Computers and Chemical Engineering 115 (2018) 246–261

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Process dynamics, control and monitoring

Active directional modifier adaptation for real-time optimization

M. Singhal[a], A.G. Marchetti[a,1], T. Faulwasser[b], D. Bonvin[a,*]

[a] *Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*
[b] *Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, 76344 Eggenstein-Leopoldshafen, Germany*

ARTICLE INFO

Article history:
Received 12 September 2017
Revised 12 February 2018
Accepted 18 February 2018

Keywords:
Real-time optimization
Plant-model mismatch
Modifier adaptation
Input dimension reduction

ABSTRACT

Modifier adaptation is a real-time optimization (RTO) methodology that uses plant gradient estimates to correct model gradients, thereby driving the *plant* to optimality. However, obtaining accurate gradient estimates requires costly plant experiments at each RTO iteration. In directional modifier adaptation (DMA), the model gradients are corrected only in a small subspace of the input space, thus requiring fewer plant experiments. DMA selects the input subspace *offline* based on the *local* sensitivity of the Lagrangian gradient with respect to the uncertain model parameters. Here, we propose an extension, whereby the input subspace is selected *at each RTO iteration* via *global* sensitivity analysis, thus making the approach more reactive to changes and robust to large parametric uncertainties. Simulation results performed on the run-to-run optimization of two different semi-batch reactors show that the proposed approach finds a nice balance between experimental cost and optimality.

1. Introduction

Industrial plants target at optimizing process economics, while respecting operational constraints such as those on product quality, safety, and environmental regulations. In the presence of plant-model mismatch and process disturbances, real-time optimization (RTO) plays a pivotal role toward operating the plant optimally. RTO typically relies on the accuracy of the process model and/or the availability of plant measurements. RTO strategies differ in the way they exploit the available data and the model to update the operating point. For instance, the most common RTO strategy proceeds by first adapting the model parameters using experimental data and then optimizing the plant economics over the adapted model. This iterative approach is known as the two-step approach (Chen and Joseph, 1987). The two-step approach is intuitive and has become industrial practice in many process industries (Naysmith and Douglas, 1995). However, this approach typically converges to a sub-optimal solution in the presence of structural plant-model mismatch (Forbes and Marlin, 1996; Marchetti, 2009).

An alternative RTO strategy consists in adapting the optimization problem directly, while keeping the model parameters at their nominal values. This involves the adaptation of bias terms added to the constraints of the optimization problem (zeroth-order corrections). This approach, which is known as constraint adaptation (Chachuat et al., 2008), has shown promising results on an experimental solid-oxide fuel cell setup developed for industrial use (Bunin et al., 2012). In addition to these bias corrections, modifier-adaptation (MA) schemes include (first-order) gradient correction terms in the cost and constraint functions of the optimization problem (Marchetti et al., 2009). MA represents an appealing solution in the presence of plant-model mismatch as it guarantees the satisfaction of the plant first-order Karush–Kuhn–Tucker (KKT) conditions upon convergence. For the implementation of MA, plant measurements are expected to be sufficiently rich to allow good estimates of the plant cost and constraint values and of their gradients. The most straightforward way of estimating gradients is via finite differences, which requires evaluating the plant outputs at several (perturbed) operating points. The required number of perturbed points depends on the number of inputs and, as a consequence, the experimental cost of gradient estimation increases with increasing input dimension.

In the past years, several methods have been proposed to obtain gradient information. In dual MA (Marchetti et al., 2010), one considers an additional constraint in the RTO problem, which restricts the location of the next RTO inputs such that reliable gradient information can be extracted using the current and previously visited operating points. Dual ISOPE (Brdyś and Tatjewski, 2005) and

[*] Corresponding author.
E-mail addresses: martand.singhal@epfl.ch (M. Singhal), marchetti@cifasis-conicet.gov.ar (A.G. Marchetti), timm.faulwasser@kit.edu (T. Faulwasser), dominique.bonvin@epfl.ch (D. Bonvin).
[1] Current address: French-Argentine International Center for Information and Systems Sciences (CIFASIS), CONICET-Universidad Nacional de Rosario (UNR), S2000EZP Rosario, Argentina.

https://doi.org/10.1016/j.compchemeng.2018.02.016
0098-1354/© 2018 Elsevier Ltd. All rights reserved.

the approach proposed by Rodger and Chachuat (2011) also make use of ‘duality constraints’ so as to simultaneously estimate gradients and optimize the plant. Recently, Gao et al. (2016) proposed to combine a quadratic approximation used in derivative-free optimization (Conn et al., 2009) and MA to improve the quality of gradient estimates in the presence of noise. Alternatively, instead of estimating gradients, one can attempt to directly compute the first-order correction terms using an additional optimization layer as proposed by Navia et al. (2015). We refer to Marchetti et al. (2016) and the references therein for a detailed literature overview on MA.

Recently, Costello et al. (2016) proposed a MA approach that reduces the burden of gradient estimation by questioning the necessity of correcting in all input directions. The approach, labeled directional modifier adaptation (DMA), proposes to correct the model gradients only in ‘privileged’ directions that span a reduced subspace of the input space. This subspace is computed once offline by means of a local sensitivity analysis conducted on the gradient of the Lagrangian function predicted by the model. The sensitivities are evaluated with respect to variations around the nominal values of the model parameters.

In this paper, we extend the concept of DMA to cover the case where the parametric uncertainty is not local, but belongs to a fairly large uncertainty set. In this case, we argue that correcting the gradients only in the privileged directions identified offline via local sensitivity analysis may result in significant sub-optimality. Instead, we propose here to perform a global sensitivity analysis using ideas derived from *active subspaces* (Constantine, 2015; Russi, 2010). The concept of active subspaces has emerged as a set of techniques for reducing the dimension of the input space. Similar ideas are used in this paper to develop an *active* directional modifier-adaptation (ADMA) algorithm.

The contribution of this paper is in establishing the theoretical foundations of ADMA via the concepts derived from active subspaces. We extend our preliminary work described in Singhal et al. (2017) by providing a formal analysis of optimality upon convergence in ADMA. We discuss the practical aspects of ADMA and we demonstrate the effectiveness of the algorithm for the run-to-run optimization of two different semi-batch reactors.

The paper is structured as follows. Preliminary material including the formulation of the optimization problem, the description of the MA and DMA schemes, and background elements from active subspace theory, are presented in Section 2. The novel RTO approach that deals with large parametric uncertainty is then proposed in Section 3. In Section 4, two case studies dealing with semi-batch reactors are presented. The first case study considers only parametric uncertainty, while the second study deals with structural plant-model mismatch. We conclude the paper in Section 5.

2. Preliminaries

2.1. Problem formulation

The plant optimization problem can be written mathematically as:

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

$$\text{s.t.} \quad \mathrm{G}_{p,i}(\mathbf{u}) := \mathrm{g}_i(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \le 0, \quad i = 1, \dots, n_\mathrm{g}, \tag{1b}$$

where $\mathbf{u} \in \mathbb{R}^{n_\mathrm{u}}$ is the vector of input variables, $\mathbf{y}_p \in \mathbb{R}^{n_\mathrm{y}}$ are the measured output variables, ϕ: $\mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}$ is the cost to be minimized, g_i: $\mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{y}} \to \mathbb{R}$, $i = 1, \dots, n_\mathrm{g}$, are the inequality constraints. The solution to Problem (1) is denoted $\mathbf{u}_p^\star$.

The main challenge in solving this optimization problem stems from the fact that the input-output mapping $\mathbf{y}_p(\mathbf{u})$ is unknown. However, an approximate process model is assumed to be available, which gives the input-output mapping $\mathbf{y}(\mathbf{u}, \theta)$, where $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ are the model parameters. Then, using the model, Problem (1) can be approximated as:

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

$$\text{s.t.} \quad \mathrm{G}_i(\mathbf{u}, \boldsymbol{\theta}) := \mathrm{g}_i(\mathbf{u}, \mathbf{y}(\mathbf{u}, \boldsymbol{\theta})) \le 0, \quad i = 1, \dots, n_\mathrm{g}. \tag{2b}$$

The nominal solution $\mathbf{u}^\star$ is found by solving Problem (2) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$, where $\boldsymbol{\theta}_0$ is the vector of nominal model parameters. In the presence of plant-model mismatch, the model optimum $\mathbf{u}^\star$ may not be equal to the plant optimum $\mathbf{u}_p^\star$. The goal of RTO is to find $\mathbf{u}_p^\star$ by iteratively modifying and solving Problem (2).

2.2. Modifier adaptation

Modifier adaptation introduces first-order correction terms that are added to the cost and constraint functions predicted by the nominal model. At the kth RTO iteration, the next inputs are computed by solving the following *modified* optimization problem (Marchetti, 2009):

$$\min_{\mathbf{u}} \quad \Phi_{m,k}(\mathbf{u}) := \Phi(\mathbf{u}, \boldsymbol{\theta}) + (\boldsymbol{\lambda}_k^{\Phi})^\mathsf{T}\mathbf{u} \tag{3a}$$

$$\text{s.t.} \quad \mathbf{G}_{m,k}(\mathbf{u}) := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\varepsilon}_k^{\mathrm{G}} + (\boldsymbol{\lambda}_k^{\mathrm{G}})^\mathsf{T}(\mathbf{u} - \mathbf{u}_k) \preceq \mathbf{0}, \tag{3b}$$

where $\mathbf{G} \in \mathbb{R}^{n_\mathrm{g}}$ is the vector of constraints G_i, $i = 1, \dots, n_\mathrm{g}$; $\boldsymbol{\varepsilon}_k^{\mathrm{G}} \in \mathbb{R}^{n_\mathrm{g}}$ is the vector of zeroth-order modifiers for the constraints; and $\boldsymbol{\lambda}_k^{\Phi} \in \mathbb{R}^{n_\mathrm{u}}$ and $\boldsymbol{\lambda}_k^{\mathrm{G}} \in \mathbb{R}^{n_\mathrm{u} \times n_\mathrm{g}}$ are the first-order modifiers for the cost and constraint functions, respectively. At the kth RTO iteration, the modifiers are computed as follows:

$$\boldsymbol{\varepsilon}_k^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}), \tag{4a}$$

$$(\boldsymbol{\lambda}_k^{\Phi})^\mathsf{T} = \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_k, \boldsymbol{\theta}), \tag{4b}$$

$$(\boldsymbol{\lambda}_k^{\mathrm{G}})^\mathsf{T} = \nabla_{\mathbf{u}}\mathbf{G}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}), \tag{4c}$$

where $\nabla_{\mathbf{u}}(\cdot)$ is the gradient of a scalar-valued function or the Jacobian of a vector-valued function with respect to $\mathbf{u}$. MA guarantees meeting the plant KKT conditions of Problem (1) upon convergence (Marchetti et al., 2009). Gradient adaptation via first-order modifiers plays a key role in meeting the plant KKT conditions. However, finding reliable plant gradients is a costly task as it requires additional plant evaluations. If, for instance, the forward finite-difference approach is used, then the number of plant evaluations at each RTO iteration increases linearly with the dimension of the input space.

2.3. Directional modifier adaptation

The dependency of MA on the knowledge of full plant gradients can be reduced with the help of a process model. As the model gradients are sensitive to model parameters, the input subspace in which the parametric uncertainty has the most influence on the solution to Problem (2) can be found via *local* sensitivity analysis. In Costello et al. (2016), this subspace is spanned by the so-called ‘privileged directions’ for the purpose of gradient estimation. DMA evaluates offline the sensitivity of the model Lagrangian gradient with respect to local parametric variations that are evaluated at the model optimum. To this end, the model Lagrangian function is defined as

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

with $\boldsymbol{\mu} \in \mathbb{R}^{n_\mathrm{g}}$ the vector of Lagrange multipliers. Then, the sensitivity matrix $\boldsymbol{A}^\star \in \mathbb{R}^{n_\mathrm{u} \times n_\theta}$ is computed as follows:

$$\boldsymbol{A}^\star := \nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}(\mathbf{u}^\star, \boldsymbol{\mu}^\star, \boldsymbol{\theta}_0) = \left.\frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}}\right|_{\mathbf{u}^\star, \boldsymbol{\mu}^\star, \boldsymbol{\theta}_0}, \tag{6}$$

where $\boldsymbol{\mu}^\star$ are the Lagrange multipliers corresponding to the nominal solution $\mathbf{u}^\star$. Singular value decomposition of $\boldsymbol{A}^\star$ gives:

$$\boldsymbol{A}^\star = \boldsymbol{W}\boldsymbol{S}\boldsymbol{V}^\top, \tag{7}$$

where $\boldsymbol{W} \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{u}}$ is an orthonormal matrix whose columns $\mathbf{w}_i$, $i = 1, \ldots, n_\mathrm{u}$, are the left singular vectors of $\boldsymbol{A}^\star$; $\boldsymbol{S} \in \mathbb{R}^{n_\mathrm{u}\times n_\theta}$ is a rectangular diagonal matrix whose diagonal elements s_i, $i = 1, \ldots, n_\mathrm{s}$ with $n_\mathrm{s} = \min\{n_\mathrm{u}, n_\theta\}$, are the singular values of $\boldsymbol{A}^\star$; and $\boldsymbol{V} \in \mathbb{R}^{n_\theta\times n_\theta}$. Through the singular values of $\boldsymbol{A}^\star$, one can rank the input directions $\mathbf{w}_i$ according to their sensitivity with respect to local parametric perturbations. The reduced matrix $\boldsymbol{W}_\mathrm{r} \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{r}}$, with $n_\mathrm{r} < n_\mathrm{u}$, can be constructed as:

$$\boldsymbol{W}_\mathrm{r} = [\mathbf{w}_1 \ \ldots \ \mathbf{w}_{n_\mathrm{r}}] : \quad \mathrm{s}_{n_\mathrm{r}+1} \ll \mathrm{s}_{n_\mathrm{r}}, \tag{8}$$

i.e., a large gap between the consecutive singular values is exploited to construct $\boldsymbol{W}_\mathrm{r}$. At each RTO iteration, the directional derivatives are estimated only in the *privileged* directions spanned by the columns of $\boldsymbol{W}_\mathrm{r}$. Note that the number of privileged directions for DMA satisfies the following condition:

$$n_\mathrm{r} \le \min\{n_\mathrm{u}, n_\theta\}. \tag{9}$$

Assuming that Φ_p is differentiable at $\mathbf{u}$, the directional derivative of Φ_p in any direction $\mathbf{r}$ contained in the input subspace is defined as

$$\nabla_{\boldsymbol{W}_\mathrm{r}}\Phi_p(\mathbf{u}) := \left.\frac{\partial \Phi_p(\mathbf{u} + \boldsymbol{W}_\mathrm{r}\mathbf{r})}{\partial \mathbf{r}}\right|_{\mathbf{r}=\mathbf{0}}, \tag{10}$$

with $\nabla_{\boldsymbol{W}_\mathrm{r}}\Phi_p \in \mathbb{R}^{1\times n_\mathrm{r}}$, and $\mathbf{r} \in \mathbb{R}^{n_\mathrm{r}}$. Note that

$$\nabla_{\boldsymbol{W}_\mathrm{r}}\Phi_p(\mathbf{u}) = \nabla_\mathbf{u}\Phi_p(\mathbf{u})\boldsymbol{W}_\mathrm{r}. \tag{11}$$

The directional derivatives for the plant constraints $\mathrm{G}_{p,\,i}$, $i = 1, \ldots, n_g$, are defined in similar fashion. These derivatives can be estimated by forward finite differences or using a duality constraint as done in dual modifier adaptation (Marchetti et al., 2010). The DMA scheme is summarized in Algorithm 1.

Past studies have shown a significant reduction in the experimental cost of gradient estimation when DMA is applied. For instance, DMA has been applied to perform RTO on an airbone-wind energy system (Costello et al., 2016; 2015). Therein, DMA significantly reduces the input space from 40 to 2 dimensions for the purpose of gradient estimation. Yet, the optimality loss is only 5 percent despite adapting the gradients in only two directions (in the other 38 directions, nominal model gradients are used).

Nevertheless, by no means can it be expected that a local sensitivity analysis will systematically yield a good approximation to global sensitivities. When this is not the case, then adapting the gradients in the privileged directions found offline by DMA may result in significant optimality loss. In order to address this issue, we propose an *online* procedure for determining the privileged directions via a *global* sensitivity analysis carried out at each RTO iteration.

2.4. Mathematical preliminaries

We present a few mathematical tools that are inspired by active subspace theory (Constantine, 2015; Russi, 2010).

Consider a twice differentiable function $f : \mathcal{U} \times \Theta \to \mathbb{R}$, where $\mathcal{U} \subseteq \mathbb{R}^{n_\mathrm{u}}$, $\Theta \subseteq \mathbb{R}^{n_\theta}$ and Θ is a bounded and connected set. Let the probability density function of $\boldsymbol{\theta}$ be $\rho(\boldsymbol{\theta})$. Also, consider that $\rho(\boldsymbol{\theta})$ is strictly positive and bounded for $\boldsymbol{\theta} \in \Theta$ and $\rho(\boldsymbol{\theta}) = 0$ for $\boldsymbol{\theta} \notin \Theta$, so that the focus is only on the parameter values of interest. Assume that ρ and Θ are such that the components of $\boldsymbol{\theta}$ are independent with mean zero and scaled according to their range. Such a normalization ensures that each parameter component is given equal importance. In addition, assume that the matrix $\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}, \boldsymbol{\theta}) := \frac{\partial^2 f(\mathbf{u},\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\,\partial\mathbf{u}} \in \mathbb{R}^{n_\mathrm{u}\times n_\theta}$ is bounded, that is,

$$||\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}, \boldsymbol{\theta})|| \le L, \quad L > 0 \qquad \forall\ \mathbf{u} \in \mathcal{U},\ \boldsymbol{\theta} \in \Theta,$$

where $||\cdot||$ is the Frobenius norm.

Next, we define the matrix $\boldsymbol{A}_k \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{u}}$ as:

$$\boldsymbol{A}_k = \int_\Theta \big(\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)\big(\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)^\top \rho\, \mathrm{d}\boldsymbol{\theta}. \tag{12}$$

It follows that each element of $\boldsymbol{A}_k$ is the average of the product of partial double derivatives (which we assume exist):

$$\mathrm{a}_{ij,k} = \int_\Theta \sum_{l=1}^{n_\theta} \left(\left.\frac{\partial^2 f(\mathbf{u},\boldsymbol{\theta})}{\partial\theta_l\,\partial u_i}\right|_{\mathbf{u}_k}\right)\left(\left.\frac{\partial^2 f(\mathbf{u},\boldsymbol{\theta})}{\partial\theta_l\,\partial u_j}\right|_{\mathbf{u}_k}\right)\rho\, \mathrm{d}\boldsymbol{\theta},$$
$$i, j = 1, \ldots, n_\mathrm{u}, \tag{13}$$

where $\mathrm{a}_{ij,\,k}$ is the (i, j) element of $\boldsymbol{A}_k$; θ_l is the lth element of $\boldsymbol{\theta}$; and u_i is the ith element of $\mathbf{u}$. The matrix $\boldsymbol{A}_k$ is positive semi-definite since

$$\boldsymbol{\nu}^\top\boldsymbol{A}_k\,\boldsymbol{\nu} = \int_\Theta \big(\boldsymbol{\nu}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)\big(\boldsymbol{\nu}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)^\top \rho\, \mathrm{d}\boldsymbol{\theta}$$
$$\ge 0 \quad \forall\,\boldsymbol{\nu} \in \mathbb{R}^{n_\mathrm{u}}.$$

Moreover, as $\boldsymbol{A}_k$ is symmetric, we can write:

$$\boldsymbol{A}_k = \boldsymbol{W}_k\,\boldsymbol{\Sigma}_k\boldsymbol{W}_k^\top, \quad \boldsymbol{\Sigma}_k = \mathrm{diag}(\sigma_{1,k}, \ldots, \sigma_{n_\mathrm{u},k}),$$
$$\sigma_{1,k} \ge \cdots \ge \sigma_{n_\mathrm{u},k} \ge 0, \tag{14}$$

where $\boldsymbol{W}_k \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{u}}$ is an orthonormal matrix whose columns $\mathbf{w}_{i,\,k}$, $i = 1, \ldots, n_\mathrm{u}$, are the normalized eigenvectors of $\boldsymbol{A}_k$.

Lemma 2.1. *For all $\mathbf{u}_k \in \mathcal{U}$, it holds that*

$$\int_\Theta \left|\left|\mathbf{w}_{i,k}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\right|\right|^2 \rho\, \mathrm{d}\boldsymbol{\theta} = \sigma_{i,k}, \quad i = 1, \ldots, n_\mathrm{u}, \tag{15}$$

where $\sigma_{i,\,k}$ is the eigenvalue corresponding to the eigenvector $\mathbf{w}_{i,\,k}$ of $\boldsymbol{A}_k$.

Proof. The definition of $\sigma_{i,\,k}$ implies

$$\sigma_{i,k} = \mathbf{w}_{i,k}^\top\boldsymbol{A}_k\,\mathbf{w}_{i,k},$$

which can be written as:

$$\sigma_{i,k} = \mathbf{w}_{i,k}^\top\left(\int_\Theta \big(\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)\big(\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)^\top \rho\, \mathrm{d}\boldsymbol{\theta}\right)\mathbf{w}_{i,k}$$
$$= \int_\Theta \big(\mathbf{w}_{i,k}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)\big(\mathbf{w}_{i,k}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\big)^\top \rho\, \mathrm{d}\boldsymbol{\theta}$$
$$= \int_\Theta \left|\left|\mathbf{w}_{i,k}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta})\right|\right|^2 \rho\, \mathrm{d}\boldsymbol{\theta}.$$

□

It follows from this lemma that, if the eigenvalue $\sigma_{i,k} = 0$, then

$$\mathbf{w}_{i,k}^\top\nabla_{\mathbf{u}\boldsymbol{\theta}}f(\mathbf{u}_k, \boldsymbol{\theta}) = \mathbf{0}, \quad \forall\,\boldsymbol{\theta} \in \Theta. \tag{16}$$

Integrating (16) with respect to $\boldsymbol{\theta}$, and using the fundamental theorem of calculus, gives

$$\sigma_{i,k} = 0, \Rightarrow \nabla_\mathbf{u}f(\mathbf{u}_k, \boldsymbol{\theta})\,\mathbf{w}_{i,k} = \mathrm{c}, \quad \mathrm{c} \in \mathbb{R}, \quad \forall\,\boldsymbol{\theta} \in \Theta. \tag{17}$$

In other words, the lemma implies that the directional derivative of f (with respect to $\mathbf{u}$ at $\mathbf{u}_k$) in the direction $\mathbf{w}_{i,\,k}$ is constant regardless of the value of the parameter $\boldsymbol{\theta}$ (as long as $\boldsymbol{\theta} \in \Theta$).

The matrix $\boldsymbol{W}_k$ can be split into two submatrices, the matrix $\boldsymbol{W}_{1,k} \in \mathbb{R}^{n_\mathrm{u}\times n_\mathrm{r}}$ and the matrix $\boldsymbol{W}_{2,k} \in \mathbb{R}^{n_\mathrm{u}\times(n_\mathrm{u}-n_\mathrm{r})}$, whereby $\boldsymbol{W}_{1,\,k}$ contains the eigenvectors $\mathbf{w}_{i,\,k}$ corresponding to the n_r non-zero eigenvalues and the matrix $\boldsymbol{W}_{2,\,k}$ collects the remaining eigenvectors corresponding to the zero eigenvalues:

$$\boldsymbol{W}_k = \big[\boldsymbol{W}_{1,k}\ \boldsymbol{W}_{2,k}\big],$$
$$\boldsymbol{W}_{1,k} = [\mathbf{w}_{1,k} \ \ldots \ \mathbf{w}_{n_\mathrm{r},k}] : \sigma_{1,k} \ge \cdots \ge \sigma_{n_\mathrm{r},k} > 0, \quad n_\mathrm{r} \le n_\mathrm{u}, \tag{18}$$
$$\boldsymbol{W}_{2,k} = [\mathbf{w}_{n_\mathrm{r}+1,k} \ \ldots \ \mathbf{w}_{n_\mathrm{u},k}] : \sigma_{n_\mathrm{r}+1,k} = \cdots = \sigma_{n_\mathrm{u},k} = 0.$$

Such a construction will let us show that, if at each RTO iteration the cost and constraint gradients of the model are adapted in the directions corresponding to the matrix $\boldsymbol{W}_{1,k}$, then the plant KKT point is reached upon convergence.

3. Active directional modifier adaptation

Privileged directions should ideally be chosen such that they capture the maximum variability of the Lagrangian gradient with respect to parametric perturbations. As parametric perturbations get large, the local sensitivity analysis conducted in DMA may not be able to yield such directions. Therefore, we propose to find the set of privileged directions based on the following global sensitivity matrix $\boldsymbol{A}_k$:

$$\boldsymbol{A}_k = \int_{\Theta} \left(\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}(\mathbf{u}_k,\boldsymbol{\mu}_k,\boldsymbol{\theta})\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}(\mathbf{u}_k,\boldsymbol{\mu}_k,\boldsymbol{\theta})\right)^{\mathsf{T}} \rho \, d\boldsymbol{\theta}. \tag{19}$$

Then, the resulting privileged direction matrix $\boldsymbol{W}_{1,\,k}$ is used to update the modifiers as follows :

$$\boldsymbol{\varepsilon}_k^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}_0), \tag{20a}$$

$$(\boldsymbol{\lambda}_k^{\Phi})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_k,\boldsymbol{\theta}_0), \tag{20b}$$

$$(\boldsymbol{\lambda}_k^{\mathrm{G}})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\mathbf{G}}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta}_0), \tag{20c}$$

where the gradients $\widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_k)$ and $\widehat{\nabla_{\mathbf{u}}\mathbf{G}}_p(\mathbf{u}_k)$ are updated as:

$$\widehat{\nabla_{\mathbf{u}}\Xi}_p(\mathbf{u}_k) = \nabla_{\mathbf{u}}\Xi(\mathbf{u}_k,\boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,k}\boldsymbol{W}_{1,k}^{+}) + \nabla_{\boldsymbol{W}_{1,k}}\Xi_p(\mathbf{u}_k)\boldsymbol{W}_{1,k}^{+}, \quad \Xi \in \{\Phi,\, \mathrm{G}_i\}, \tag{21}$$

where $\nabla_{\boldsymbol{W}_{1,k}}\Xi_p(\mathbf{u}_k)$ is the directional derivative of Ξ_p at $\mathbf{u}_k$, and $\boldsymbol{W}_{1,k}^{+}$ is the Moore–Penrose pseudo-inverse of $\boldsymbol{W}_{1,\,k}$. For the sake of simplicity, the filter matrices used in Algorithm 1 are dropped from the modifiers in (20). Updating the modifiers using first-order filters does not affect the validity of the results presented hereafter.

3.1. KKT matching under parametric plant-model mismatch

Our further developments are based on the following technical assumptions.

Assumption 1 (Parametric plant-model mismatch)**.** Let $\boldsymbol{\theta}_p \in \Theta$ be such that

$$\Phi(\mathbf{u},\boldsymbol{\theta}_p) = \Phi_p(\mathbf{u}), \tag{22a}$$

$$\mathbf{G}(\mathbf{u},\boldsymbol{\theta}_p) = \mathbf{G}_p(\mathbf{u}), \tag{22b}$$

$\boldsymbol{\theta}_p$ is the vector of plant parameters, and Θ is a bounded and connected set in which $\boldsymbol{\theta}$ lies with the probability density function ρ.

Assumption 2 (Exact plant directional derivatives)**.** At each RTO iteration k, exact plant directional derivatives are available for the cost and constraint functions in the directions given by the columns of the matrix $\boldsymbol{W}_{1,\,k}$.

Assumption 3 (Exact sensitivity information)**.** The matrix $\boldsymbol{A}_k$ in (19) is perfectly known at each RTO iteration k.

Theorem 3.1 (Optimality upon convergence)**.** *Consider the optimization Problem (3) with the modifiers (20) and the gradient updates (21) with $\boldsymbol{W}_{1,\,k}$ satisfying (18). Let Assumptions 1–3 hold. Also, assume that $\boldsymbol{\theta}_0 \in \Theta$. If the iterative solution to this problem converges to the fixed point $(\mathbf{u}_\infty,\, \boldsymbol{\varepsilon}_\infty^{\mathrm{G}},\, \boldsymbol{\lambda}_\infty^{\Phi},\, \boldsymbol{\lambda}_\infty^{\mathrm{G}})$, with $\mathbf{u}_\infty$ being a KKT point of Problem (3), then $\mathbf{u}_\infty$ is also a KKT point for the plant Problem (1).*

Proof. The modifiers take the following values upon convergence to $\mathbf{u}_\infty$:

$$\boldsymbol{\varepsilon}_\infty^{\mathrm{G}} = \mathbf{G}_p(\mathbf{u}_\infty) - \mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta}_0), \tag{23a}$$

$$(\boldsymbol{\lambda}_\infty^{\Phi})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0), \tag{23b}$$

$$(\boldsymbol{\lambda}_\infty^{\mathrm{G}})^{\mathsf{T}} = \widehat{\nabla_{\mathbf{u}}\mathbf{G}}_p(\mathbf{u}_\infty) - \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta}_0). \tag{23c}$$

The KKT conditions at $\mathbf{u}_\infty$ for Problem (3) read:

$$\mathbf{G}_{m,\infty}(\mathbf{u}_\infty) \preceq \mathbf{0}, \tag{24a}$$

$$\boldsymbol{\mu}_\infty^{\mathsf{T}}\mathbf{G}_{m,\infty}(\mathbf{u}_\infty) = 0, \quad \boldsymbol{\mu}_\infty \succeq \mathbf{0}, \tag{24b}$$

$$\nabla_{\mathbf{u}}\Phi_{m,\infty}(\mathbf{u}_\infty) + \boldsymbol{\mu}_\infty\nabla_{\mathbf{u}}\mathbf{G}_{m,\infty}(\mathbf{u}_\infty) = \mathbf{0}. \tag{24c}$$

From (3), (23) and (24), one can write:

$$\mathbf{G}_p(\mathbf{u}_\infty) \preceq \mathbf{0}, \tag{25a}$$

$$\boldsymbol{\mu}_\infty^{\mathsf{T}}\mathbf{G}_p(\mathbf{u}_\infty) = 0, \quad \boldsymbol{\mu}_\infty \succeq \mathbf{0}, \tag{25b}$$

$$\widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) + \boldsymbol{\mu}_\infty^{\mathsf{T}}\widehat{\nabla_{\mathbf{u}}\mathbf{G}}_p(\mathbf{u}_\infty) = \mathbf{0}. \tag{25c}$$

Next, consider (21) at $\mathbf{u}_\infty$:

$$\begin{aligned}\widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}) \\ &\quad + \nabla_{\boldsymbol{W}_{1,\infty}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}^{+}.\end{aligned}$$

It follows from (11) that

$$\begin{aligned}\widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}) \\ &\quad + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}, \\ \widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0)(\boldsymbol{W}_{\infty}\boldsymbol{W}_{\infty}^{\mathsf{T}} - \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}) \\ &\quad + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}, \\ \widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0)(\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{\mathsf{T}} + \boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^{\mathsf{T}} \\ &\quad - \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}) + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{+}.\end{aligned}$$

Since the matrix $\boldsymbol{W}_{1,\,\infty}$ has orthonormal columns, $\boldsymbol{W}_{1,\infty}^{+} = \boldsymbol{W}_{1,\infty}^{\mathsf{T}}$, and

$$\begin{aligned}\widehat{\nabla_{\mathbf{u}}\Phi}_p(\mathbf{u}_\infty) &= \nabla_{\mathbf{u}}\Phi(\mathbf{u}_\infty,\boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^{\mathsf{T}} \\ &\quad + \nabla_{\mathbf{u}}\Phi_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{\mathsf{T}}.\end{aligned} \tag{26}$$

Similarly, for the constraints, one can write:

$$\widehat{\nabla_{\mathbf{u}}\mathbf{G}}_p(\mathbf{u}_\infty) = \nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^{\mathsf{T}} + \nabla_{\mathbf{u}}\mathbf{G}_p(\mathbf{u}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{\mathsf{T}}. \tag{27}$$

Using (26), (27) and (25c) gives:

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^{\mathsf{T}} + \nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)\boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{\mathsf{T}} = \mathbf{0}, \tag{28}$$

where $\mathcal{L}_p(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty) = \Phi_p(\mathbf{u}_\infty) + \boldsymbol{\mu}_\infty^{\mathsf{T}}\mathbf{G}_p(\mathbf{u}_\infty)$.

We know from Lemma 2.1 that the directional derivatives of the Lagrangian are constant in the directions given by the columns of the matrix $\boldsymbol{W}_{2,\,\infty}$, since the corresponding eigenvalues are zero. We also know that these directional derivatives are independent of the parameter values $\boldsymbol{\theta} \in \Theta$. Hence,

$$\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\theta}_0)\boldsymbol{W}_{2,\infty} = \nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\theta}_p)\boldsymbol{W}_{2,\infty} = \mathbf{c} \in \mathbb{R}^{1\times(n_{\mathrm{u}}-n_{\mathrm{r}})}. \tag{29}$$

It follows from (28), (29) and Assumption 1 that

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)(\boldsymbol{W}_{2,\infty}\boldsymbol{W}_{2,\infty}^{\mathsf{T}} + \boldsymbol{W}_{1,\infty}\boldsymbol{W}_{1,\infty}^{\mathsf{T}}) = \mathbf{0}, \tag{30a}$$

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)(\boldsymbol{W}_{\infty}\boldsymbol{W}_{\infty}^{\mathsf{T}}) = \mathbf{0}, \tag{30b}$$

$$\nabla_{\mathbf{u}}\mathcal{L}_p(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty) = \mathbf{0}. \tag{30c}$$

Then, from Equations (25a), (25b) and (30c), we conclude that $\mathbf{u}_\infty$ is a KKT point of the plant Problem (1). □

Remark 1. An implicit assumption in Theorem 3.1 is that the matrix $\boldsymbol{A}_k$ has at least one eigenvalue equal to zero. However, most often in practice, none of the eigenvalues is exactly zero; instead, some eigenvalues are small compared to others and can be discarded without much information loss. Note that, if the matrix $\boldsymbol{A}_k$ has no eigenvalues that can be discarded, the partitioning (18) gives $\boldsymbol{W}_{1,k} = \boldsymbol{W}_k$, and ADMA reduces to standard MA.

Algorithm 1 Directional Modifier Adaptation (DMA)

```
Step 0 (Initialization): Compute the nominal solution $\mathbf{u}^\star$ and the corresponding Lagrange multipliers $\boldsymbol{\mu}^\star$ by solving Problem (2) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$. Evaluate the sensitivity matrix $\boldsymbol{A}^\star$ in (6), perform singular value decomposition and determine the privileged directions $\boldsymbol{W}_{\mathrm{r}}$.
    Set the initial values of the modifiers to zero, $\boldsymbol{\varepsilon}_0^{\mathrm{G}} = \mathbf{0}$, $\boldsymbol{\lambda}_0^{\Phi} = \mathbf{0}$ and $\boldsymbol{\lambda}_0^{\mathrm{G}_i} = \mathbf{0}$, and the values of the filter matrices $\boldsymbol{K}^{\boldsymbol{\varepsilon}}$, $\boldsymbol{K}^{\Phi}$ and $\boldsymbol{K}^{\mathrm{G}_i}$ (typically diagonal matrices) with eigenvalues in the interval $(0, 1]$. Also, set arbitrarily $\mathbf{u}_0 = \mathbf{0}$.

for $k = 0 \to \infty$.

Step 1 (Optimization): Solve the modified model-based Problem (3) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ to compute the optimal inputs $\mathbf{u}_{k+1}$.

Step 2 (Plant evaluation): Apply $\mathbf{u}_{k+1}$ to the plant and collect the measurements $\mathbf{y}_p(\mathbf{u}_{k+1})$.
    Use these measurements to compute $\Phi_p(\mathbf{u}_{k+1})$ and $\mathbf{G}_p(\mathbf{u}_{k+1})$.

Step 3 (Estimation of directional derivatives): Estimate the directional derivative of the plant cost $\nabla_{\boldsymbol{W}_{\mathrm{r}}}\Phi_p(\mathbf{u}_{k+1})$ and of the constraints $\nabla_{\boldsymbol{W}_{\mathrm{r}}}\mathrm{G}_{p,i}(\mathbf{u}_{k+1})$, $i = 1, \ldots, n_{\mathrm{g}}$, as per (6)-(10). At $\mathbf{u}_{k+1}$, the full gradients are computed as:

    $$\widehat{\nabla_{\mathbf{u}}\Xi_p}(\mathbf{u}_{k+1}) := \nabla_{\mathbf{u}}\Xi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big(\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{W}_{\mathrm{r}}\boldsymbol{W}_{\mathrm{r}}^{+}\big) + \nabla_{\boldsymbol{W}_{\mathrm{r}}}\Xi_p(\mathbf{u}_{k+1})\boldsymbol{W}_{\mathrm{r}}^{+},$$

    with $\Xi \in \{\Phi, \mathrm{G}_i\}$, and $\boldsymbol{W}_{\mathrm{r}}^{+}$ the Moore-Penrose pseudo-inverse of $\boldsymbol{W}_{\mathrm{r}}$.

Step 4 (Modifier update): Update the modifiers using first-order filters:

    $$\boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}} = (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\boldsymbol{\varepsilon}})\boldsymbol{\varepsilon}_k^{\mathrm{G}} + \boldsymbol{K}^{\boldsymbol{\varepsilon}}\big(\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big),$$
    $$\boldsymbol{\lambda}_{k+1}^{\Phi} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\Phi})\boldsymbol{\lambda}_k^{\Phi} + \boldsymbol{K}^{\Phi}\big(\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}},$$
    $$\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\mathrm{G}_i})\boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i}\big(\widehat{\nabla_{\mathbf{u}}\mathrm{G}_{p,i}}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\big)^{\mathsf{T}}, \quad i = 1, \ldots, n_{\mathrm{g}}.$$

end
```

3.2. Practical aspects of ADMA

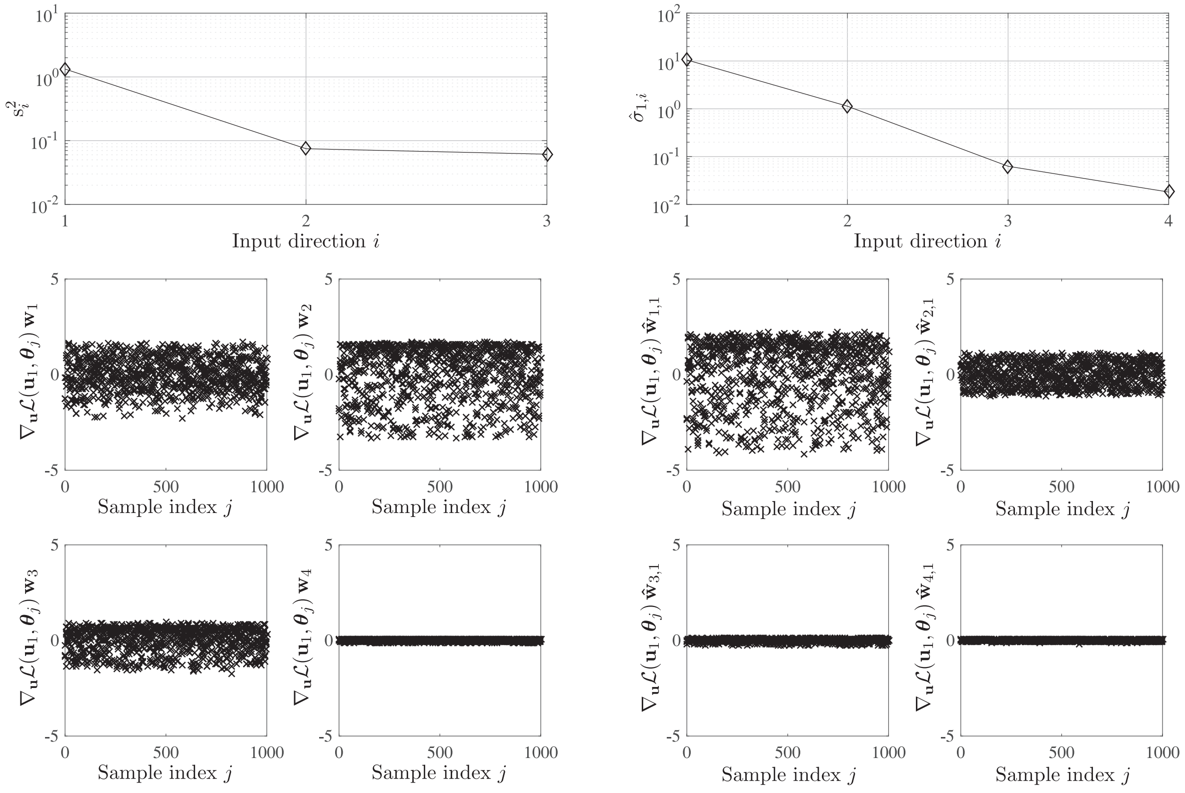
ADMA requires the knowledge of $\boldsymbol{A}_k$ and $\boldsymbol{W}_k$, which are approximated as the estimates $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$.

3.2.1. Computation of $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$

The available process models are often too complex for allowing the derivation of analytical expressions for the matrices $\boldsymbol{A}_k$ and $\boldsymbol{W}_k$. For these cases, we propose to estimate these matrices from data using a sampling-based Monte-Carlo approach. It is recommended to scale the inputs and parameters so that they lie between −1 and 1. The estimates $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$ are computed via the following steps:[2]

The sample size N should be chosen such that increasing N has a negligible effect on the eigenvalues of the matrix $\hat{\boldsymbol{A}}_k$.

[2] Note that similar steps have been proposed for computing active subspaces in Constantine (2015).



(a) **Top plot:** Squared singular values of matrix $\boldsymbol{A}^{\star}$. **Middle and bottom plots:** Directional derivatives computed at $\boldsymbol{\theta}_j$, $j = 1, \ldots, N$, along the left singular vectors of $\boldsymbol{A}^{\star}$. The plotted data is mean centered.

(b) **Top plot:** Eigenvalues of the matrix $\hat{\boldsymbol{A}}_1$. **Middle and bottom plots:** Directional derivatives computed at $\boldsymbol{\theta}_j$, $j = 1, \ldots, N$, along the eigenvectors of $\hat{\boldsymbol{A}}_1$. The plotted data is mean centered.

Fig. 1. Comparison of the sensitivity matrices $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}_1$ corresponding to DMA and ADMA, respectively.

3.2.2. Formal difference between DMA and ADMA

An alternative approach based on singular value decomposition (SVD) can be used to compute the eigenpairs of $\hat{\boldsymbol{A}}_k$. Note that we can write $\hat{\boldsymbol{A}}_k = \hat{\boldsymbol{B}}_k\hat{\boldsymbol{B}}_k^{\mathsf{T}}$, where the matrix $\hat{\boldsymbol{B}}_k \in \mathbb{R}^{n_{\mathrm{u}} \times n_{\theta} N}$ is

$$\hat{\boldsymbol{B}}_k = \frac{1}{\sqrt{N}}\left[\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}_k^{(1)} \cdots \nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}_k^{(N)}\right]. \tag{32}$$

Applying SVD to $\hat{\boldsymbol{B}}_k$ and using the well-known relation of SVD to eigenvalue decomposition gives:

$$\hat{\boldsymbol{B}}_k = \hat{\boldsymbol{W}}_k\hat{\boldsymbol{S}}_k\hat{\boldsymbol{V}}_k^{\mathsf{T}}, \quad \text{with } \hat{\boldsymbol{S}}_k\hat{\boldsymbol{S}}_k^{\mathsf{T}} = \hat{\boldsymbol{\Sigma}}_k. \tag{33}$$

This allows comparing the SVDs performed in DMA and in ADMA. In the former, SVD is performed on the sensitivity matrix $\boldsymbol{A}^{\star}$ that is evaluated for the nominal parameters $\boldsymbol{\theta}_0$. In the latter, SVD is performed on the matrix $\hat{\boldsymbol{B}}_k$ that stacks the local sensitivity matrices evaluated at N randomly chosen realizations of the parametric uncertainty into a single matrix, thereby representing global sensitivity.

The sensitivity matrix $\boldsymbol{A}^{\star}$ is local in both the inputs $\mathbf{u}$ and the parameters $\boldsymbol{\theta}$. In contrast, the sensitivity matrix $\hat{\boldsymbol{A}}_k$ (or $\hat{\boldsymbol{B}}_k$) is local in the inputs $\mathbf{u}$ but global in the parameters $\boldsymbol{\theta}$. If the model Lagrangian is linear in the parameters, then the sensitivity matrices $\boldsymbol{A}^{\star}$ and $\hat{\boldsymbol{A}}_k$ are equal when computed for the same inputs, that is, when computed at $(\mathbf{u}^{\star}, \boldsymbol{\mu}^{\star}) = (\mathbf{u}_k, \boldsymbol{\mu}_k)$.

Numerical example

To verify whether $\hat{\boldsymbol{A}}_k$ successfully captures the global sensitivities with respect to $\boldsymbol{\theta}$, in particular when the model Lagrangian is a nonlinear function of the parameters, and to compare the performance of $\hat{\boldsymbol{A}}_k$ to that of $\boldsymbol{A}^{\star}$, let us consider the following exemplary Lagrangian function:

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\theta}) = \exp(\theta_1 u_1 + \theta_2 u_2) + \theta_3^2\,(u_3 + u_4) + \theta_2\,(0.5u_3 - u_4), \tag{34a}$$

$$-1 \le u_i \le 1, \quad i = 1, \ldots, 4. \tag{34b}$$

Since the constraints are independent of the parameters $\boldsymbol{\theta}$, the dependency of the Lagrangian on the Lagrange multipliers $\boldsymbol{\mu}$ is omitted in this example.

Let us assume that all the elements of the vector $\boldsymbol{\theta} = [\theta_1,\ \theta_2,\ \theta_3]^{\mathsf{T}}$ are uniformly distributed in the interval $[-2, 0]$. The sensitivity matrix $\boldsymbol{A}^{\star}$ is constructed from the knowledge of $\boldsymbol{\theta}_0 = [-0.5,\ -0.5,\ -0.1]^{\mathsf{T}}$ and $\mathbf{u}^{\star} = \mathbf{u}_1 = [1,\ 1,\ 1,\ -1]^{\mathsf{T}}$. The matrix $\hat{\boldsymbol{A}}_1$ is constructed using Algorithm 2 on the basis of $N = 1000$ Monte-Carlo samples. To compare the two sensitivity matrices, the gradient $\nabla_{\mathbf{u}}\mathcal{L}(\mathbf{u}_1, \boldsymbol{\theta}_j)$ is evaluated for all 1000 Monte-Carlo samples. Each sampled gradient is then projected onto each of the left singular vectors of $\boldsymbol{A}^{\star}$. The resulting projections are plotted in the middle and bottom plots of Fig. 1a. These plots represent the sensitivities of the directional derivatives to parametric perturbations,

Algorithm 2 Computation of matrices $\hat{\boldsymbol{A}}_k$ and $\hat{\boldsymbol{W}}_k$.

```
Step 1: Draw $N$ independent samples $\boldsymbol{\theta}_j$ from $\Theta$ using the probability density $\rho$.
Step 2: Compute the $(n_{\mathrm{u}} \times n_\theta)$-dimensional sensitivity matrix of the Lagrangian gradient:
    $\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}_k^{(j)} := \frac{\partial^2 \mathcal{L}}{\partial \boldsymbol{\theta}\, \partial \mathbf{u}}(\mathbf{u}_k, \boldsymbol{\mu}_k, \boldsymbol{\theta}_j), \quad j = 1, \ldots, N.$
Step 3: Compute $\hat{\boldsymbol{A}}_k$ as follows:
    $\hat{\boldsymbol{A}}_k = \frac{1}{N}\sum_{j=1}^{N}\left(\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}_k^{(j)}\right)\left(\nabla_{\mathbf{u}\boldsymbol{\theta}}\mathcal{L}_k^{(j)}\right)^{\mathsf{T}}.$    (31)
Step 4: Compute the eigenvalue decomposition of $\hat{\boldsymbol{A}}_k$ to obtain $\hat{\boldsymbol{W}}_k$:
    $\hat{\boldsymbol{A}}_k = \hat{\boldsymbol{W}}_k \hat{\boldsymbol{\Sigma}}_k \hat{\boldsymbol{W}}_k^{\mathsf{T}}.$
```

with the vertical width of each plot being a measure of variance. The squared singular values are plotted in the top plot of Fig. 1a. The magnitude of a singular value quantifies the sensitivity of the corresponding directional derivative to parametric variations, that is, the larger the singular value s_i, the larger the variance along the left singular vector $\mathbf{w}_i$. However, as seen in Fig. 1a, the singular vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ do not show this behavior.

The sampled gradients are also projected onto the eigenvectors of $\hat{\boldsymbol{A}}_1$ (the left singular vectors of $\hat{\boldsymbol{B}}_1$) and the resulting projections are shown in the middle and the bottom plots of Fig. 1b. The eigenvalues of $\hat{\boldsymbol{A}}_1$ are plotted in the top plot of Fig. 1b. Here, the eigenvalue magnitude quantifies the parametric sensitivity of $\nabla_{\mathbf{u}}\mathcal{L}$ in the direction given by the corresponding eigenvector. One sees that the eigenvectors of $\hat{\boldsymbol{A}}_1$ are ranked correctly by the corresponding eigenvalues.

3.2.3. Choice of privileged directions

The aforementioned analysis indicates that the privileged directions can also be chosen based on the variance of the directional derivatives. The variance along the direction $\mathbf{d} \in \mathbb{R}^{n_{\mathrm{u}}}$ is computed as

$$\mathrm{Var}\left(\nabla_{\mathbf{u}}\mathcal{L}_k\,\mathbf{d}\right) = \frac{1}{N}\sum_{j=1}^{N}\left|\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)}\,\mathbf{d} - \mathrm{m}\right|^2 \quad \text{with} \quad \mathrm{m} = \frac{1}{N}\sum_{j=1}^{N}\nabla_{\mathbf{u}}\mathcal{L}_k^{(j)}\,\mathbf{d}. \tag{35}$$

For the numerical example at hand, the variance is computed from $N = 1000$ Monte-Carlo samples and plotted in Fig. 2. One sees that the variance does not decrease monotonically for the left singular vectors of $\boldsymbol{A}^{\star}$, whereas it decreases monotonically for the eigenvectors of $\hat{\boldsymbol{A}}_1$. A monotonic variance decrease indicates that the eigenvalues are ranking the eigenvectors in the right order. To determine the privileged directions, a threshold value on the variance is fixed at 10^{-2}, and the directions that result in a variance larger than the threshold value are chosen as privileged directions. A variance smaller than 10^{-2} indicates that the parametric changes do not cause a significant change in the gradients along that direction and, therefore, the gradient errors along that direction are relatively small. The global sensitivity matrix $\hat{\boldsymbol{A}}_1$ yields $n_{\mathrm{r}} = 2$ privileged directions, namely $\hat{\mathbf{w}}_{1,1}$ and $\hat{\mathbf{w}}_{2,1}$, whereas the local sensitivity matrix $\boldsymbol{A}^{\star}$ yields $n_{\mathrm{r}} = 3$ privileged directions, namely $\mathbf{w}_1, \mathbf{w}_2$ and $\mathbf{w}_3$. Hence, the global sensitivity matrix $\hat{\boldsymbol{A}}_1$ finds a smaller set of privileged directions, thereby reducing the number of required plant experiments for gradient estimation. At the same time, $\hat{\boldsymbol{A}}_1$ ensures that the gradient errors due to parametric perturbations are small along the neglected directions.

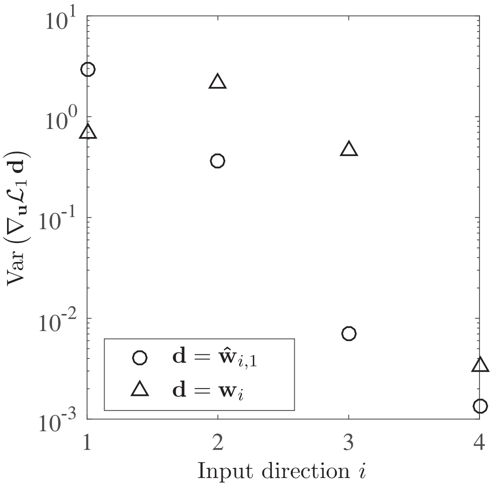


Fig. 2. Variance plot for the sensitivity matrices $\boldsymbol{A}^{\star}$ and $\hat{\boldsymbol{A}}_1$ computed for the example (34).

Often, one selects the maximal number of privileged directions, $n_{\max}$, so as to upper-bound the experimental budget per RTO iteration. Then, on the basis of the eigenvalues $\hat{\sigma}_{i,k}$, the variance $\mathrm{Var}\left(\nabla_{\mathbf{u}}\mathcal{L}_k\,\mathbf{d}\right)$ and $n_{\max}$, one can choose the number of privileged directions n_{r} using one of the following two criteria:

Criterion 1

$$n_{\mathrm{r}} = \min\{i, n_{\max}\} : \hat{\sigma}_{i+1,k} \ll \hat{\sigma}_{i,k}\,, \tag{36}$$

Criterion 2

$$\begin{aligned} n_{\mathrm{r}} &= \min\{i, n_{\max}\} : \mathrm{Var}\left(\nabla_{\mathbf{u}}\mathcal{L}_k\,\hat{\mathbf{w}}_{i,k}\right) \geq \mathrm{v}_{\min} \text{ and} \\ &\quad \mathrm{Var}\left(\nabla_{\mathbf{u}}\mathcal{L}_k\,\hat{\mathbf{w}}_{i+1,k}\right) < \mathrm{v}_{\min} \end{aligned} \tag{37}$$

where $n_{\max}$ and the threshold variance $\mathrm{v}_{\min}$ are user-defined parameters. The matrix of privileged directions $\hat{\boldsymbol{W}}_{1,k}$ then becomes:

$$\hat{\boldsymbol{W}}_{1,k} = [\hat{\mathbf{w}}_{1,k} \ \ldots \ \hat{\mathbf{w}}_{n_{\mathrm{r}},k}]. \tag{38}$$

3.2.4. Evaluation of plant directional derivatives

In practice, one often relies on finite differences to evaluate the plant gradients. To reduce the number of plant experiments for gradient estimation, well-excited plant measurements obtained at past RTO iterations can be used. To this end, the optimization objective and the gradient estimation objective are combined by enforcing duality constraints at the RTO layer (Brdys and Tatjewski, 1994; Marchetti et al., 2010). Furthermore, estimating gradients from noisy measurements can be reliably achieved by quadratic-approximation of the plant mapping (Gao et al., 2016). For a comparative study of different gradient estimation techniques in RTO, we recommend the paper by Mansour and Ellis (2003). The different approaches mentioned here can also be exploited to estimate the plant directional derivatives defined in (10).

3.2.5. Structural plant-model mismatch

Assumption 1regarding parametric plant-model mismatch may not be met in practice. In the case of structural plant-model mismatch, ADMA still drives the plant toward optimality in the subspace given by the privileged directions. The following theorem does not require Assumption 1.

Theorem 3.2 (Optimality limited to privileged directions)**.** *Consider the optimization Problem (3) with the modifiers (20) and the gradient*

Algorithm 3 Active Directional Modifier Adaptation (ADMA)

```
Step 0 (Initialization): Set the initial values of the modifiers to zero, $\boldsymbol{\varepsilon}_0^{\mathrm{G}} = \mathbf{0}$, $\boldsymbol{\lambda}_0^{\Phi} = \mathbf{0}$ and
    $\boldsymbol{\lambda}_0^{\mathrm{G}_i} = \mathbf{0}$, and the values of the filter matrices $\boldsymbol{K}^{\varepsilon}$, $\boldsymbol{K}^{\Phi}$ and $\boldsymbol{K}^{\mathrm{G}_i}$ (typically diagonal matrices)
    with eigenvalues in the interval $(0,1]$. Also, set arbitrarily $\mathbf{u}_0 = \mathbf{0}$ and select values for $n_{\max}$
    and $\mathrm{v}_{\min}$. and set the values of $n_{\max}$ and $\mathrm{v}_{\min}$. Scale the parameters $\boldsymbol{\theta}$ such that the scaled
    parametric uncertainty range is $[\mathbf{-1}, \mathbf{1}]$.

for $k = 0 \rightarrow \infty$

Step 1 (Optimization): Solve the modified optimization Problem (3) for $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ to generate the
    optimal inputs $\mathbf{u}_{k+1}$ and the corresponding Lagrange multipliers $\boldsymbol{\mu}_{k+1}$.

Step 2 (Plant evaluation): Apply $\mathbf{u}_{k+1}$ to the plant and collect the measurements $\mathbf{y}_p(\mathbf{u}_{k+1})$.
    Use these measurements to evaluate $\Phi_p(\mathbf{u}_{k+1})$ and $\mathbf{G}_p(\mathbf{u}_{k+1})$.

Step 3 (Computation of privileged directions): Compute $\hat{\boldsymbol{W}}_{k+1}$ using Algorithm 2 and the
    privileged direction matrix $\hat{\boldsymbol{W}}_{1,k+1}$ using either Criterion 1 in (36) or Criterion 2 in (37).

Step 4 (Estimation of directional derivatives): Estimate the directional derivatives of the
    plant cost $\nabla_{\hat{\mathrm{W}}_{1,k+1}}\Phi_p(\mathbf{u}_{k+1})$ and of the constraints $\nabla_{\hat{\mathrm{W}}_{1,k+1}}\mathrm{G}_{p,i}(\mathbf{u}_{k+1})$, $i = 1, \ldots, n_{\mathrm{g}}$. At
    $\mathbf{u}_{k+1}$, the full gradients are computed as:

    $\widehat{\nabla_{\mathbf{u}}\Xi_p}(\mathbf{u}_{k+1}) := \nabla_{\mathbf{u}}\Xi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)(\boldsymbol{I}_{n_{\mathrm{u}}} - \hat{\boldsymbol{W}}_{1,k+1}\hat{\boldsymbol{W}}_{1,k+1}^{+}) + \nabla_{\hat{\mathrm{W}}_{1,k+1}}\Xi_p(\mathbf{u}_{k+1})\hat{\boldsymbol{W}}_{1,k+1}^{+}$,

    with $\Xi \in \{\Phi, \mathrm{G}_i\}$, and $\hat{\boldsymbol{W}}_{1,k+1}^{+}$ the Moore-Penrose pseudo-inverse of $\hat{\boldsymbol{W}}_{1,k+1}$.

Step 5 (Modifier update): Update the modifiers using first-order filters:

    $\boldsymbol{\varepsilon}_{k+1}^{\mathrm{G}} = (\boldsymbol{I}_{n_{\mathrm{g}}} - \boldsymbol{K}^{\varepsilon})\boldsymbol{\varepsilon}_k^{\mathrm{G}} + \boldsymbol{K}^{\varepsilon}\left(\mathbf{G}_p(\mathbf{u}_{k+1}) - \mathbf{G}(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right)$,
    $\boldsymbol{\lambda}_{k+1}^{\Phi} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\Phi})\boldsymbol{\lambda}_k^{\Phi} + \boldsymbol{K}^{\Phi}\left(\widehat{\nabla_{\mathbf{u}}\Phi_p}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\Phi(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right)^{\mathsf{T}}$,
    $\boldsymbol{\lambda}_{k+1}^{\mathrm{G}_i} = (\boldsymbol{I}_{n_{\mathrm{u}}} - \boldsymbol{K}^{\mathrm{G}_i})\boldsymbol{\lambda}_k^{\mathrm{G}_i} + \boldsymbol{K}^{\mathrm{G}_i}\left(\widehat{\nabla_{\mathbf{u}}\mathrm{G}_{p,i}}(\mathbf{u}_{k+1}) - \nabla_{\mathbf{u}}\mathrm{G}_i(\mathbf{u}_{k+1}, \boldsymbol{\theta}_0)\right)^{\mathsf{T}}$,   $i = 1, \ldots, n_{\mathrm{g}}$.

end
```

updates (21). Also, assume that $\boldsymbol{\theta}_0 \in \Theta$. Let Assumptions 2 and 3 hold. If the iterative solution to this problem converges to the fixed point $\left(\mathbf{u}_\infty, \boldsymbol{\varepsilon}_\infty^{\mathrm{G}}, \boldsymbol{\lambda}_\infty^{\Phi}, \boldsymbol{\lambda}_\infty^{\mathrm{G}}\right)$, with $\mathbf{u}_\infty$ being the KKT point of Problem (3), then $\mathbf{u}_\infty$ is also a KKT point for the plant in the directions given by the columns of the matrix $\boldsymbol{W}_{1,\infty}$.

Proof. See Costello et al. (2016) □

3.3. ADMA algorithm

The proposed ADMA algorithm is summarized in Algorithm 3.

4. Case studies

The proposed approach is illustrated next via the simulation of two semi-batch reactors.

Table 1
Williams-Otto reactor: plant-model mismatch.

Parameter	Plant value	Nominal model value	Uncertainty range	Probability distribution
b_1 (K)	6000	6666.7	[5334, 8000]	Uniform
b_2 (K)	8333.3	8750	[7500, 9166]	Uniform
a_1 (s^{-1})	$1.6599 \cdot 10^6$	$1.8259 \cdot 10^6$	$[1.4109 \cdot 10^6, 1.9089 \cdot 10^6]$	Uniform
a_2 (s^{-1})	$7.2117 \cdot 10^8$	$6.8511 \cdot 10^8$	$[6.1299 \cdot 10^8, 8.2935 \cdot 10^8]$	Uniform

4.1. Williams-Otto semi-batch reactor

ADMA is applied to the problem of run-to run (batch-to-batch) optimization of the Williams-Otto semi-batch reactor described in Würth et al. (2009) and Salau et al. (2014). The following reactions take place in the reactor:

$$A + B \xrightarrow{k_1} C, \quad C + B \xrightarrow{k_2} P + E, \quad P + C \xrightarrow{k_3} G.$$

Reactant A is initially present in the reactor, whereas reactant B is continuously fed during the batch. As a result of the exothermic reactions, the desired products P and E and the side product G are formed. The heat generated by the exothermic reactions is removed via a cooling jacket, whose temperature is controlled by manipulating the cooling water temperature. The model equations and parameter values are given in Appendix A.

The objective is to maximize the revenue generated by selling the products produced at the end of the batch, while respecting path constraints on the inlet flowrate of reactant B (F_B), the reactor temperature (T_r), the reactor volume (V) and the cooling water temperature (T_w). The manipulated variables are the time-varying profiles $F_B(t)$ and $T_w(t)$. The dynamic optimization problem can be written mathematically as follows:

$$\max_{F_B(t), T_w(t)} \; P_P \, m_P(t_f) + P_E \, m_E(t_f) \tag{39a}$$

$$\text{s.t.} \quad \text{model equations } (A.1) \tag{39b}$$

$$0 \leq F_B(t) \leq 5.784 \, \text{kg/s} \tag{39c}$$

$$V(t) \leq 5 \, \text{m}^3 \tag{39d}$$

$$20\,^\circ\text{C} \leq T_w(t) \leq 100\,^\circ\text{C} \tag{39e}$$

$$60\,^\circ\text{C} \leq T_r(t) \leq 90\,^\circ\text{C}. \tag{39f}$$

The batch time t_f is fixed at 1000 s. Problem (39) is transformed into a nonlinear program (NLP) via direct single shooting. This is done by discretizing the problem in time over n_{cs} control stages. For each time interval, the dynamic input variables are parametrized using low-order polynomials. We parameterize each time-varying input using $n_{cs} = 40$ piecewise-constant values. This results in the input dimension $n_u = 80$.

In this simulation study, the plant is substituted by a simulated reality. The simulated reality is then treated as a black box and it is assumed that the concentration measurements of the formed products are available only at the final batch time t_f. This permits the simulated reality to act as a real system/plant.

The experimental cost of evaluating the plant gradients via finite differences is found as follows:

$$\begin{aligned} &\text{Experimental cost per RTO iteration} \\ &\quad = \text{Total number of privileged directions} \times \text{Batch time}. \end{aligned}$$

This implies that the cost to evaluate the full plant gradients is $n_u \times t_f$. However, with the DMA and ADMA algorithms, only the derivatives in the privileged directions need to be evaluated, with the experimental cost $n_r \times t_f$.

Plant-model mismatch is introduced by considering parametric uncertainty in the values of the activation energies b_1 and b_2 and the pre-exponential factors a_1 and a_2. The parameter values for the plant and the nominal model, and the uncertainty ranges are given in Table 1. The parameters are assumed to be uniformly distributed within the given uncertainty intervals.

The optimal input profiles obtained by solving Problem (39) for both the nominal model and the plant are shown in the top plots of Fig. 3. As seen in the figure, the model and plant solutions are quite different. The optimal revenue for the plant is $3.14 \cdot 10^6$. However, upon applying the model solution to the plant, a sub-optimal revenue of $1.66 \cdot 10^6$ is obtained.

The optimal input profiles for the plant are assumed to be unknown. Hence, the goal of the different RTO methods is to improve upon the sub-optimal revenue resulting from applying the model optimal solution. The best RTO method is the one that requires the minimal experimental effort to reach plant optimality.

At first, MA with full gradient estimates is implemented. The top plots in Fig. 5 show the input and output profiles obtained with MA upon convergence. Although MA starts from the input sequences given by the model solution, it is able to identify the correct set of constraints that are active at the plant optimum, thereby reaching the maximal possible revenue. However, MA requires full gradient estimation and, thus, incurs a large experimental cost at each RTO iteration, which makes the application of this type of MA prohibitive in practice.

To implement the DMA and ADMA algorithms, the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_k$ are constructed. The number of Monte-Carlo samples for constructing $\hat{\boldsymbol{A}}_k$ is $N = 200$. The number of privileged directions is determined successively based on Criteria 1 and 2 in (36) and (37). To this end, the values of $n_{\max}$ and $v_{\min}$ are fixed at 4 and 1, respectively.

Criterion 1. The squared singular values of $\boldsymbol{A}^\star$ and the eigenvalues of $\hat{\boldsymbol{A}}_1$ at the model solution at $k = 1$ are plotted in Fig. 4a. Notice that there is a large gap between the second and third singular values of $\boldsymbol{A}^\star$. Hence, based on Condition (8) or (36), the number of privileged directions for DMA can be fixed at $n_r = 2$. However, for the same gap, the number of privileged directions with $\hat{\boldsymbol{A}}_1$ is more than $n_{\max} = 4$. Hence, for a fair comparison between DMA and ADMA, we fix the number of privileged directions for both algorithms at 4.

Criterion 2. The number of privileged directions can also be determined on the basis of variance values as described by (37). The variances associated with both the left singular vectors of $\boldsymbol{A}^\star$ and the eigenvalues of $\hat{\boldsymbol{A}}_1$ are plotted in Fig. 4b. It is seen that the variances associated with the left singular vectors of $\boldsymbol{A}^\star$ do not exhibit a monotonic decrease except for the first few left singular vectors. In contrast, the variances associated with the eigenvectors of $\hat{\boldsymbol{A}}_1$ show a monotonic decrease. The median and the minimal value of the variance for $\hat{\boldsymbol{A}}_1$ are 0.51 and $1.1 \cdot 10^{-15}$, respectively. In comparison, the median and the minimal values of the variance for $\boldsymbol{A}^\star$ are $2.3 \cdot 10^8$ and $6.7 \cdot 10^5$, respectively. Hence, the eigenvectors of $\hat{\boldsymbol{A}}_1$ give a much better orthogonal decomposition of the input space, which is able to capture most of the global sensitivity in a relatively low number of directions. The threshold $v_{\min} = 1$ is not very useful here as it retains all 80 directions for DMA and 40 directions for ADMA. Hence, Criterion 2 also gives $n_r = 4$ for both DMA and ADMA. In the ADMA algorithm, we then keep the number of privileged directions fixed at 4 for every RTO iteration for

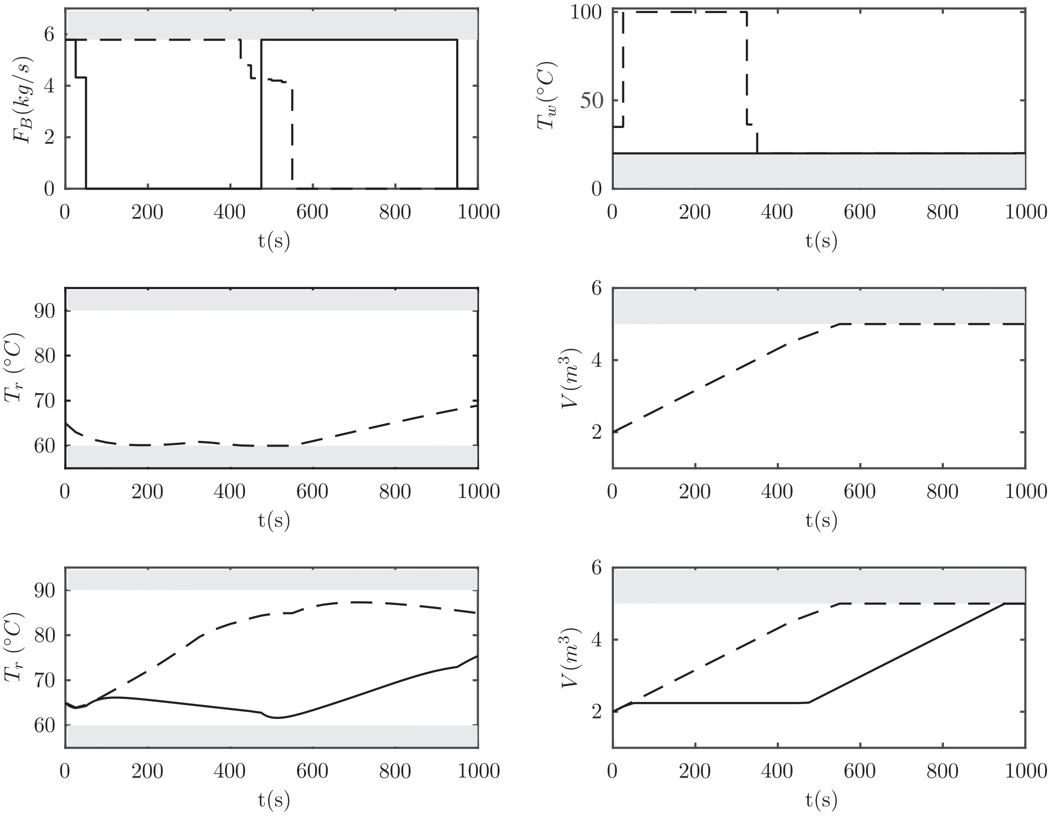
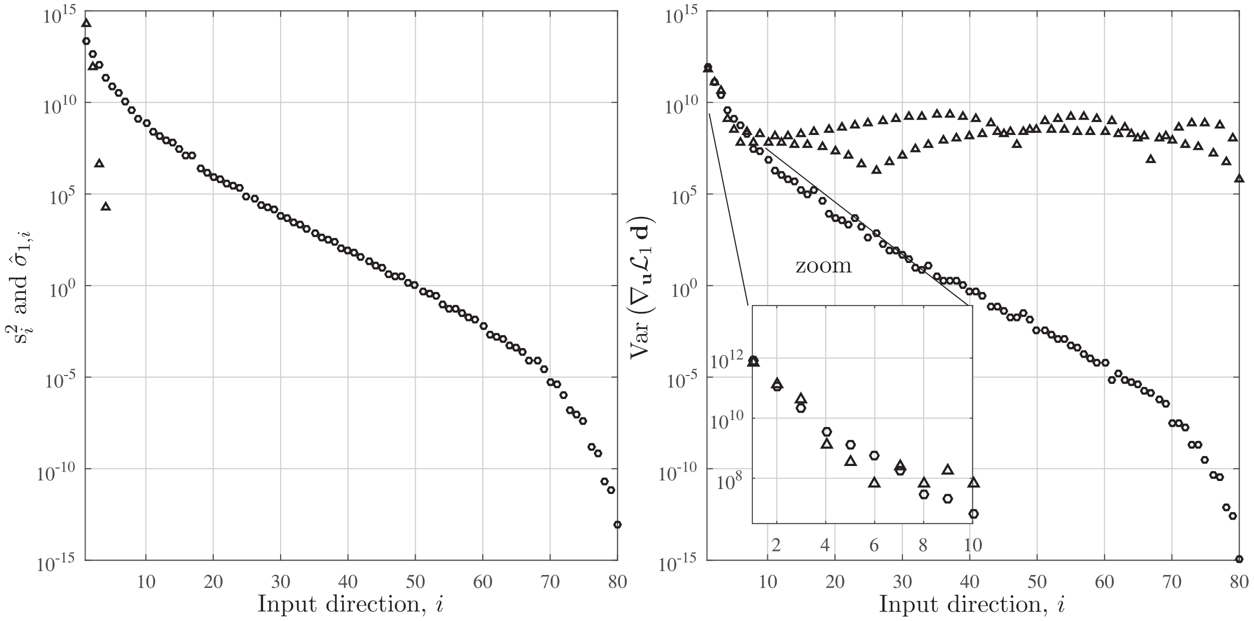


Fig. 3. Williams-Otto reactor. Shaded area: infeasible region. **Top plots:** *input variables*: dashed lines: model optimal solution; solid lines: plant optimal solution. **Middle plots:** *constrained output variables*: model optimal solution. **Bottom plots:** *constrained output variables*: dashed lines: plant at the model optimal solution; solid lines: plant at the plant optimal solution.



(a) **Triangle**, △ **:** Squared singular values of $\boldsymbol{A}^\star$. **Circle**, ○ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$.

(b) **Triangle**, △ **:** $\mathbf{d} = \mathbf{w}_i$, the left singular vectors of $\boldsymbol{A}^\star$. **Circle**, ○ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$.

Fig. 4. Williams-Otto reactor: comparison of the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_1$.

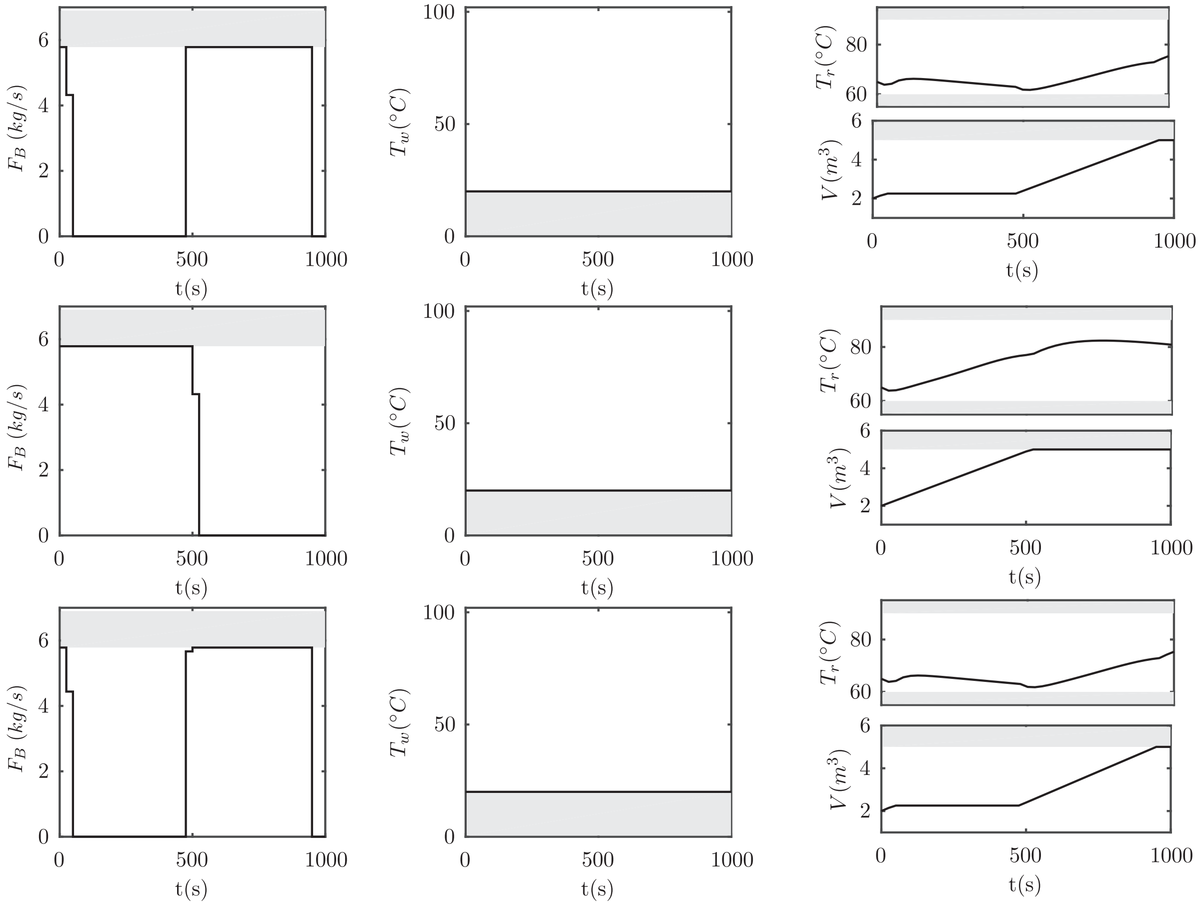


Fig. 5. Williams-Otto reactor. Plant input and output profiles upon convergence with different RTO methods. **Top plots:** MA. **Middle plots:** DMA. **Bottom plots:** ADMA..

Table 2
Williams-Otto reactor: comparison of different RTO methods.

RTO method	Per RTO iteration			Revenue $\cdot(10^6)$
	No. of privileged directions, n_r	Avg. computational time of sensitivity matrix	Experimental cost	
MA	80	–	22.23 h	3.14
DMA	4	0 s ($\boldsymbol{A}^{\star}$ computed only once offline)	1.12 h	2.13
ADMA	4	90.92 s ($\hat{\boldsymbol{A}}_k$ computed via Algorithm 2)	1.12 h	3.14

the sake of comparison with DMA. The sets of 4 privileged directions computed by DMA and ADMA at the first RTO iteration are different as the model Lagrangian is a highly nonlinear function of the model parameters b_1 and b_2. Note that the 4 privileged directions in ADMA change from one iteration to the next due to the re-computation of $\hat{\boldsymbol{A}}_k$ at each iteration.

Upon application of DMA Algorithm 1, the plant input and output profiles reached upon convergence are shown in the middle plots of Fig. 5. Although DMA successfully finds the optimal water temperature profile, it is unable to find the optimal profile for the feedrate of B. Obviously, adapting the gradients in the 4 privileged directions found by DMA is not sufficient to reach plant optimality as the gradient uncertainty along these directions is not sufficiently representative. The bottom plots of Fig. 5 show the input and output profiles obtained with the ADMA Algorithm 3. As seen, ADMA successfully reaches the plant optimal profiles (see also Fig. 6). This indicates that most of the gradient errors lie along the 4 privileged directions of ADMA.

The comparison of the different MA-based RTO methods is summarized in Table 2. As MA requires full gradient estimation, one must have 80 batches to evaluate the plant gradients at each

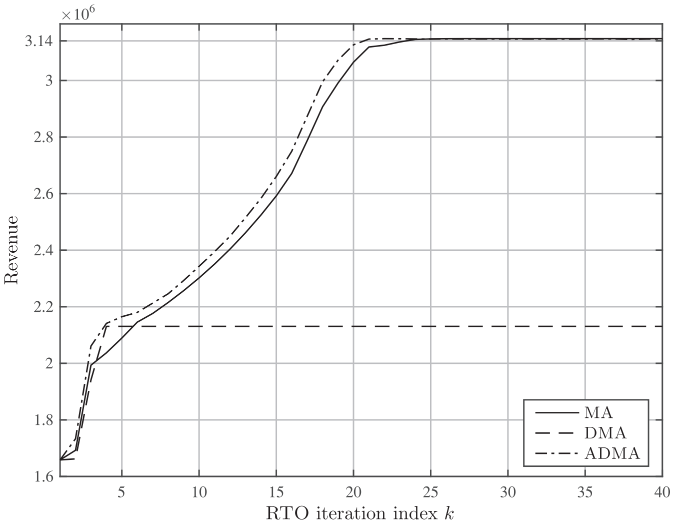


Fig. 6. Willliams-Otto reactor: Revenue evolution of the plant with different RTO methods.

Table 3
Kinetic parameters for diketene-pyrrole reaction.

Parameter	Plant value	Nominal model value	Uncertainty range	Probability distribution
k_1 (Lmol^{-1}min^{-1})	0.053	0.053	[0.0424, 0.0636]	Uniform
k_2 (Lmol^{-1}min^{-1})	0.128	0.128	[0.1024, 0.1536]	Uniform
k_3 (min^{-1})	0.028	0	–	–
k_4 (Lmol^{-1}min^{-1})	0.001	0	–	–

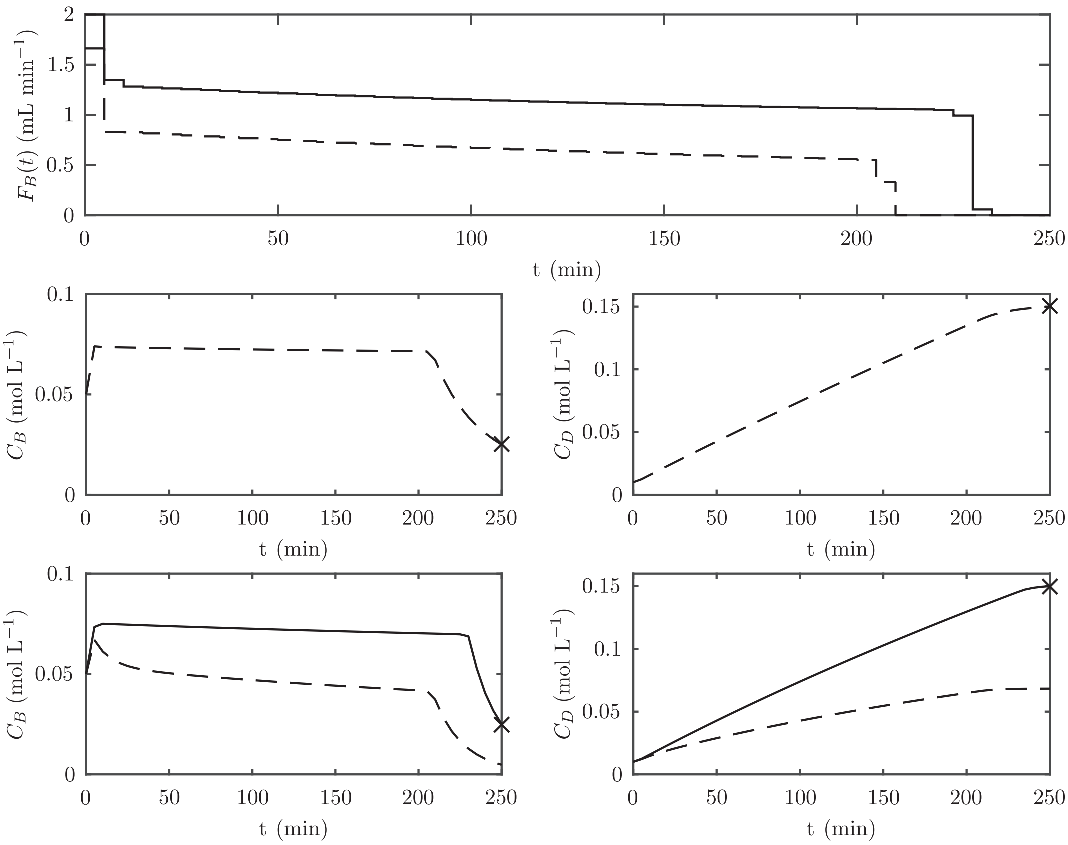


Fig. 7. Diketene-pyrrole reaction. **Top plots:** *input profile*: dashed line: model optimal solution; solid line: plant optimal solution. **Middle plots:** *output variables that are constrained at final time*: model optimal solution. **Bottom plots:** *output variables that are constrained at final time*: dashed lines: plant response at the model optimal solution; solid lines: plant response at the plant solution. Cross, × : terminal constraint threshold.

RTO iteration. That amounts to 22.23 h of waiting time per RTO iteration. This experimental time is reduced by applying DMA, which requires directional derivatives to be computed in only 4 directions, thus, needing only 4 batches, which reduces the experimental cost to 1.12 h. However, the maximal revenue reached by DMA is only $2.13 \cdot 10^6$. ADMA, on the other hand, gives an optimal revenue of $3.14 \cdot 10^6$, while incurring the same experimental cost as DMA. This increase in revenue is made possible by the matrix $\hat{\boldsymbol{A}}_k$ that requires on average a computational time of 90.92 s per RTO iteration when computed via Algorithm 2.[3]

4.2. Diketene-pyrrole reaction system

Next, we compare the performance of the different RTO methods on the run-to-run optimization of the semi-batch reactor given in Ruppen et al. (1997) or Chachuat et al. (2009). The reaction system is the acetoacetylation of pyrrole with diketene and consists of following reactions:

$$A + B \xrightarrow{k_1} C, \quad 2B \xrightarrow{k_2} D, \quad B \xrightarrow{k_3} E, \quad C + B \xrightarrow{k_4} F.$$

The model equations, the initial conditions and the concentration of B in the feed used in this simulation study are given in Appendix B. The objective is to maximize the yield of product C, while penalizing large changes in the feedrate F_B of reactant B. The optimization problem can be written mathematically as:

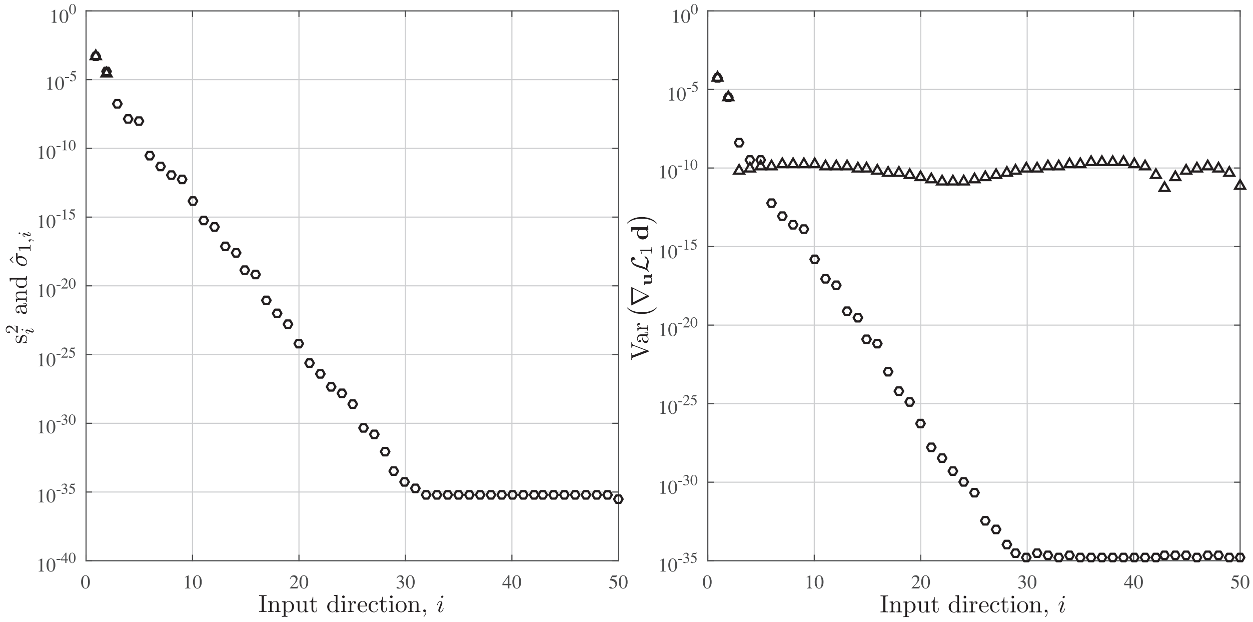
$$\max_{F_B(t)} \quad c_C(t_f)V(t_f) - \omega \int_0^{t_f} F_B^2(t)\,dt \tag{40a}$$

$$\text{s.t.} \quad \text{model Eq. (B.1)} \tag{40b}$$

$$c_B(t_f) \le c_B^{\max} \tag{40c}$$

$$c_D(t_f) \le c_D^{\max} \tag{40d}$$

[3] Simulations were conducted on a MacBook Pro with 2.5 GHz intel Core i7 processor. The software used is CasADi (Andersson, 2013) version 3.2.3 in MATLAB version R2016a.



(a) **Triangle**, △ **:** Squared singular values of $\boldsymbol{A}^\star$. **Circle**, ○ **:** Eigenvalues of $\hat{\boldsymbol{A}}_1$.

(b) **Triangle**, △ **:** $\mathbf{d} = \mathbf{w}_i$, the left singular vectors of $\boldsymbol{A}^\star$. **Circle**, ○ **:** $\mathbf{d} = \hat{\mathbf{w}}_{1,i}$, the eigenvectors of $\hat{\boldsymbol{A}}_1$.

Fig. 8. Diketene-pyrrole reaction: comparison of the sensitivity matrices $\boldsymbol{A}^\star$ and $\hat{\boldsymbol{A}}_1$.

$$0 \le F_B(t) \le F_B^{\max}. \tag{40e}$$

The values of the final batch time t_f, the maximal inlet flowrate $F_B^{\max}$, the maximal concentrations of species B and D at final time and the value of the weight ω are given in Table B.6. The problem is formulated as an NLP by using a piecewise-constant discretization of the input $F_B(t)$ comprised of 50 control stages. Hence, the input dimension is $n_u = 50$.

Structural plant-model mismatch is introduced by ignoring the third and fourth reactions in the model, that is, by taking $k_3 = 0$ and $k_4 = 0$ for the model. Also, it is assumed that the model parameters k_1 and k_2 are uncertain and uniformly distributed within the ranges corresponding to ±20% of the nominal values. The mismatch considered and the uncertainty ranges are given in Table 3. The optimal solutions for the model and the plant are shown in the top plot of Fig. 7. The two input profiles are quite different from each other. The evolution of the model $c_B(t)$ and $c_D(t)$ at the model optimal solution is shown in the middle plots of Fig. 7. The bottom plots of Fig. 7 show the evolution of the plant $c_B(t)$ and $c_D(t)$ obtained upon application of both the model and the plant optimal solutions. It is observed that the terminal constraint on the concentration of reactant B is not at its upper limit for the plant when the model optimal solution is applied. The model optimal solution applied to the plant result in a sub-optimal yield of 0.3865 moles, whereas the plant optimal yield is 0.5050 moles (Table A.5).

Since only the two parameters k_1 and k_2 are uncertain, the local sensitivity matrix $\boldsymbol{A}^\star$ generates 2 privileged directions as per Condition (9). The variances along the left singular vectors of $\boldsymbol{A}^\star$ are plotted in Fig. 8b. It is seen that the variances along the remaining 48 directions do not decrease monotonically and, thus, more privileged directions cannot be selected. To construct $\hat{\boldsymbol{A}}_k$ via Algorithm 2, the number of Monte-Carlo samples is fixed at $N = 100$. Here, in contrast to the previous case study, we do not fix the number of privileged directions in ADMA; instead, we apply Criterion 1 in (36) at each RTO iteration by fixing $n_{\max} = 4$. The eigenvalues and variances computed at $k = 1$ are plotted in Fig. 8. Criterion 1 gives 2 privileged directions at the first RTO iteration for ADMA. In this example, the two privileged directions found by DMA and ADMA are the same, which results from the fact that the model Lagrangian is only a weakly nonlinear function of the parameters k_1 and k_2. Note that these 2 privileged directions are *less privileged* at the next iterations since the privileged directions change with the input $F_B(t)$ from iteration to iteration. However, since ADMA recomputes the privileged directions at each RTO iteration, it is able to always work with the most appropriate set of privileged directions. The number of privileged directions found at each RTO iteration using Criterion 1 is plotted in Fig. 10b. The input and output profiles reached upon convergence with MA, DMA and ADMA are shown in Fig. 9. The evolution of the yield with the different RTO methods is shown in Fig. 10a. Clearly, DMA exhibits a slight sub-optimality, whereas the MA and ADMA algorithms converge to plant optimality (at least as far as the yield value is concerned) despite the presence of significant plant-model mismatch.

The performance of the different RTO methods is compared in Table 4. MA with full gradient estimation reaches the optimal yield of 0.5050 moles at the large experimental cost of 208.34 h per RTO iteration. DMA significantly reduces the experimental cost to 8.34 h per RTO iteration, but reaches a final yield of only 0.5009 moles. In comparison, ADMA nearly reaches optimal yield at an experimental cost of 8.34–12.5 h. The average computational cost of $\hat{\boldsymbol{A}}_k$ is 5.43 s. ADMA gives the best performance as it comes very close to the plant optimum at a relatively low experimental cost. The computer and software used to perform the simulations are the same as in the previous case study.

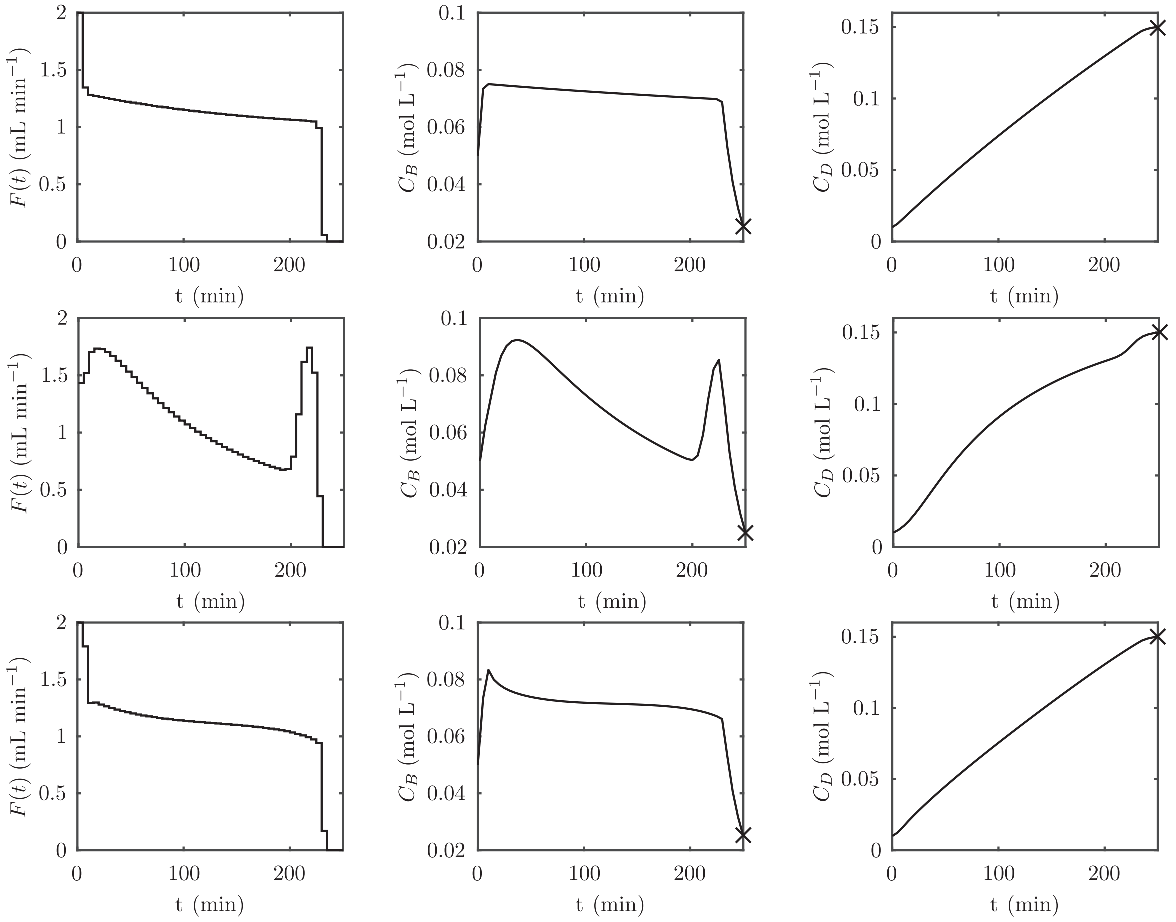
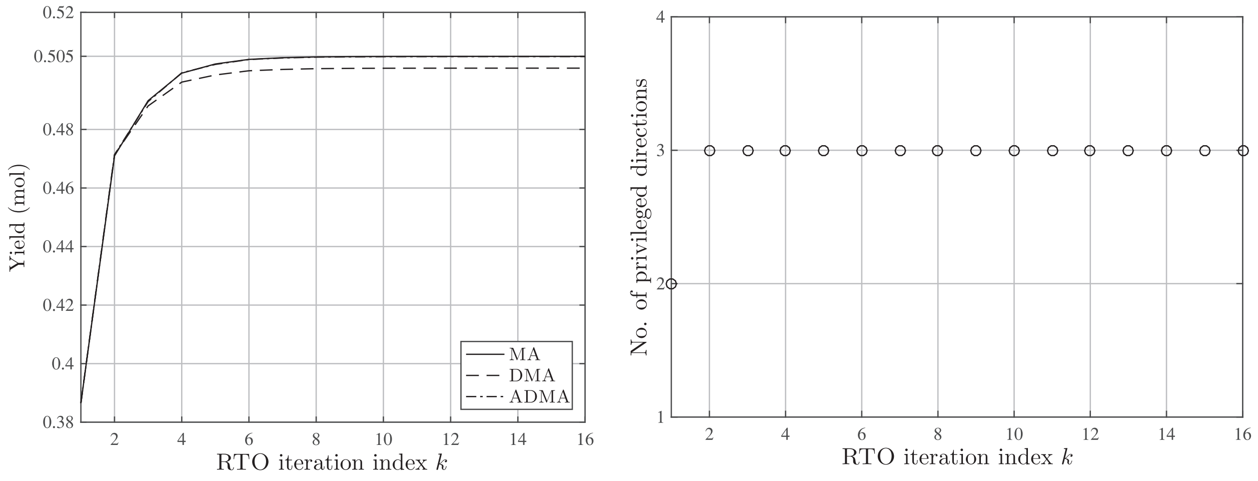


Fig. 9. Diketene-pyrrole reaction: plant input and output profiles upon convergence with different RTO methods. **Top plots:** MA. **Middle plots:** DMA. **Bottom plots:** ADMA with $\hat{\boldsymbol{A}}_k$ computed via Algorithm 2.



(a) Yield evolution for different RTO methods

(b) Number of privileged directions at successive iterations

Fig. 10. Diketene-pyrrole reaction: batch-to-batch evolution of the optimization strategy.

Table 4
Diketene-pyrrole reaction: comparison of different RTO methods.

RTO method	Per RTO iteration			Yield (mol)
	No. of privileged directions, n_r	Avg. computational time of sensitivity matrix	Experimental cost	
MA	50	–	208.34 h	0.5050
DMA	2	0 s ($\boldsymbol{A}^*$ computed only once offline)	8.34 h	0.5009
ADMA	2–3	5.43 s ($\hat{\boldsymbol{A}}_k$ computed via Algorithm 2)	8.34–12.5 h	0.5049

5. Conclusions

This paper presents a novel methodology for reducing the burden of excessive plant experiments for gradient estimation in MA schemes. It is proposed to estimate the plant gradients only in a subspace of the input space. This subspace is found via sensitivity analysis with respect to parametric uncertainty. A global sensitivity analysis inspired by active subspace theory is used to rank the input directions in terms of the sensitivity of the directional derivatives with respect to parametric variations, thus revealing a set of privileged directions. The effectiveness of the proposed ADMA scheme is demonstrated on the run-to-run optimization of two different semi-batch reactors. It is shown that ADMA offers a nice balance between experimental cost and achieved performance compared to both MA with full gradient estimation and the DMA approach based on local sensitivities.

Acknowledgments

The research leading to these results has received funding from the Swiss National Science Foundation under the project number 149258.

Appendix A. Williams-Otto reactor model

The model equations are as follows:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \frac{F_B}{q} \tag{A.1a}$$

$$\frac{\mathrm{d}m_A}{\mathrm{d}t} = -r_1 V \tag{A.1b}$$

$$\frac{\mathrm{d}m_B}{\mathrm{d}t} = F_B - \frac{M_{w_A}}{M_{w_B}} r_1 V + r_2 V \tag{A.1c}$$

$$\frac{\mathrm{d}m_C}{\mathrm{d}t} = \frac{M_{w_C}}{M_{w_A}} r_1 V - \frac{M_{w_C}}{M_{w_B}} r_2 V - r_3 V \tag{A.1d}$$

$$\frac{\mathrm{d}m_E}{\mathrm{d}t} = \frac{M_{w_E}}{M_{w_B}} r_2 V \tag{A.1e}$$

$$\frac{\mathrm{d}m_G}{\mathrm{d}t} = \frac{M_{w_G}}{M_{w_C}} r_3 V \tag{A.1f}$$

$$\frac{\mathrm{d}m_P}{\mathrm{d}t} = \frac{M_{w_P}}{M_{w_B}} r_2 V - \frac{M_{w_P}}{M_{w_C}} r_3 V \tag{A.1g}$$

$$\frac{\mathrm{d}T_r}{\mathrm{d}t} = \frac{H}{VC_p} \tag{A.1h}$$

$$c_i = m_i/V; \quad i = A, B, C, P, E, G$$
$$k_i = a_i e^{-b_i(T_r + T_{ref})}; \quad i = 1, 2, 3$$
$$r_1 = k_1 c_A c_B; \quad r_2 = k_2 c_B c_C; \quad r_3 = k_3 c_C c_P$$
$$H = F_B C_p T_{in} - \Delta H_1 r_1 V - \Delta H_2 r_2 V - \Delta H_3 r_3 V - V \frac{A_0}{V_0} U(T_r - T_w)$$

Appendix B. Diketene-pyrrole reactor model

The first-principles model for the semi-batch reactor reads:

$$\frac{\mathrm{d}c_A}{\mathrm{d}t} = -\mathrm{k}_1 c_A c_B - \frac{F_B}{V} c_A \tag{B.1a}$$

$$\frac{\mathrm{d}c_B}{\mathrm{d}t} = -\mathrm{k}_1 c_A c_B - 2\mathrm{k}_2 c_B^2 - \mathrm{k}_3 c_B - \mathrm{k}_4 c_B c_C + \frac{F_B}{V}(c_B^{in} - c_B) \tag{B.1b}$$

$$\frac{\mathrm{d}c_C}{\mathrm{d}t} = \mathrm{k}_1 c_A c_B - \mathrm{k}_4 c_B c_C - \frac{F_B}{V}(c_C) \tag{B.1c}$$

$$\frac{\mathrm{d}c_D}{\mathrm{d}t} = \mathrm{k}_2 c_B^2 - \frac{F_B}{V}(c_D) \tag{B.1d}$$

$$\frac{\mathrm{d}V}{\mathrm{d}t} = F_B, \tag{B.1e}$$

where c_A, c_B, c_C and c_D represent the concentrations of the species A, B, C and D, respectively. V is the reactor volume, F_B is the inlet flowrate of species B, and c_B^{in} is the concentration of B in the feed.

Table A.5
Reaction parameters and operating conditions for the Williams-Otto semi-batch reactor.

Variable	Definition	Value
M_A, M_B, M_P	Molar mass - components A, B, P	100 kg kmol^{-1}
M_C, M_E	Molar mass - component C, G	200 kg kmol^{-1}
M_G	Molar mass - component G	300 kg kmol^{-1}
a_3	Pre-exponential fraction - reaction 3	$2.6745 \cdot 10^{12}$ s^{-1}
b_3	Activation energy - reaction 3	11111 K
T_{ref}	Reference temperature	273.15 K
T_{in}	Inlet temperature (B)	308.15 K
ΔH_1	Enthalpy - reaction 1	236.8 kJ kg^{-1}
ΔH_2	Enthalpy - reaction 2	158.3 kJ kg^{-1}
ΔH_3	Enthalpy - reaction 3	226.3 kJ kg^{-1}
A_0	Heat-transfer area	9.2903 m^2
V_0	Cooling jacket volume	2.1052 m^3
U	Heat-transfer coefficient	0.23082 kJ (m^2 K s)$^{-1}$
$V(0)$	Initial reactor volume	2 m^3
$T_r(0)$	Initial reactor temperature	338.15 K
$m_A(0)$	Initial mass - component A	2000 kg
$m_B(0)$, $m_C(0)$, $m_P(0)$, $m_E(0)$, $m_G(0)$	Initial mass - components B, C, P, E, G	0 kg
C_p	Specific heat capacity	4.184 kJ kg^{-1} C^{-1}
q	Fluid density	1000 kg m^3
P_P	Price of P	555.4 \$ kg^{-1}
P_E	Price of E	125.91 \$ kg^{-1}

Table B.6
Reaction parameters and operating conditions for the diketene-pyrrole semi-batch reactor.

Variable	Definition	Value
c_B^{in}	Concentration of B in the feed	5 mol L^{-1}
$V(0)$	Initial reactor volume 0	1 L
$c_A(0)$	Initial concentration of A	0.72 mol L^{-1}
$c_B(0)$	Initial concentration of B	0.05 mol L^{-1}
$c_C(0)$	Initial concentration of C	0.08 mol L^{-1}
$c_D(0)$	Initial concentration of D	0.01 mol L^{-1}
t_f	Final time	250 min
$F_B^{\max}$	Maximal inlet flowrate	$2 \cdot 10^{-3}$ L $\min^{-1}$
$c_B^{\max}$	Maximal concentration of *B* at final time	0.025 mol L^{-1}
$c_D^{\max}$	Maximal concentration of *D* at final time	0.15 mol L^{-1}
ω	Weight	10 mol min L^{-2}

References

Andersson, J., 2013. A General-Purpose Software Framework for Dynamic Optimization. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium.

Brdys, M., Tatjewski, P., 1994. An algorithm for steady-state optimising dual control of uncertain plants. In: Proceedings of the IFAC Workshop on New Trends in Design of Control Systems, pp. 249–254.

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

Bunin, G.A., Wuillemin, Z., François, G., Nakajo, A., Tsikonis, L., Bonvin, D., 2012. Experimental real-time optimization of a solid oxide fuel cell stack via constraint adaptation. Energy 39 (1), 54–62.

Chachuat, B., Marchetti, A., Bonvin, D., 2008. Process optimization via constraints adaptation. J. Process Control 18 (3), 244–257.

Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. Comp. Chem. Eng. 33 (10), 1557–1567.

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.

Conn, A., Scheinberg, K., Vicente, L., 2009. Introduction to Derivative-Free Optimization. MOS-SIAM Series on Optimization, Philadelphia, PA, USA.

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

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

Costello, S., François, G., Bonvin, D., 2015. Directional real-time optimization applied to a kite-control simulation benchmark. In: Proceedings of the European Control Conference, pp. 1594–1601.

Forbes, J., Marlin, T., 1996. Design cost: a systematic approach to technology selection for model-based real-time optimization systems. Comp. Chem. Eng. 20 (6), 717–734.

Gao, W., Wenzel, S., Engell, S., 2016. A reliable modifier-adaptation strategy for real-time optimization. Comp. Chem. Eng. 91, 318–328. 12th Int. Symp. on Process Systems Engineering & 25th European Symp. of Computer Aided Process Engineering, Copenhagen, Denmark.

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

Marchetti, A., 2009. Modifier-Adaptation Methodology for Real-Time Optimization. Ecole Polytechnique Fédérale de Lausanne PhD thesis, # 4449.

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

Marchetti, A., Chachuat, B., Bonvin, D., 2010. A dual modifier-adaptation approach for real-time optimization. J. Process Control 20 (9), 1027–1037.

Marchetti, A., 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.

Navia, D., Briceño, L., Gutiérrez, G., De Prada, C., 2015. Modifier-adaptation methodology for real-time optimization reformulated as a nested optimization problem. Ind. Eng. Chem. Res. 54 (48), 12054–12071.

Naysmith, M.R., Douglas, P.L., 1995. Review of real-time optimization in the chemical process industries. Asia Pacific J. Chem. Eng. 3 (2), 67–87.

Rodger, E.A., Chachuat, B., 2011. Design methodology of modifier adaptation for on-line optimization of uncertain processes. IFAC Proc. Vol. 44 (1), 4113–4118.

Ruppen, D., Bonvin, D., Rippin, D.W.T., 1997. Implementation of adaptive optimal operation for a semi-batch reaction system. Comp. Chem. Eng. 22 (1–2), 185–199.

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

Salau, N.P., Trierweiler, J.O., Secchi, A.R., 2014. Observability analysis and model formulation for nonlinear state estimation. App. Math. Model. 38 (23), 5407–5420.

Singhal, M., Marchetti, A., Faulwasser, T., Bonvin, D., 2017. Improved directional derivatives for modifier-adaptation schemes. In: Proceedings of the 20th IFAC World Congress, Toulouse, France, pp. 5883–5888.

Würth, L., Hannemann, R., Marquardt, W., 2009. Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization. J. Process Control 19, 1277–1288.