Improved Directional Derivatives for Modifier-Adaptation Schemes

Martand Singhal [*], **Alejandro G. Marchetti** [*,1], **Timm Faulwasser** [*,**] **and Dominique Bonvin** [*]

[*] *Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland* {*martand.singhal, alejandro.marchetti, dominique.bonvin*}*@epfl.ch*
[**] *Institute for Applied Computer Science, Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany, timm.faulwasser@kit.edu*

Abstract: Modifier adaptation enables the real-time optimization (RTO) of plant operation in the presence of considerable plant-model mismatch. For this, modifier adaptation requires the estimation of plant gradients, which is experimentally expensive as this might involves several online experiments. Recently, a *directional* modifier-adaptation approach has been proposed, which uses the process model to compute offline a subset of input directions that are critical for plant optimization. This allows estimating directional derivatives only in the critical directions instead of full gradients, thereby reducing the burden of gradient estimation. However, in certain cases such as change of active constraints and large parametric uncertainties, directional modifier adaptation may lead to significant suboptimality. Here, we propose an extension to directional modifier adaptation, whereby, at each RTO iteration, we compute a set of critical directions that are robust to large parametric perturbations. We draw upon a simulation study of the run-to-run optimization of the Williams-Otto semi-batch reactor to illustrate the performance of the proposed extension.

Keywords: real-time optimization, modifier adaptation, directional gradient estimation

1. INTRODUCTION

The process industry aims to maximize economic benefits, while respecting safety and environmental constraints. In this endeavor, the process performance is optimized using real-time optimization (RTO). The attractive feature of RTO is that it takes into account plant-model mismatch and slow process disturbances in the optimization scheme. RTO methods typically rely on the available process model (not necessarily an exact representation of the plant) and plant measurements to steer the process to some economic optimum. In fact, the interplay between the model and the plant measurements differentiates the various RTO schemes. For instance, the traditional two-step approach (Chen and Joseph, 1987) uses plant measurements to iteratively update the model parameters and solve the updated optimization problem until convergence is reached. While this approach is inexpensive (as no additional online experiments are required), it often converges to sub-optimal points unless the considered problem satisfies stringent model-adequacy conditions (Forbes and Marlin, 1996).

On the other hand, modifier adaptation (MA) (Gao and Engell, 2005; Marchetti et al., 2009, 2016) uses plant measurements to iteratively add bias and gradient-correction terms to the modeled cost and constraint functions of the optimization problem. The key feature of MA is its guarantee of attaining plant optimality upon convergence. We refer to Marchetti et al. (2016) for an overview of MA schemes.

Unfortunately, MA requires plant gradient information at each RTO step. The estimation of plant gradients at run-time can become quite expensive with a large number of input variables. In the literature, various approaches have been proposed to make gradient estimation less expensive. Some of these approaches utilize past operating data to estimate the plant gradients (Marchetti et al., 2010; Gao et al., 2015). Unfortunately, these approaches do not scale well for high-dimensional problems.

Recently, a variant of MA known as directional modifier adaptation (DMA) has been proposed (Costello et al., 2015, 2016). DMA uses the available process model to identify a small number of *critical* input directions that are key for plant performance. Consequently, it often suffices to estimate the plant gradients in these directions, thereby making the overall approach less expensive. In DMA, the *local sensitivity* of the Lagrangian gradient is computed using the nominal model. Using singular-value decomposition (SVD) of the sensitivity matrix, one can categorize the input directions based on their sensitivities to small parametric perturbations around the nominal values. The gradients are then corrected only in the subspace spanned by the most critical (i.e., sensitive) directions (Costello et al., 2016). Note that, in DMA, this input subspace remains unchanged for all RTO iterations.

[1] Also with the French-Argentine International Center for Information and Systems Sciences (CIFASIS), CONICET-Universidad Nacional de Rosario (UNR), S2000EZP Rosario, Argentina.

The present paper goes a step further. At each RTO iteration, we evaluate an approximation of the *global sensitivity* of the Lagrangian gradient with respect to parametric uncertainties. This approach is most effective when the parametric perturbations are large. By means of simulations, we illustrate that the proposed *active* directional modifier adaptation (ADMA) scheme achieves a good trade-off between the number of selected privileged input directions and plant optimality.

The paper is structured as follows. Section 2 briefly recalls the standard MA scheme and its directional variant. Section 3 presents the proposed ADMA scheme, while Section 4 illustrates the performance of the proposed RTO scheme on a semi-batch reactor and draws a comparison with other schemes. Finally, Section 5 concludes the paper.

2. PROBLEM STATEMENT AND PRELIMINARIES

The optimization problem for the plant can be written as:

$$\min_u \Phi_p(u) := \phi(u, y_p(u)) \tag{1a}$$
$$\text{s.t.} \quad G_p(u) := g(u, y_p(u)) \preceq 0, \tag{1b}$$
$$u^L \preceq u \preceq u^U, \tag{1c}$$

where $u \in \mathbb{R}^{n_u}$ is the vector of input variables, $y_p \in \mathbb{R}^{n_y}$ are the measured output variables, $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost to be minimized, $g : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}^{n_g}$ is the vector of plant constraints. The symbol ($\preceq$) denotes component-wise inequality of vectors. We denote the optimal inputs to Problem (1) as $u_p^\star$.

The main challenge in solving the above optimization problem stems from the fact that the input-output mapping $y_p(u)$ is unknown. However, an approximate model is assumed to be available, with the input-output mapping $y(u, \theta)$, where $\theta \in \mathbb{R}^{n_\theta}$ are the model parameters. Using the model, one can approximate Problem (1) as:

$$\min_u \Phi(u, \theta) := \phi(u, y(u, \theta)) \tag{2a}$$
$$\text{s.t.} \quad G(u, \theta) := g(u, y(u, \theta)) \preceq 0, \tag{2b}$$
$$u^L \preceq u \preceq u^U. \tag{2c}$$

The optimal input $u^\star$ is found by solving Problem (2) for $\theta = \theta_0$, where θ_0 is the vector of nominal model parameters. In the presence of plant-model mismatch, the model optimum $u^\star$ may not be equal to the plant optimum $u_p^\star$. The goal of RTO is to find $u_p^\star$ by iteratively modifying and solving Problem (2).

Next, we consider RTO schemes relying on direct adaptation of the optimization problem, which are also known as modifier adaptation (Gao and Engell, 2005; Chachuat et al., 2008; Marchetti et al., 2009, 2016).

2.1 Modifier Adaptation

At the k^{th} RTO iteration, MA adds zero-order and first-order correction terms to the optimization problem (2) and solves the following modified problem:

$$\min_u \underbrace{\Phi(u, \theta) + (\lambda_k^\Phi)^T u}_{=:\Phi_{m,k}(u,\theta)} \tag{3a}$$
$$\text{s.t.} \quad \underbrace{G(u, \theta) + \varepsilon_k^G + (\lambda_k^G)^T (u - u_k)}_{=:G_{m,k}(u,\theta)} \preceq 0 \tag{3b}$$
$$u^L \preceq u \preceq u^U. \tag{3c}$$

The subscript $(.)_m$ indicates a modified quantity; $\varepsilon_k^G \in \mathbb{R}^{n_g}$ are the zero-order modifiers for the constraints, whereas $\lambda_k^\Phi \in \mathbb{R}^{n_u}$ and $\lambda_k^G \in \mathbb{R}^{n_u \times n_g}$ are the first-order modifiers for the cost and the constraint functions, respectively. At the k^{th} iteration, the modifier terms are computed as follows:

$$\varepsilon_k^G = G_p(u_k) - G(u_k, \theta), \tag{4a}$$
$$(\lambda_k^\Phi)^T = \nabla_u \Phi_p(u_k) - \nabla_u \Phi(u_k, \theta), \tag{4b}$$
$$(\lambda_k^G)^T = \nabla_u G_p(u_k) - \nabla_u G(u_k, \theta). \tag{4c}$$

Typically, a filter is applied to either the modifier terms or the input step to avoid excessive corrections that can compromise convergence (Marchetti et al., 2016)

The main feature of MA is that, upon convergence, a KKT point of Problem (1) is reached. This key feature comes at the high cost of having to estimate the plant cost and constraint gradients online, for example via finite differences. Clearly, the expense of online gradient estimation increases with the number n_u of input variables.

2.2 Gradient Estimation for Modifier Adaptation

There are different ways of estimating gradients in MA. One of the methods is dual MA, which relies on finite differences using data from past RTO iterations. There are two key requirements for finite-difference schemes: (i) ensure that there is sufficient excitation in all directions of the n_u-dimensional input space, and (ii) adjust the finite-difference steps to maintain the quality of the estimated gradients, that is, the steps should neither be too large (to reduce truncation errors) nor too small (to reduce the effect of measurement noise). Dual MA achieves both by adding extra constraints to Problem (3) (Marchetti et al., 2010). These constraints shrink the feasible set to ensure acceptable gradient estimates at the next iteration. However, dual MA suffers from the same scaling issue as standard MA in terms of the number of inputs considered in gradient estimation.

Another approach proposes to use quadratic approximations (Gao et al., 2015). Similar to dual MA, the quadratic approximation uses past data to fit a quadratic function. The quadratic approximation contains gradient and curvature information. This scheme has been shown to give good gradient estimates (Gao et al., 2015, 2016). The quadratic approximation ideas are derived from derivative-free optimization methods (Conn et al., 2009). However, the scaling issue persists.

2.3 Directional Modifier Adaptation

One can try to overcome the curse of dimensionality by relaxing the requirement of full-gradient information in MA. The properties of the model-based optimization Problem (2) can be exploited to identify a promising subset of input directions for gradient estimation. Costello et al.

(2016) proposed DMA that exploits the fact that the gradients in some input directions are more vulnerable (sensitive) to parametric uncertainty than in other directions. Hence, online gradient estimation is done only in these more vulnerable directions. To find the desired set of vulnerable directions, DMA evaluates the sensitivity of the Lagrangian gradient with respect to local parametric variations, evaluated at the nominal model optimum. To this end, one computes the sensitivity matrix $A \in \mathbb{R}^{n_u \times n_\theta}$,

$$A = \nabla_{u,\theta} L(u^\star, \mu^\star, \theta_0), \tag{5}$$

$$\text{with } L := \Phi(u,\theta) + (\mu^\star)^T G(u,\theta), \tag{6}$$

where θ_0 are the nominal model parameters, $u^\star$ is the nominal optimum, and $\mu^\star$ are the corresponding Lagrange multipliers. SVD of the matrix A gives

$$A = W \Sigma V^T,$$

where the singular values σ_i, $i = 1, \dots, n_\sigma$, are the diagonal elements of $\Sigma \in \mathbb{R}^{n_u \times n_\theta}$ and w_i are the columns of $W \in \mathbb{R}^{n_u \times n_u}$. Note that the number of singular values is $n_\sigma = \min\{n_u, n_\theta\}$.

Through the singular values of A, one can rank the input directions w_i according to their sensitivity with respect to local parametric perturbations. Assuming that there exists $n_r < n_u$ such that $\sigma_{n_{r+1}} \ll \sigma_1$, one can construct the reduced matrix $W_r \in \mathbb{R}^{n_u \times n_r}$ such that

$$W_r = [w_1\ w_2 \dots w_{n_r}], \quad \text{such that } \sigma_{n_{r+1}} \ll \sigma_1.$$

In other words, at each RTO iteration, the directional derivatives are estimated only in the directions corresponding to the columns of the matrix W_r. The directional derivatives of the plant cost function Φ_p in the n_r directions specified by W_r are

$$\nabla_{W_r} \Phi_p(u) = \left. \frac{\partial \Phi_p(u + W_r r)}{\partial r} \right|_{r=0},$$

where $\nabla_{W_r} \Phi_p \in \mathbb{R}^{1 \times n_r}$ and $r \in \mathbb{R}^{n_r}$.

The directional derivatives for the plant constraints G_p are defined in a similar way. These derivatives can be estimated by finite differences or using a duality constraint as done in dual modifier adaptation. The DMA algorithm is summarized in Algorithm 1.

In some cases, DMA can significantly reduce the expense of gradient estimation with negligible optimality loss, as shown in the simulation study of an airborne-wind energy system (Costello et al., 2016). As illustrated therein, the key strength of DMA lies in its ability to significantly reduce the number of input directions for gradient estimation. However, this approach is valid only for small uncertainty in the model parameters θ. In cases where the parametric uncertainty is large and the Lagrangian gradient is nonlinear in θ, Algorithm 1 may converge to a significantly suboptimal point. Moreover, DMA finds the Lagrangian gradient sensitivity only at the nominal model optimum $u^\star$. This sensitivity will typically change whenever the RTO iterates move away from $u^\star$, e.g. this might happen when the set of active constraints changes. Next, we propose an algorithm to overcome the limitations of local DMA.

Algorithm 1 Directional Modifier Adaptation

Step 0 (Initialization): Compute the nominal optimum using $\varepsilon_0^G = 0$, $\lambda_0^\Phi = 0$, $\lambda_0^{G_i} = 0$, and $u_0 = 0$. Set the values of the filter matrices K^ε, K^Φ, K^{G_i} (typically diagonal matrices) with eigenvalues in the interval $(0, 1]$.

for $k = 0 \to \infty$

Step 1 (Optimization): Solve the modified Problem (3):

$$\begin{aligned} u_{k+1} = \underset{u}{\text{argmin}}\ & \Phi(u,\theta) + (\lambda_k^\Phi)^T u \\ \text{s.t.}\ \ & G(u,\theta) + \varepsilon_k^G + (\lambda_k^G)^T (u - u_k) \preceq 0. \\ & u^L \preceq u \preceq u^U. \end{aligned}$$

Step 2 (Plant evaluation): Apply u_{k+1} to the plant, measure $y_p(u_{k+1})$, and compute $\Phi_p(u_{k+1}), G_p(u_{k+1})$ from $y_p(u_{k+1})$.

Step 3 (Estimation of directional derivatives): Estimate the directional derivatives $\nabla_{W_r} \Phi_{p,k+1}$ and $\nabla_{W_r} G_{p,k+1}$. Approximate the full gradients at u_{k+1} by:

$$\nabla_u \Xi_{p,k+1} = \nabla_u \Xi(u_{k+1}, \theta_0)(I_{n_u} - W_r W_r^+) + \nabla_{W_r} \Xi_{p,k+1} W_r^+,$$

where $\Xi \in \{\Phi, G_i\}$, and W_r^+ is the Moore-Penrose pseudo-inverse of W_r, and $i = 1, ..., n_g$.

Step 4 (Modifier update): Update the modifiers and include first-order filtering:

$$\begin{aligned} \varepsilon_{k+1}^G &= (I_{n_g} - K^\varepsilon)\varepsilon_k^G + K^\varepsilon (G_p(u_{k+1}) - G(u_{k+1}, \theta_0)), \\ \lambda_{k+1}^\Phi &= (I_{n_u} - K^\Phi)\lambda_k^\Phi + K^\Phi (\nabla_u \Phi_{p,k+1} - \nabla_u \Phi(u_{k+1}, \theta_0))^T, \\ \lambda_{k+1}^{G_i} &= (I_{n_u} - K^{G_i})\lambda_k^{G_i} + K^{G_i} (\nabla_u G_{p_i,k+1} - \nabla_u G_i(u_{k+1}, \theta_0))^T, \end{aligned}$$

where $i = 1, ..., n_g$.

end

3. ACTIVE DIRECTIONAL MODIFIER ADAPTATION

In general, the Lagrangian (6) is sensitive to changes in parameter values. Hence, if the actual parameters, i.e. the plant parameters, lie far from their nominal values, then the 'nominal' sensitivity of the Lagrangian model gradient may not be a good approximation to the true plant sensitivity. Algorithm 2 proposes a DMA extension that tackles this issue, labeled active directional modifier adaptation (ADMA).

To this end, as the plant parameters are unknown, we aim to approximate the global sensitivity of the Lagrangian gradient with respect to the bounded parametric uncertainty set $\Theta \subset \mathbb{R}^{n_\theta}$. Note that we assume here only parametric plant-model mismatch. We also assume that the plant and model parameters belong to a bounded uncertainty set, that is, $\theta_p, \theta_0 \in \Theta$.

Observe that Steps 0–2 are the same as in Algorithm 1. In Step 3, however, we compute the privileged directions iteratively at each RTO iteration, while, in Algorithm 1, these direction are computed only once.

In other words, at each RTO iteration, Monte-Carlo sampling is performed in the uncertainty set Θ, and the parametric sensitivity of the Lagrangian gradient is found for each parameter value. The procedure is illustrated in Figure 1. The Lagrangian sensitivity matrices computed

Algorithm 2 Active Directional Modifier Adaptation

Step 0 (Initialization): Compute nominal optimum using $\varepsilon_0^G = 0$, $\lambda_0^\Phi = 0$, $\lambda_0^{G_i} = 0$, and $u_0 = 0$. Set the values of the filter matrices K^ε, K^Φ, K^{G_i} (typically diagonal matrices) with eigenvalues in the interval $(0, 1]$.

for $k = 0 \to \infty$

Step 1 (Optimization): Solve the modified Problem (3):

$$\begin{aligned} u_{k+1} &= \underset{u}{\operatorname{argmin}}\ \Phi(u,\theta) + (\lambda_k^\Phi)^T u \\ \text{s.t.}\quad & G(u,\theta) + \varepsilon_k^G + (\lambda_k^G)^T(u - u_k) \preceq 0. \\ & u^L \preceq u \preceq u^U. \end{aligned}$$

Step 2 (Plant evaluation): Apply u_{k+1} to the plant, measure $y_p(u_{k+1})$, and compute $\Phi_p(u_{k+1}), G_p(u_{k+1})$ from $y_p(u_{k+1})$.

Step 3 (Computation of privileged directions): Perform Monte Carlo sampling and compute $W_{r,k+1}$ via (7)-(10).

Step 4 (Estimation of directional derivatives): Estimate the directional derivatives $\nabla_{W_{r,k+1}}\Phi_{p,k+1}$ and $\nabla_{W_{r,k+1}}G_{p,k+1}$. Approximate the full gradient at u_{k+1} by:

$$\begin{aligned} \nabla_u \Xi_{p,k+1} = \nabla_u \Xi(u_{k+1},\theta_0)(I_{n_u} - W_{r,k+1}W_{r,k+1}^+) \\ + \nabla_{W_{r,k+1}}\Xi_{p,k+1} W_{r,k+1}^+, \end{aligned}$$

where $\Xi \in \{\Phi, G_i\}$, and $W_{r,k}^+$ is the Moore-Penrose pseudo-inverse of $W_{r,k}$, and $i = 1, ..., n_g$.

Step 5: (Modifier update): Update the modifiers and include first-order filtering:

$$\begin{aligned} \varepsilon_{k+1}^G &= (I_{n_g} - K^\varepsilon)\varepsilon_k^G + K^\varepsilon(G_p(u_{k+1}) - G(u_{k+1},\theta_0)), \\ \lambda_{k+1}^\Phi &= (I_{n_u} - K^\Phi)\lambda_k^\Phi + K^\Phi(\nabla_u\Phi_{p,k+1} - \nabla_u\Phi(u_{k+1},\theta_0))^T, \\ \lambda_{k+1}^{G_i} &= (I_{n_u} - K^{G_i})\lambda_k^{G_i} + K^{G_i}(\nabla_u G_{p_{i,k+1}} - \nabla_u G_i(u_{k+1},\theta_0))^T, \end{aligned}$$

where $i = 1, ..., n_g$.

end

at the k^{th} iteration are combined into the single matrix A_k as:

$$A_k = [S_{1,k}\ S_{2,k} \dots S_{n,k}], \tag{7}$$

$$\text{with}\quad S_{i,k} = \nabla_{u,\theta} L(u_k, \mu_k, \theta_i), \quad i = 1, \dots, n, \tag{8}$$

where n is the number of Monte-Carlo samples. The matrix W_k is obtained by SVD of A_k,

$$A_k = W_k \Sigma_k V_k^T. \tag{9}$$

We choose the first n_r columns of W_k corresponding to the largest singular values such that $\sigma_{n_{r+1}} \ll \sigma_1$. Then, we construct the matrix $W_{r,k} \in \mathbb{R}^{n_u \times n_r}$ of privileged input directions as

$$W_{r,k} = [w_{1,k}\ w_{2,k} \dots w_{n_r,k}], \tag{10}$$

where $w_{i,k}$ is the i^{th} column of W_k.

Each matrix $S_{i,k}$ represents the local sensitivity of the Lagrangian evaluated with the corresponding parameters θ_i. Since A_k consists of all the $S_{i,k}$ matrices, it contains the gradient sensitivity for the whole parametric uncertainty set Θ. Hence, performing SVD on this matrix reveals the most vulnerable directions for the set Θ. In other words, as opposed to DMA, where the matrix A contained local information both for θ_0 and at $u^\star$, the matrix A_k contains

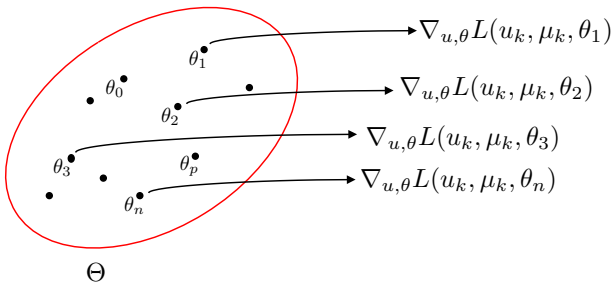

Fig. 1. Computation of the parametric sensitivity of the Lagrangian gradient for each Monte-Carlo sample in the parametric set Θ.

global information for all $\theta \in \Theta$ and local information at u_k. Note that, in order to have a good approximation of the global sensitivity in A_k, the value of n should be chosen sufficiently large. [2]

After the computation of the privileged directions, Algorithm 2 continues with the same steps as Algorithm 1, namely, one estimates the directional derivatives (Step 4) and updates the modifiers via the standard filter equations (Step 5).

Since ADMA requires Lagrangian sensitivity computation at each Monte-Carlo sample, it is computationally more expensive than DMA. However, the computational burden is not the main concern for MA schemes. Rather, the crucial barrier to overcome in applications is a significant reduction in the total number of required plant evaluations.We remark that the sensitivity information in A_k is only an approximation since, in computing $W_{r,k}$, we neglect the input directions corresponding to small singular values. Hence, estimating plant gradients only in the directions given by $W_{r,k}$ may result in gradient errors, which, in turn, may lead to sub-optimality. Also, SVD may not necessarily lead to a significant reduction in the number of privileged directions. However, as shown in the simulation study of next section, Algorithm 2 can produce powerful results in many cases. Moreover, note that one can potentially extend the ADMA Algorithm 2 to include the property of guaranteed global convergence via trust-region framework as shown in Bunin (2014), Biegler et al. (2014).

4. SIMULATION STUDY

We test the different MA-based algorithms discussed in the previous sections on the problem of run-to run optimization of the Williams-Otto semi-batch reactor described in Würth et al. (2009). The following reactions take place:

$$\begin{aligned} A + B &\xrightarrow{k_1} C \\ C + B &\xrightarrow{k_2} P + E \\ P + C &\xrightarrow{k_3} G, \end{aligned}$$

where $k_i = a_i \exp(\frac{b_i}{T_r + 273.15}), i = 1, 2, 3$. The reactant A is already present in the reactor, while the reactant B is fed to the reactor. During the exothermic reactions, the

[2] Here, for sake of simplicity, we determined heuristically a fixed number of samples such that adding more samples has only a minor effect on the singular values in Σ_k.

desired products P and E as well as the side-product G are formed. The heat generated through the exothermic reaction is removed by a cooling jacket, which is controlled by manipulating the cooling water temperature. During the batch, path constraints on the inlet flowrate of reactant B (F_{Bin}), the reactor temperature (T_r), the reactor volume (V) and the cooling water temperature (T_w) must be observed. The manipulated variables for this process are the time-varying profiles $F_{Bin}(t)$ and $T_w(t)$. The model equations and the parameter values used in this study can be found in Würth et al. (2009). The economic objective is to maximize the yield of the desired products at the final time t_f. The dynamic optimization problem is defined as follows:

$$\begin{aligned} \max_{F_{Bin}(\cdot),T_w(\cdot)} \; & J := (c_P n_P(t_f)V(t_f) + c_E n_E(t_f)V(t_f)) \\ & \text{s.t. } 0 \leq F_{Bin}(t) \leq 5.784\ kgs^{-1} \\ & V(t_f) \leq 5\ m^3 \\ & 20\ ^\circ C \leq T_w(t) \leq 100\ ^\circ C \\ & 60\ ^\circ C \leq T_r(t) \leq 90\ ^\circ C\,. \end{aligned} \tag{11}$$

For a given run, the dynamic optimization problem (11) is solved by conversion to an NLP via direct single-shooting. This is done by discretizing the problem over n_s control stages in time. For each time interval, the dynamic input variables are parametrized into low-order polynomials. We parametrized each time-varying input into $n_s = 40$ piecewise-constant inputs.

Plant-model mismatch is considered by introducing parametric uncertainty. The actual value of the parameter b_1 is unknown and lies within $\pm 20\%$ from its nominal value. In the simulations, the plant value is considered to be 10% lower than the nominal value of 6667 K. The optimal input profiles obtained by solving Problem (11) using the nominal model are shown in Figure 2. The plant outputs obtained by applying this nominal solution to the plant are shown in Figure 3. Note that the path constraint on the reactor temperature T_r is violated for the plant.

We start by implementing standard MA with full gradient estimates. Figures 4 and 5 show that the optimal input and output profiles obtained upon convergence closely match the plant optimum profiles. Although MA starts from the active constraints given by the nominal solution, it is able to identify the correct set of active constraints at the plant optimum. However, MA requires full gradient estimation. This implies that, if the gradient is identified using finite differences with added input perturbations, the plant is perturbed 80 times at each RTO iteration. Hence, one has to wait for 80 transient operations to settle between each RTO iteration, which would make the application of this type of MA on industrial-scale reactors rather prohibitive.

The DMA and ADMA algorithms are applied next. The input profiles reached upon convergence for both algorithms are shown in Figure 6, while the corresponding outputs obtained in the plant are shown in Figure 7. As seen in the figures, ADMA successfully finds the plant optimal solution, whereas DMA gives only a feasible (sub-optimal) solution. Although DMA successfully finds the optimal water temperature profile, it is unable to find the optimal profile for the feed of B. In fact, the sub-optimality for DMA is large as it gives a profit of 2.18 million \$ as opposed to the plant optimal profit of 3.14 million \$ (Figure 8). On the other hand, DMA updates the gradients only in a single direction, whereas ADMA updates the gradients in at most 4 input directions at each RTO iteration. Note that the number of privileged directions found by ADMA change from iteration to iteration as shown in Figure 9. Also for ADMA, the number of Monte-Carlo samples taken for the uncertain kinetic parameter b_1 is 250 samples. The comparison of the three schemes is summarized in the Table 1.

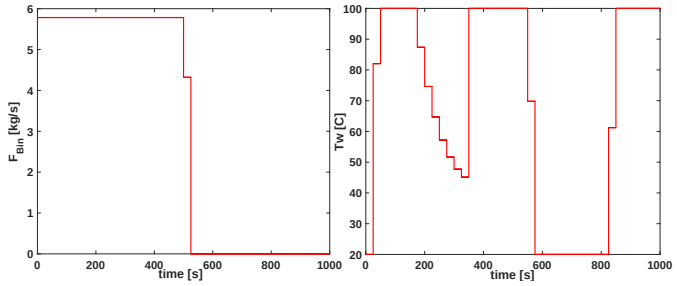

Fig. 2. Optimal input profiles (nominal model).

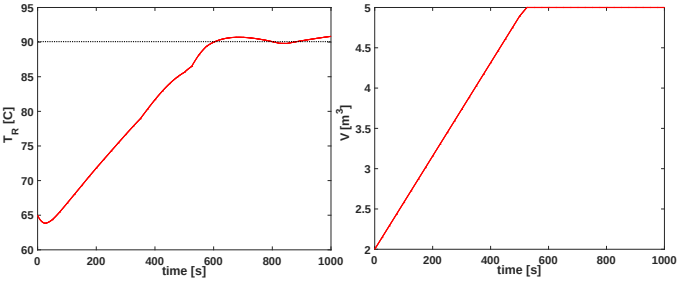

Fig. 3. Temperature and volume profiles (nominal model).

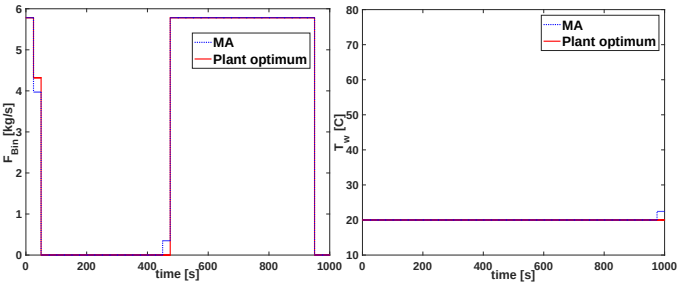

Fig. 4. Input profiles upon convergence for MA with full gradient update.

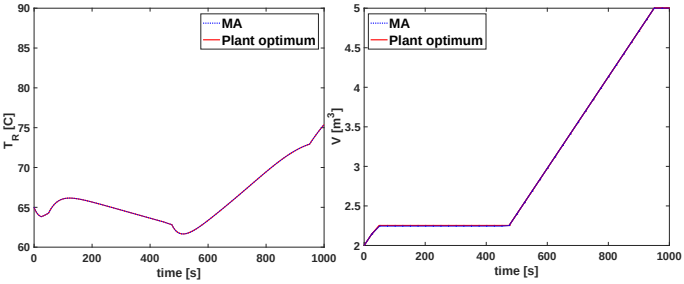

Fig. 5. Reactor temperature and volume profiles upon convergence for MA with full gradient update.

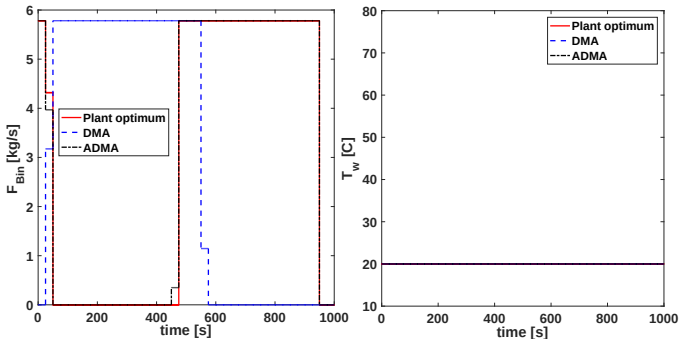

Fig. 6. Converged input profiles (DMA and ADMA).

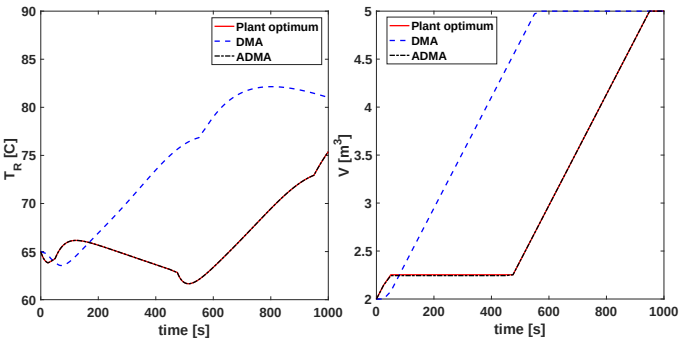

Fig. 7. Reactor temperature and volume profiles upon convergence for DMA and ADMA.

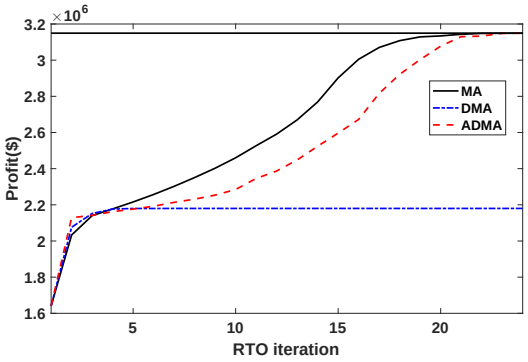

Fig. 8. Profit evolution over the RTO iterations.

Table 1. Summary of the three RTO schemes

RTO scheme	Plant profit (million $)	No. of RTO iterations	No. of gradient update directions
MA	3.14	21	80
DMA	2.18	5	1
ADMA	3.14	23	3 to 4

5. CONCLUSIONS

The paper has discussed a novel variant of modifier adaptation, namely, active directional modifier adaptation. At each RTO step, the proposed scheme applies parametric perturbation analysis on the Lagrangian gradient of the model to determine privileged input directions, that is, those directions that profit most from gradient estimation. This way, it is possible to significantly reduce the number of directional derivatives to be estimated experimentally. In other words, the effort for gradient estimation in MA is kept low, while good process performance can still be attained. A simulation study of a semi-batch reactor underpins that the proposed scheme can outperform standard DMA in terms of optimality, while being less expensive than standard MA.

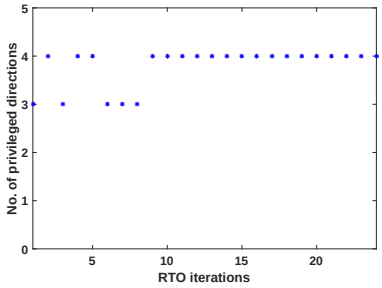

Fig. 9. Number of privileged directions with ADMA.

REFERENCES

Biegler, L., Lang, Y.D., and Lin, W. (2014). Multi-scale optimization for process systems engineering. *Comput. Chem. Engng*, 60, 17–30.

Bunin, G.A. (2014). On the equivalence between the modifier-adaptation and trust-region frameworks. *Comput. Chem. Engng*, 71, 154 – 157.

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

Chen, C.Y. and Joseph, B. (1987). On-line optimization using a two-phase approach: An application study. *I&EC Research*, 26(9), 1924–1930.

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

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

Costello, S., François, G., and Bonvin, D. (2015). Directional real-time optimization applied to a kite-control simulation benchmark. In *Control Conference (ECC), 2015 European*, 1594–1601. IEEE.

Forbes, J. and Marlin, T. (1996). Design cost: A systematic approach to technology selection for model-based real-time optimization systems. *Comput. Chem. Engng*, 20(6), 717–734.

Gao, W. and Engell, S. (2005). Iterative set-point optimization of batch chromatography. *Comput. Chem. Engng*, 29(6), 1401–1409.

Gao, W., Wenzel, S., and Engell, S. (2015). Modifier adaptation with quadratic approximation in iterative optimizing control. In *Proceedings of the 2015 European Control Conference*.

Gao, W., Wenzel, S., and Engell, S. (2016). A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Engng*, 91, 318–328.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *I&EC Research*, 48(13), 6022–6033.

Marchetti, A., Chachuat, B., and 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., and Bonvin, D. (2016). Modifier adaptation for real-time optimization – Methods and applications. *Processes*, 4(55). doi:10.3390/pr4040055.

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