C}} + F_{R,\mathrm{E}} + F_{R,\mathrm{G}} + F_{R,\mathrm{P}}) = 0, \tag{56g}$$

$$r_1 = \mathrm{k}_1 \frac{F_{R,\mathrm{A}} F_{R,\mathrm{B}}}{(F_R)^2}, \quad r_2 = \mathrm{k}_2 \frac{F_{R,\mathrm{B}} F_{R,\mathrm{C}}}{(F_R)^2}, \quad r_3 = \mathrm{k}_3 \frac{F_{R,\mathrm{C}} F_{R,\mathrm{P}}}{(F_R)^2}, \tag{56h}$$

$$\mathrm{k}_i = \mathrm{A}_i \exp\left(\frac{-\mathrm{E}_i}{T_r + 273.15}\right), \qquad i = 1, 2, 3. \tag{56i}$$

The decanter unit assumes perfect recovery of the product G. The material balances for the decanter read:

$$F_{S,i} = F_{R,i}, \quad i = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{E}, \mathrm{P}, \tag{57a}$$

$$F_{S,\mathrm{G}} = 0, \tag{57b}$$

$$F_S = F_{S,\mathrm{A}} + F_{S,\mathrm{B}} + F_{S,\mathrm{C}} + F_{S,\mathrm{E}} + F_{S,\mathrm{P}}, \tag{57c}$$

$$F_{G,i} = 0, \quad i = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{E}, \mathrm{P}, \tag{57d}$$

$$F_{G,\mathrm{G}} = F_{R,\mathrm{G}}, \tag{57e}$$

$$F_G = F_{G,\mathrm{G}}. \tag{57f}$$

The distillation column assumes the separation of pure product P overhead but also that some of the product is retained in the bottoms due to formation of an azeotrope with E (the fraction β of the mass flowrate of species E in the column feed is taken as the amount of P retained in the bottoms). The material balances for the distillation column read:

$$F_{P,i} = 0, \quad i = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{E}, \tag{58a}$$

$$F_{P,\mathrm{P}} = F_{S,\mathrm{P}} - \beta\,F_{S,\mathrm{E}}, \tag{58b}$$

$$F_P = F_{P,\mathrm{P}}, \tag{58c}$$

$$F_{Y,i} - F_{S,i} = 0, \quad i = \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{E}, \tag{58d}$$

$$F_{Y,\mathrm{P}} = \beta\,F_{S,\mathrm{E}}, \tag{58e}$$

$$F_Y = F_{Y,\mathrm{A}} + F_{Y,\mathrm{B}} + F_{Y,\mathrm{C}} + F_{Y,\mathrm{E}} + F_{Y,\mathrm{P}}. \tag{58f}$$

The splitter equations are:

$$F_{T,i} = \alpha\, F_{Y,i}, \quad i = \mathrm{A, B, C, E, P}, \tag{59a}$$

$$F_T = F_{T,\mathrm{A}} + F_{T,\mathrm{B}} + F_{T,\mathrm{C}} + F_{T,\mathrm{E}} + F_{T,\mathrm{P}}, \tag{59b}$$

$$F_{D,i} = (1-\alpha)\, F_{Y,i}, \quad i = \mathrm{A, B, C, E, P}, \tag{59c}$$

$$F_D = F_{D,\mathrm{A}} + F_{D,\mathrm{B}} + F_{D,\mathrm{C}} + F_{D,\mathrm{E}} + F_{D,\mathrm{P}}. \tag{59d}$$

The parameter values for the plant are given in Table 3.

D2. Model equations

The material balance and reaction equations around the reactor for the 2-reaction model are:

$$F_A + F_{T,\mathrm{A}} - F_{R,\mathrm{A}} - V\,\bar{r}_1 - V\,\bar{r}_2 = 0, \tag{60a}$$

$$F_B + F_{T,\mathrm{B}} - F_{R,\mathrm{B}} - 2V\,\bar{r}_1 - V\,\bar{r}_2 = 0, \tag{60b}$$

$$F_{T,\mathrm{E}} - F_{R,\mathrm{E}} + \frac{M_\mathrm{E}}{M_\mathrm{A}} V\,\bar{r}_1 = 0, \tag{60c}$$

$$F_{T,\mathrm{G}} - F_{R,\mathrm{G}} + \frac{M_\mathrm{G}}{M_\mathrm{A}} V\,\bar{r}_2 = 0, \tag{60d}$$

$$F_{T,\mathrm{P}} - F_{R,\mathrm{P}} + \frac{M_\mathrm{P}}{M_\mathrm{A}} V\,\bar{r}_1 - \frac{M_\mathrm{P}}{M_\mathrm{A}} V\,\bar{r}_2 = 0, \tag{60e}$$

$$F_R - (F_{R,\mathrm{A}} + F_{R,\mathrm{B}} + F_{R,\mathrm{E}} + F_{R,\mathrm{G}} + F_{R,\mathrm{P}}) = 0, \tag{60f}$$

$$\bar{r}_1 = \bar{\mathrm{k}}_1 \frac{F_{R,\mathrm{A}}\,(F_{R,\mathrm{B}})^2}{(F_R)^3}, \quad \bar{r}_2 = \bar{\mathrm{k}}_2 \frac{F_{R,\mathrm{A}}\,F_{R,\mathrm{B}}\,F_{R,\mathrm{P}}}{(F_R)^3}, \tag{60g}$$

$$\bar{\mathrm{k}}_i = \bar{\mathrm{A}}_i \exp\left(\frac{-\bar{\mathrm{E}}_i}{T_r + 273.15}\right); \qquad i = 1, 2. \tag{60h}$$

With regard to the other units, the model has the same material balance equations as those for the plant. Of course, the intermediate species C is dropped from the equations. The model parameter values $\boldsymbol{\theta}$, as well as the uncertainty ranges and the probability distribution that are used to construct the probability density function $\rho(\boldsymbol{\theta})$ in Appendix A.3, are given in Table 4. Note that the limits of the uncertainty ranges have also been used as bounds for the estimation of the model parameter values $\boldsymbol{\theta}$.

References

Abdel-Khalik, H.S., Bang, Y., Wang, C., 2013. Overview of hybrid subspace methods for uncertainty quantification, sensitivity analysis. Ann. Nucl. Energy 52, 28–46. doi:10.1016/j.anucene.2012.07.020.

Bazaraa, M.S., Sherali, H.D., Shetty, C.M., 2006. Nonlinear Programming: Theory and Algorithms, third ed. John Wiley and Sons, New Jersey.

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

Chen, C.Y., Joseph, B., 1987. On-line optimization using a two-phase approach: an application study. Ind. Eng. Chem. Res. 26 (9), 1924–1930.

Constantine, P.G., 2015. Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies. SIAM, Philadelphia, PA, USA.

Constantine, P.G., Dow, E., Wang, Q., 2014. Active subspace methods in theory and practice: applications to kriging surfaces. SIAM J. Sci. Comput. 36 (4), A1500–A1524.

Costello, S., François, G., Bonvin, D., 2016. A directional modifier-adaptation algorithm for real-time optimization. J. Process Control 39, 64–76.

François, G., Srinivasan, B., Bonvin, D., 2005. Use of measurements for enforcing the necessary conditions of optimality in the presence of constraints and uncertainty. J. Process Control 15 (6), 701–712.

Gao, W., Wenzel, S., Engell, S., 2015. Comparison of modifier adaptation schemes in real-time optimization. IFAC-PapersOnLine 48 (8), 182–187. doi:10.1016/j.ifacol.2015.08.178. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015

Mandur, J.S., Budman, H.M., 2015. Simultaneous model identification and optimization in presence of model-plant mismatch. Chem. Eng. Sci. 129 (10), 106–115.

Marchetti, A.G., de Avila Ferreira, T., Costello, S., Bonvin, D., 2020. Modifier adaptation as a feedback control scheme. Ind. Eng. Chem. Res. 59 (6), 2261–2274.

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

Marchetti, A.G., François, G., Faulwasser, T., Bonvin, D., 2016. Modifier adaptation for real-time optimization—Methods and applications. Processes 4 (55), 1–35. doi:10.3390/pr4040055.

Marlin, T., Hrymak, A., 1997. Real-time operations optimization of continuous processes. In: AICHE Symposium Series - CPC-V, vol. 93, pp. 156–164.

Navia, D., Gloria, G., De Prada, C., 2013. Nested modifier-adaptation for RTO in Otto-Williams reactor. In: IFAC Proc. Dycops., vol. 46, pp. 123–128. doi:10.3182/20131218-3-IN-2045.00101.

Papasavvas, A., 2021. From real-time optimization techniques to an autopilot for steady-state processes. arXiv preprint arXiv:2108.08715

Papasavvas, A., de Avila Ferreira, T., Marchetti, A.G., Bonvin, D., 2019. Analysis of output modifier adaptation for real-time optimization. Comput. Chem. Eng. 121, 285–293.

Roberts, P.D., 1995. Coping with model-reality differences in industrial process optimisation—A review of integrated system optimisation and parameter estimation (ISOPE). Comput. Ind. 26, 281–290.

Russi, T.M., 2010. Uncertainty Quantification with Experimental Data and Complex System Models. University of California, Berkeley.

Singhal, M., 2018. On Enforcing the Necessary Conditions of Optimality Under Plant-Model Mismatch—What to Measure and What to Adapt?. Ecole Polytechnique Fédérale de Lausanne Ph.D. thesis, 8803.

Singhal, M., Marchetti, A.G., Faulwasser, T., Bonvin, D., 2018. Active directional modifier adaptation for real-time optimization. Comput. Chem. Eng. 115 (6), 246–261.

Smith, R.C., 2014. Uncertainty Quantification: Theory, Implementation, and Applications. SIAM Computational Science & Engineering Series: Philadelphia, USA.

Sobol, I.M., 2001. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. Math. Comput. Simul. 55 (1), 271–280. doi:10.1016/S0378-4754(00)00270-6. The Second IMACS Seminar on Monte Carlo Methods

Srinivasan, B., Bonvin, D., Visser, E., Palanki, S., 2003. Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. Comput. Chem. Eng. 44, 27–44.

Williams, T., Otto, R., 1960. A generalized chemical processing model for the investigation of computer control. Trans. Am. Inst. Electr. Eng., Part I 79 (5), 458–473.