Directional modifier adaptation based on input selection for real-time optimization

Gabriel D. Patrón[a], Luis Ricardez-Sandoval[a]*

[a]Department of Chemical Engineering, University of Waterloo, Waterloo, ON, Canada N2L 3G1

*Corresponding author: e-mail: laricard@uwaterloo.ca, phone: (+1) 519 888 4567 x38667, fax, (+1) 519 888 4347

Abstract

Modifier adaptation (MA) is commonly used for economic optimization of systems under model uncertainty. In MA, gradient correction terms require estimation through perturbations, thus delaying the optimization procedure. A directional modifier adaptation method is proposed whereby a subset of available gradient corrections are made. An algorithm that evaluates possible adaptation strategies and chooses those with the largest economic effect is proposed, thereby allowing economical operation with less delay. The proposed scheme, named dMAIS, is deployed on the Williams-Otto process where it is found to outperform MA if not inhibited by filtering. Systems can also suffer from constraint violation if uncertainty is present, hampering safety and profitability. An adjustment step is proposed as part of dMAIS, whereby gradients are used to drive the plant to constraint satisfaction. The adjustments are studied in an evaporator case with a product quality constraint whereby dMAIS is shown to violate the constraint infrequently leading to higher throughput. The proposed approach was also compared to standard directional modifier adaptation in the evaporator case study, where it was found to be economically beneficial. The benefits of dMAIS are observed most salient for systems with increasing disturbance frequency.

Keywords: Real-time optimization; Modifier adaptation; Plant-model mismatch; Frequent disturbances

*Corresponding author: e-mail: laricard@uwaterloo.ca, phone: (+1) 519 888 4567 x38667, fax, (+1) 519 888 4347

1. Introduction

The increasing interconnectedness of supply chains is necessitating a shift to real-time updates of operating strategies in the chemical and processing industries; this is facilitated by growing computational capabilities and the digitalization of manufacturers to aid in decision-making. As profitability is among the main drivers of these industries (i.e., along with sustainability), the focus of such computational tools has shifted to economic optimization, whereby process operating conditions can be determined such that they result in cost optimality subject to changes in upstream disturbances or downstream product demand. Model-based steady-state economic optimization is a particularly appealing strategy as existing nonlinear programming (NLP) methods can be deployed with complex process models, yielding solutions that can be implemented online in principle. The models used for these strategies serve as a representation of the actual plant that replicates its behaviour; ideally, the plant and model are perfectly matched such that their optimums are equivalent. In practice, plant and model optimums are mismatched as simplifications, assumptions, and estimates are required in the construction of process models; these are known as model uncertainties, which must be addressed to ensure the plant operation is indeed optimal.

Several techniques exist to abate or hedge against model uncertainty in the online economic optimization literature; they can be broadly categorized into robust and adaptive approaches. Robust approaches, which include stochastic optimization (Zhang et al., 2002), can be used to compute operating points that are optimal for a set of potential uncertain parameter eventualities; however, these techniques result in performance loss as they find a robust solution that will be good regardless of the actual uncertainty realization and are not truly optimal for any single individual realization. In contrast, adaptive approaches (Chachuat et al., 2009) use online measurements to modify the process model such that it matches the plant. Chiefly among the adaptive approaches is the two-step real-time optimization (RTO) scheme (Chen and Joseph, 1987), which reconciles the model and plant prior to the economic optimization procedure through a parameter estimation procedure. The estimated parameters, which often capture process phenomena (e.g., thermodynamic activities, activation energies), are subsequently provided to the RTO such that economic optimization problem is solved with the updated model. Two-step RTO requires there to be minimal structural mismatch such that the plant and model optimality can be reconciled through the parameters alone; this is restrictive in cases where the model does not consider all the process phenomena (e.g., when it is simplified and not mechanistic) (Darby et al., 2011). Models used for the two-step approach must also fulfil the adequacy conditions outlined by Forbes and Marlin (1994, 1996). These state that the estimation/optimization problems must meet gradient/Hessian necessary conditions, which lead to plant optimality in the presence of constraints (i.e., KKT matching); however, this may not be the case in practice. Increasingly, however, modifier adaptation (MA) (Gao and Engell, 2005; Marchetti et al., 2009) and its many variants (Marchetti et al., 2010; del Rio Chanona et al., 2021) are being investigated for situations of structural model uncertainty where robust and two-step approaches are not suitable. A comprehensive review of MA can be found in Marchetti et al (2016).

Instead of adapting model parameters, MA adapts the economic optimization problem via its objective function and inequality constraints. By introducing 0th order bias terms and 1st order gradient modifications with respect to the decision variables, MA has been proven to match plant and model KKT conditions (Marchetti et al., 2009). Assuming full state accessibility, the bias terms are straightforward to compute; however, the gradient estimation is more intensive as it requires system perturbations. Imposing perturbations on the system requires small changes to be made on the input variables such that they produce a correspondingly small changes in the output variables in the neighbourhood of the current operating point. Note that perturbations can also refer to disturbances; henceforth we only use perturbations only to refer to small user-defined input changes. The gradient near the current operating point can then be estimated as the ratio of output to input perturbations, which allows for gradient modifiers to be computed and the process operating point to be updated. The gradient modifiers are then recomputed at the newly defined operating point as the local gradient changes with new operating points. This refinement process of updating operating points and gradients is repeated until the true (i.e., plant) operating point is reached by the model. If a disturbance occurs, the gradient computation and modifier refinement process can detect this mismatch such that the new plant optimal operating point is found again. Each input perturbation requires the system to undergo dynamic operation until the perturbed state is reached, thus delaying the operating point update. This can become detrimental if: i) there are many inputs such that many perturbations must occur; ii) the process dynamics are slow such that the gradient estimation is time consuming; and iii) the process disturbances occur at a high frequency. These conditions affect the amount of time it takes to converge to plant optimality. For instance, if disturbances are occurring frequently, the modifier refinement process may be interrupted before convergence to the optimum. Typically, layered RTO approaches are generally deployed for high-frequency disturbance scenarios (Bottari et al., 2020), whereby different timescales have individuated control schemes. More broadly, if the dynamics are slow, there are numerous inputs, or the disturbances occur at a high frequency, the system will not converge to an optimal solution in time to accrue the benefits of the true optimum. These represent the main weakness of the existing MA algorithms as constructed and deployed in the literature (Marchetti et al., 2016). A few MA variants have been proposed to circumvent the perturbation delay. Dual MA (Marchetti et al., 2010) has been proposed to estimate gradients using past operating points whereby new successive operating points are placed such that they contain sufficient information for gradient estimation. Gao et al., (2016) proposed the use of local approximations of the cost and constraint functions, which could be differentiated to produce gradient approximations. Some studies have taken the approach of using transient measurements to speed up the MA procedure (e.g., de Avila Ferreira et al., 2017; Marchetti et al., 2020). These generally use neighboring extremals, which assume that the uncertainty is parametric (François and Bonvin, 2014).

Most notably, directional MA (dMA; Costello et al., 2016), which updates the cost and constraint gradients according to "privileged" input directions chosen through sensitivity analysis of the Lagrangian function, has been proposed. dMA does not ensure KKT matching but ensures that the cost cannot be improved further in the privileged directions upon its convergence. dMA requires the knowledge of which parameters are uncertain, their distributions, and the sensitivity of the optima to these parameters. The main difficulty that arises when using dMA is the requirement of model Lagrangian cross derivatives with respect to inputs and uncertain parameters. Model derivatives are acquired through a perturbation process or analytically; however, these are only computed once and generate local sensitivities. Costello et al. (2016) compute their privileged directions based on local model sensitivities with respect to inputs and uncertain parameters. In reality, structurally mismatched problems may not contain uncertain parameters. Even for cases with uncertain parameters, their distributions are unlikely to be known *a priori* and their sensitivities are unlikely to be the same across all potential operational points. Singhal et al. (2018) and Rodriguez et al. (2022) present a method to compute global (instead of local) parameter sensitivities; this allows for changing privileged directions at different operating points, which yields more flexibility to the dMA method. These global sensitivity methods also require parameter probability densities to be known *a priori*, which enable parameter Monte Carlo sampling. In addition to the assumptions regarding *a priori* knowledge of the uncertain parameters, these dMA methods compute directions based on the proposed predictive model, (i.e., not plant quantities), hence the directions that are privileged for the model may not necessarily be suitable for the actual plant. Despite this progress on gradient estimation and many-input systems, no prior MA scheme is aimed at frequent periodic disturbances.

Another issue in MA is that of constraint satisfaction during modifier refinement. MA only guarantees satisfaction upon convergence; however, satisfaction is not guaranteed in the modifier refinement iterations. Bunin et al. (2011) presented a method to determine upper bounds on filter gains such that satisfaction is guaranteed. However, limiting the filter gain may slow convergence speeds. Previous studies have also proposed schemes to ensure feasible-side convergence, whereby each iterate is guaranteed to be constraint-satisfying (Marchetti et al., 2017a). These require the constraint and objective function be made strictly convex upper-bounding functions via additional quadratic terms; to do so, the estimation of Hessian matrices is needed, which may be impractical. Furthermore, Marchetti et al. (2017b) also deployed robust constraint upper bounds, which result in backoff from the true constraint but ensure iteration feasibility in the presence of gradient uncertainty; this scheme also requires process Hessians. A gap exists in the literature for an MA constraint-satisfaction scheme that enables the use of little filtering and does not require Hessian information, which is difficult to acquire in practice.

In this work, we propose an MA variant for frequently disturbed periodic systems. Instead of adapting the MA problem with respect to all process inputs modifiers, the subset of modifiers is chosen that have the largest economic effects on the operating point. The proposed approach is shown to be a special case of dMA where the modified directions are limited to only include single inputs. While past approaches like dMA have used dimensionality reduction to address plant-model mismatch in many-input systems; we propose that dimensionality reduction can also be used in frequently-disturbed periodic systems, which has not been previously investigated in the context of dMA. The decreased experimental burden enabled by dimensionality reduction enables quicker action in the proposed approach. Moreover, an ancillary optimization problem is also proposed, which uses available plant and gradient information to drive the system to constraint adhering regions between MA iterations. To the authors' knowledge, the work presented in this study is the first dMA scheme to choose modification directions based on both model and plant (as opposed to only model) economics; thus, choosing modification directions based on plant knowledge (i.e., not solely based on model quantities). Moreover, it is the first dMA approach to address the effect of frequent periodic disturbances. This is also the first MA study in which constraint satisfaction during modifier refinement has been addressed through an optimal approach. Algorithms are outlined to integrate the directional modification and constraint satisfaction problems into a joint scheme referred from henceforth as directional modifier adaptation with input selection (dMAIS). The dMAIS algorithms determine: 1) which inputs to use for adaptation such that perturbation time is reduced, 2) when and by how much to adjust the operating point to ensure constraint satisfaction, 3) the number of directions to modify for a given disturbance frequency. Using the proposed method, the system can approach the economic optimum before the occurrence of a new disturbance while improving constraint satisfaction at each iteration. The proposed scheme is evaluated through two case studies, which investigate the effect of number of modifiers, disturbance period, and filtering, as well as providing a comparison to the standard dMA scheme on the basis of process cost and constraint satisfaction.

This work is structured as follows: section 2 reviews the standard MA formulation, implementation procedure, and assumptions in this work; section 3 builds on this formation and provides the dMAIS formulation, the constraint adjustment formulation, dMAIS properties, determination of modification directions, and corresponding algorithms; section 4 tests the dMAIS schemes in a variety of systems; and conclusion are outlined in section 5.

2. Modifier adaptation

The standard steady-state economic optimization problem is formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{u_t}} \phi_t \\ &\boldsymbol{f}(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{d_t}) = 0 \\ &\boldsymbol{g}(\boldsymbol{x_t}, \boldsymbol{u_t}, \boldsymbol{d_t}) \leq 0 \\ &\boldsymbol{u_{lb}} \leq \boldsymbol{u_t} \leq \boldsymbol{u_{ub}} \end{aligned} \tag{1}$$

where $\boldsymbol{x_t} \in \mathbb{R}^{n_x}$, $\boldsymbol{u_t} \in \mathbb{R}^{n_u}$, and $\boldsymbol{d_t} \in \mathbb{R}^{n_d}$ denote the model-predicted process states, inputs, and measured/estimated disturbances, respectively, at time t (i.e., the current time at which the solution will be conveyed to the plant). $\boldsymbol{\phi_t} \in \mathbb{R}$ denotes economic objective function (e.g., steady-state cost, energy consumption); in this work we take the convention of minimization, however maximization is equally valid through the requisite reformulations. $\boldsymbol{f}: \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_x}$ denotes steady-state process model, which maps the disturbances and inputs to the states (this model must fulfil the adequacy conditions like a positive definite Lagrangian Hessian matrix as outlined in Marchetti et al. (2009)). $\boldsymbol{u_l} \in \mathbb{R}^{n_u}$ and $\boldsymbol{u_u} \in \mathbb{R}^{n_u}$ denote the lower and upper bounds, respectively, for the process inputs. $\boldsymbol{g}: \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^{n_g}$ denotes the process inequality constraints (e.g., grade requirements, safety specifications). Formulation (1) does not address uncertainty and is susceptible to model inaccuracies as $\boldsymbol{f}$ may not fully match the true plant $\boldsymbol{f_p}$. Accordingly, MA adjusts the inequality constraints as follows:

$$\boldsymbol{g_{MA,t}} \coloneqq \boldsymbol{g}(\boldsymbol{u_t}, \boldsymbol{d_t}) + \boldsymbol{\varepsilon_{g,t}} + \boldsymbol{\mu}_{\boldsymbol{g,t}}^T(\boldsymbol{u_t} - \boldsymbol{u_{t-1}}) \tag{2}$$

where $\boldsymbol{\varepsilon_{g,t}} \in \mathbb{R}^{n_g}$ are 0th order modifiers (i.e., bias terms) and $\boldsymbol{\mu_{g,t}} \in \mathbb{R}^{n_u \times n_g}$ are 1st order modifiers (i.e., gradient correction terms). Moreover, $\boldsymbol{u_{t-1}} \in \mathbb{R}^{n_u}$ denotes the inputs from the previous MA execution with which the plant is operating prior to solving the updated MA problem. The 1st order modifiers, as will be explained later in this section, capture the difference between plant and model gradients (i.e., gradient error); hence the use of an input difference in equation (2). Additionally, MA modifies the objective function as follows:

$$\phi_{MA,t} \coloneqq \phi_t + \boldsymbol{\mu}_{\boldsymbol{\phi,t}}^T \boldsymbol{u_t} \tag{3}$$

where $\boldsymbol{\mu_{\phi,t}} \in \mathbb{R}^{n_u}$ are 1st order modifiers. Note that the objective function is only adapted in the constraint gradients with respect to the decision variables (as opposed to the difference between the inputs and previous inputs). This occurs as objective bias terms ($\boldsymbol{\varepsilon_{\phi,t}}$) and modifiers with respect to prior inputs ($\boldsymbol{\mu}_{\boldsymbol{\phi,t}}^T \boldsymbol{u_{t-1}}$) would be constant terms, thus would not contribute to the objective function as they would contribute to the feasible region via the constraints in equation (2) (Marchetti et al., 2016).

Plant quantities are denoted with subscript p while model quantities are denoted with the subscript m. The 0th order modifiers are the difference between the plant and model constraint predictions at the pre-update operating point defined by the previous MA iteration, defined as follows:

$$\boldsymbol{\varepsilon}_{\boldsymbol{g},\boldsymbol{t}} = \boldsymbol{g}_{\boldsymbol{p},\boldsymbol{t-1}} - \boldsymbol{g}_{\boldsymbol{m},\boldsymbol{t-1}} \tag{4}$$

where $\boldsymbol{g}_{\boldsymbol{p},\boldsymbol{t-1}} \in \mathbb{R}^{n_g}$ and $\boldsymbol{g}_{\boldsymbol{m},\boldsymbol{t-1}} \in \mathbb{R}^{n_g}$ denote the plant and model constraints under the inputs ($\boldsymbol{u}_{\boldsymbol{t-1}}$) from the previous MA execution. Similarly, the 1st order modifiers are the difference between plant and model gradient predictions at the current time, i.e.,

$$\boldsymbol{\mu}_{\boldsymbol{g},\boldsymbol{t}} = \boldsymbol{J}_{\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{u}_{\boldsymbol{t-1}})} - \boldsymbol{J}_{\boldsymbol{g}_{\boldsymbol{m}}(\boldsymbol{u}_{\boldsymbol{t-1}})} \tag{5}$$

$$\boldsymbol{\mu}_{\boldsymbol{\phi},\boldsymbol{t}} = \nabla_{\boldsymbol{u}_{\boldsymbol{t-1}}}\phi_p - \nabla_{\boldsymbol{u}_{\boldsymbol{t-1}}}\phi_m \tag{6}$$

where $\nabla_{\boldsymbol{u}_{\boldsymbol{t-1}}}$ denotes the gradient operator applied to scalar-valued plant and model economic functions (ϕ_p and ϕ_m , respectively) at the operating point corresponding to $\boldsymbol{u}_{\boldsymbol{t-1}}$. $\boldsymbol{J}_{\boldsymbol{g}_{\boldsymbol{p}}(\boldsymbol{u}_{\boldsymbol{t-1}})} \in \mathbb{R}^{n_g \times n_u}$ and $\boldsymbol{J}_{\boldsymbol{g}_{\boldsymbol{m}}(\boldsymbol{u}_{\boldsymbol{t-1}})} \in \mathbb{R}^{n_g \times n_u}$ denote, respectively, the Jacobian matrices of the plant and model constraints with respect to the inputs at $\boldsymbol{u}_{\boldsymbol{t-1}}$. The modifiers in equations (5) and (6) are calculated by perturbing the inputs around the operating point (i.e., corresponding to $\boldsymbol{u}_{\boldsymbol{t-1}}$) and using a gradient estimation method. For this study, finite difference approximation (FDA) is used as it has been found to perform adequately in past studies (Marchetti et al., 2016); however, other techniques exist, which have been previously compared in the literature (Mansour and Ellis, 2003). Accordingly, the gradients from FDA, which populate the Jacobian and gradients in equations (5) and (6) are defined as follows:

$$\frac{\partial g_i}{\partial u_j} = \frac{g_{i,j,pert} - g_{i,nom}}{\delta u_j} \qquad \forall i \in \{1, \dots, n_g\}, \quad \forall j \in \{1, \dots, n_u\} \tag{7}$$

$$\frac{\partial \phi}{\partial u_j} = \frac{\phi_{j,pert} - \phi_{nom}}{\delta u_j} \qquad \forall j \in \{1, \dots, n_u\} \tag{8}$$

where δu_j denotes a small change (i.e., a perturbation) in the jth input u_j. Note that the subscript $pert$ and nom refer to the perturbed (i.e., post-perturbation) and nominal (i.e., pre-perturbation) quantifies, respectively.

Furthermore, modifiers are also passed through first-order filters to abate the effect of measurement noise and ensure a smooth convergence to the true plant optimum; this also prevent sudden operating point changes that may be impractical from an instrumentation perspective (i.e., overly aggressive control actions). The filters are defined as follows:

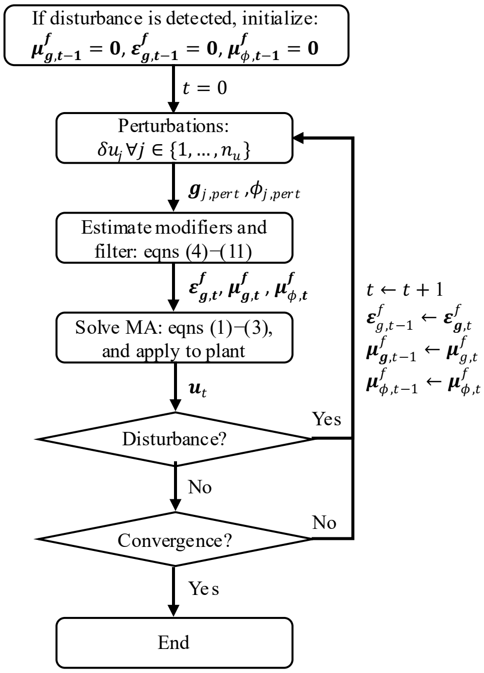
$$\boldsymbol{\varepsilon}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t}} = (\boldsymbol{I}_{\boldsymbol{n}_{\boldsymbol{g}}} - \boldsymbol{\lambda}_{\boldsymbol{\varepsilon}})\boldsymbol{\varepsilon}^{\boldsymbol{e}}_{\boldsymbol{g},\boldsymbol{t}} - \boldsymbol{\lambda}_{\boldsymbol{\varepsilon}}\boldsymbol{\varepsilon}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t-1}} \tag{9}$$

$$\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t}} = (\mathbf{1}_{n_g \times n_u} - \boldsymbol{\lambda}_{\boldsymbol{g}})\odot\boldsymbol{\mu}^{\boldsymbol{e}}_{\boldsymbol{g},\boldsymbol{t}} - \boldsymbol{\lambda}_{\boldsymbol{g}}\odot\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t-1}} \forall j \in \{1, \dots, n_u\} \tag{10}$$

$$\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{\phi},\boldsymbol{t}} = (\boldsymbol{I}_{\boldsymbol{n}_{\boldsymbol{\phi}}} - \boldsymbol{\lambda}_{\boldsymbol{\phi}})\boldsymbol{\mu}^{\boldsymbol{e}}_{\boldsymbol{\phi},\boldsymbol{t}} - \boldsymbol{\lambda}_{\boldsymbol{\phi}}\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{\phi},\boldsymbol{t-1}} \tag{11}$$

where $\boldsymbol{\lambda}_{\boldsymbol{\varepsilon}} \in \mathbb{R}^{n_g \times n_g}$ and $\boldsymbol{\lambda}_{\boldsymbol{\phi}} \in \mathbb{R}^{n_u \times n_u}$ are diagonal weighting matrices, which act on their respective modifers via matrix multiplication. The e and f superscripts denote estimated (via FDA) and filtered modifiers, respectively. Moreover, $\boldsymbol{\lambda}_{\boldsymbol{g}} \in \mathbb{R}^{n_g \times n_u}$ is a nonzero weighting matrix and $\mathbf{1}_{n_g \times n_u} \in \mathbb{R}^{n_g \times n_u}$ is a matrix of ones; these act on its modifiers via the element-wise multiplication $\odot$. The elements of the filter matrices $\lambda \in [0,1)$ are user-defined tunable parameters that determines the rate of convergence of the MA scheme.

Figure 1 illustrates the standard MA procedure whereby the filtered modifiers are initialized at zero, n_u perturbations occur, modifiers are calculated and filtered, the operating point is updated, and iterative refinement of the modifiers occurs.



If disturbance is detected, initialize:
$\boldsymbol{\mu}^f_{g,t-1} = \mathbf{0}, \boldsymbol{\varepsilon}^f_{g,t-1} = \mathbf{0}, \boldsymbol{\mu}^f_{\phi,t-1} = \mathbf{0}$
$t = 0$
Perturbations:
$\delta u_j \forall j \in \{1, \dots, n_u\}$
$\boldsymbol{g}_{j,pert}, \phi_{j,pert}$
Estimate modifiers and filter: eqns (4)–(11)
$\boldsymbol{\varepsilon}^f_{g,t}, \boldsymbol{\mu}^f_{g,t}, \boldsymbol{\mu}^f_{\phi,t}$
Solve MA: eqns (1)–(3), and apply to plant
$\boldsymbol{u}_t$
$t \leftarrow t+1$
$\boldsymbol{\varepsilon}^f_{g,t-1} \leftarrow \boldsymbol{\varepsilon}^f_{g,t}$
$\boldsymbol{\mu}^f_{g,t-1} \leftarrow \boldsymbol{\mu}^f_{g,t}$
$\boldsymbol{\mu}^f_{\phi,t-1} \leftarrow \boldsymbol{\mu}^f_{\phi,t}$
Disturbance?
Yes
No
Convergence?
No
Yes
End

Figure 1: Depiction of the standard MA algorithm.

The standard MA procedure as depicted in Figure 1, and all the methods presented hereafter, are subject to the following assumptions for this work:

1. The plant experiences periodic disturbance (i.e., they occur at fixed time intervals).
2. Uncertain model parameters and their distributions are unknown as the plant only contains structural mismatch.
3. Disturbances can be readily detected.

Assumption 1 is applicable to many plants whereby inlet raw material grades are updated on a regular basis. These are treated as disturbances, which vary diurnally, seasonally, or according to upstream production schedules. Examples of periodic disturbances include: energy systems (fired power plants: Patrón and Ricardez-Sandoval, 2022a; industrial boilers: Yip and Marlin, 2004), chemical plants (ethylene production: Tian et al, 2013; polyamine production, distillation: Pan and Lee, 2003), biological systems (nitrification/denitrification: Kornaros et al., 2012), and agricultural systems (greenhouse: Pawlowski et al., 2011). Assumption 2 is generally the case in models that are not mechanistic whereby simplifying assumptions are made for the model to be solvable. A process operator may deliberately choose to omit phenomena from a process model to make it more parsimonious or the modeler may have formulated a mismatched model as complex analytical expressions can cause problems in optimization programs. In any case, most models have some degree of structural mismatch. Assumption 3 assumes that disturbance/steady-state detection is readily available; this means that the operating mode (i.e., transient or steady) can be ascertained at any given time. While this is a non-trivial problem, it is outside the scope of the current study. Examples of disturbance/steady-state detection use test statistics (Cao and Rhinehart, 1995), Monte Carlo sampling (Hou et al., 2016) and Wavelet transforms (Jiang et al., 2003).

3. Directional modifier adaptation with input selection(dMAIS)

A new variant on the standard MA scheme outlined in section 2 is proposed whereby only some input modifiers are continually refined; thus, achieving quicker action in the presence of frequent periodic disturbances. This scheme is denoted as directional modifier adaptation with input selection (dMAIS) and is depicted in Figure 2.

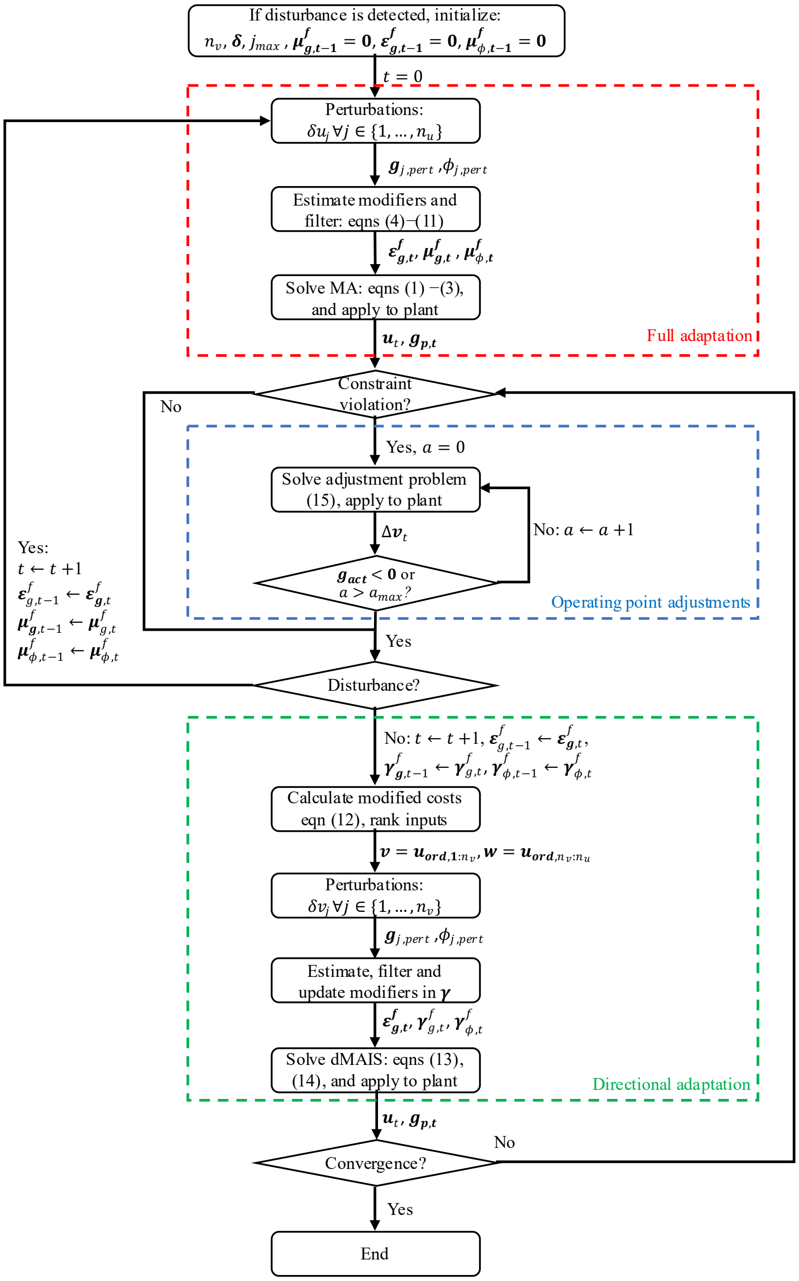

If disturbance is detected, initialize:
n_v, $\boldsymbol{\delta}$, j_{max} , $\boldsymbol{\mu}^f_{g,t-1} = \mathbf{0}$, $\boldsymbol{\varepsilon}^f_{g,t-1} = \mathbf{0}$, $\boldsymbol{\mu}^f_{\phi,t-1} = \mathbf{0}$
$t = 0$
Perturbations:
$\delta u_j \forall j \in \{1, \ldots, n_u\}$
$\boldsymbol{g}_{j,pert}$,$\phi_{j,pert}$
Estimate modifiers and filter: eqns (4)–(11)
$\boldsymbol{\varepsilon}^f_{g,t}$, $\boldsymbol{\mu}^f_{g,t}$, $\boldsymbol{\mu}^f_{\phi,t}$
Solve MA: eqns (1) −(3), and apply to plant
$\boldsymbol{u}_t$, $\boldsymbol{g}_{p,t}$
Full adaptation
Constraint violation?
No
Yes, $a = 0$
Solve adjustment problem (15), apply to plant
$\Delta \boldsymbol{v}_t$
No: $a \leftarrow a + 1$
$\boldsymbol{g}_{act} < \mathbf{0}$ or $a > a_{max}$?
Operating point adjustments
Yes:
$t \leftarrow t + 1$
$\boldsymbol{\varepsilon}^f_{g,t-1} \leftarrow \boldsymbol{\varepsilon}^f_{g,t}$
$\boldsymbol{\mu}^f_{g,t-1} \leftarrow \boldsymbol{\mu}^f_{g,t}$
$\boldsymbol{\mu}^f_{\phi,t-1} \leftarrow \boldsymbol{\mu}^f_{\phi,t}$
Yes
Disturbance?
No: $t \leftarrow t + 1$, $\boldsymbol{\varepsilon}^f_{g,t-1} \leftarrow \boldsymbol{\varepsilon}^f_{g,t}$,
$\boldsymbol{\gamma}^f_{g,t-1} \leftarrow \boldsymbol{\gamma}^f_{g,t}$, $\boldsymbol{\gamma}^f_{\phi,t-1} \leftarrow \boldsymbol{\gamma}^f_{\phi,t}$
Calculate modified costs eqn (12), rank inputs
$\boldsymbol{v} = \boldsymbol{u}_{\boldsymbol{ord},\mathbf{1}:n_v}$, $\boldsymbol{w} = \boldsymbol{u}_{\boldsymbol{ord},n_v:n_u}$
Perturbations:
$\delta v_j \forall j \in \{1, \ldots, n_v\}$
$\boldsymbol{g}_{j,pert}$,$\phi_{j,pert}$
Estimate, filter and update modifiers in $\boldsymbol{\gamma}$
$\boldsymbol{\varepsilon}^f_{g,t}$, $\boldsymbol{\gamma}^f_{g,t}$, $\boldsymbol{\gamma}^f_{\phi,t}$
Solve dMAIS: eqns (13), (14), and apply to plant
Directional adaptation
$\boldsymbol{u}_t$, $\boldsymbol{g}_{p,t}$
No
Convergence?
Yes
End

Figure 2: Depiction of the dMAIS algorithm.

3.1. dMAIS formulations and algorithm

As shown in Figure 2, there are three distinct operations that can occur within the proposed algorithm. Prior to the first dMAIS iteration, initialization occurs whereby the filtered modifier values are defined as zero and the number of gradient modifiers n_v is chosen; the choice of this term is discussed in section 3.3. Furthermore, we introduce $\boldsymbol{\alpha}$ and a_{max}, which are defined for the operating point adjustment at the outset; these are discussed in depth later. For $t = 0$ the MA procedure proceeds as usual through the full MA (i.e., red block in Figure 2); n_u perturbations are made and the corresponding modifiers $\boldsymbol{\varepsilon}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t}}$, $\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t}}$, and $\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{\phi},\boldsymbol{t}}$ are computed as depicted in Figure 2.

After every operating point update, the new operating point is checked for constraint violations. If violations are detected, the algorithm proceeds through the operating point adjustment (blue block in Figure 2), which is discussed later. If no violations are present, the system is checked for any new disturbances; a new disturbance would trigger the full adaptation procedure as depicted in Figure 2, otherwise the dMAIS scheme is engaged as depicted in the green block in Figure 2.

Assuming no disturbances and no constraint violations, the dMAIS adaptation can begin after the initial iteration (i.e., $t > 0$). The modifiers corresponding to $n_v \in \{\mathbb{Z}^+ : n_v < n_u\}$ inputs are iteratively refined; these are chosen based on their impact on the economic predictions of the MA problem such that the inputs with the largest effect are chosen. This leverages the fact that not all inputs have the same effect on optimality and choosing the appropriate inputs will yield a sufficiently good operating point without taking as much perturbation time. To evaluate the impact of individual input modifiers on economic predictions, the "modified costs" are introduced herein to choose which n_v inputs modifiers should be used. Since all modifiers are calculated in the first full MA iteration, they can be used to make process predictions at the current operating point; accordingly, the modified costs are defined as follows:

$$\hat{\phi}_{j,t} \coloneqq \mu_{\phi,t,j} u_{t,j} \qquad \forall j \in \{1, \ldots, n_u\} \tag{12}$$

where the modified costs are sorted into the ordered set (i.e., sequence) $U = \{\hat{\phi}_{j,t}\}_{j=0}^{n_u}$ and $\boldsymbol{u}_{\boldsymbol{ord}} \in \mathbb{R}^{n_u}$ is the corresponding vector of inputs sorted by modified cost. The modified cost in equation (12) is used to rank possible single-input modifications using the most recently available modifiers. Accordingly, the dMAIS scheme uses both plant and model information to choose the inputs with respect to which the cost gradient has the largest gradient error (i.e., the largest modifiers). This modifier is then multiplied by the most recent input value to normalize gradient with respect to the input magnitude. Accordingly, the inputs that are chosen by equation (12) are those that will lead to the largest corrections in the cost gradient. The modified cost of each input will be different owing to their distinct values and gradients. However, a situation may arise in which the difference is relatively small. Even if the small difference is owed to numerical or process noise, both inputs in question will have similar effects on plant-model mismatch so the choice of either input will have similar adaptation effects on the system.

With the minimization convention, the first n_v elements of $\boldsymbol{u}_{\boldsymbol{ord}}$ are stored in the vector $\left(\boldsymbol{v} = \boldsymbol{u}_{\boldsymbol{ord},\boldsymbol{1}:n_M}\right) \in \mathbb{R}^{n_v}$ as these inputs yield the lowest modified costs; these correspond to the set of modified costs $V = \{U : j \leq n_v\}$. Moreover, the remaining $n_u - n_v$ elements of $\boldsymbol{u}_{\boldsymbol{ord}}$ are stored in the vector $\left(\boldsymbol{w} = \boldsymbol{u}_{\boldsymbol{ord},n_M:n_u}\right) \in \mathbb{R}^{(n_u - n_v)}$; these correspond to the set of modified costs $W = \{U : j > n_v\}$. In other words, the input variable vector is decomposed into two subvectors. $\boldsymbol{u} = [\boldsymbol{v} \quad \boldsymbol{w}]^T$ and the sequence of modified costs is such that $U = V \cup W$. The inputs in $\boldsymbol{v}$ are those whose modifiers lead to the best predicted economics (i.e., the least predicted costs); thus, only the modifiers corresponding to $\boldsymbol{v}$ are adapted with respect to in the next dMAIS execution. Thus, the objective function becomes:

$$\phi_{pMA,t} \coloneqq \phi_t + \boldsymbol{\gamma}_{\boldsymbol{\phi},t}^T \boldsymbol{v_t} + \boldsymbol{\omega}_{\boldsymbol{\phi},t}^T \boldsymbol{w_t} \tag{13}$$

where $\boldsymbol{\gamma_{\phi,t}} \in \mathbb{R}^{n_v}$ and $\boldsymbol{\omega_{\phi,t}} \in \mathbb{R}^{(n_u-n_v)}$ are the 1st order objective function modifiers corresponding to $\boldsymbol{v}$ and $\boldsymbol{w}$, respectively. Accordingly, like the input vector, the vector of objective function modifiers is decomposed into two subvectors $\boldsymbol{\mu_\phi} = [\boldsymbol{\gamma_\phi} \quad \boldsymbol{\omega_\phi}]^T$.

Likewise, the adapted constraints become:

$$\boldsymbol{g_{pMA,t}} \coloneqq \boldsymbol{g}(\boldsymbol{u_t}, \boldsymbol{d_t}) + \boldsymbol{\varepsilon_{g,t}} + \boldsymbol{\gamma}_{\boldsymbol{g},t}^T(\boldsymbol{v_t} - \boldsymbol{v_{t-1}}) + \boldsymbol{\omega}_{\boldsymbol{g},t}^T(\boldsymbol{w_t} - \boldsymbol{w_{t-1}}) \tag{14}$$

where $\boldsymbol{\gamma_{g,t}} \in \mathbb{R}^{n_v \times n_g}$ and $\boldsymbol{\omega_{g,t}} \in \mathbb{R}^{(n_u-n_v)\times n_g}$ are the 1st order constraint modifiers corresponding to $\boldsymbol{v}$ and $\boldsymbol{w}$, respectively. Accordingly, like the input vector, the matrix of gradient modifiers is decomposed into two block matrices, i.e., $\boldsymbol{\mu_g} = [\boldsymbol{\gamma_g} \quad \boldsymbol{\omega_g}]^T$.

At each dMAIS iteration, the members of V and W (and their corresponding vectors $\boldsymbol{v}$ and $\boldsymbol{w}$) are refined according to the ranked modified cost sequence in equation (12) as described above. This sequence is updated using the newly updated economic function gradient modifiers ($\boldsymbol{\gamma_\phi}$) along with the outdated modifiers ($\boldsymbol{\omega_\phi}$). This allows for iterative refinement of the inputs that have the largest effect on economic function adaptation until convergence to their final modifiers and final membership of V and W. This process of refinement is depicted in the "rank inputs" block of Figure 2. We note that, the ability to change modification directions through V and the modified cost is present in dMAIS but not the standard dMA (Costello et al., 2016); the latter can yield myopic behaviour owed to its directional inflexibility.

If a disturbance is detected after any dMAIS iteration, the previous members of V and W are no longer valid as the operating point has changed and the gradients may be different in the new operating neighbourhood. This triggers the disturbance block in Figure 2 that toggles between the full and dMAIS adaptation schemes. The toggling of schemes allows for a full set of modifiers $\boldsymbol{\mu_\phi}$ and $\boldsymbol{\mu_g}$ to be computed such that an entirely new U and its corresponding $\boldsymbol{u_{ord}}$ can be found at the new operating point.

As only plant economics are considered in the input ranking equation (12), constraint satisfaction is unaddressed at each dMAIS iteration. Furthermore, constraint satisfaction upon convergence is not guaranteed when using dMAIS (i.e., no plant/model KKT matching). Even with consideration of the Lagrangian as in directional MA (Costello et al., 2016), iteration satisfaction is not guaranteed. Indeed, full MA schemes (i.e., not only dMAIS) only ensure constraint satisfaction at convergence; hence, the "path" to the optimum may be subject to iterations where violations occur (Marchetti, 2022). Thus, recourse action is needed to avoid constraint violations at dMAIS iterations and upon convergence where these violations could lead to safety or economic concerns (e.g., violation of temperature limits or production below purity specification). In systems with frequent disturbances, it is advantageous to satisfy constraints along the path as convergence to a final steady state may never be achieved.

To minimize these iterative constraint violations, operating point "adjustments" ancillary problems are proposed to be solved after every adaptation iteration where constraint violations are detected. These are depicted in the blue block of Figure 2. The adjustment problems use process measurements and the plant gradient data available from the dMAIS perturbations to formulate of a quadratic problem (QP) that reduces or altogether closes the constraint violation gap. The dMAIS section of the algorithm features adjustments occurring after the solution is applied to the plant. As shown in Figure 2, these adjustments only occur if a constraint violation is detected. Once this occurs, the following problem is solved:

$$\begin{aligned}
&\min_{\Delta \boldsymbol{v}} \|\widehat{\boldsymbol{g}}_{\boldsymbol{act}}\|_{\boldsymbol{QA}}{}^{2} \\
&\widehat{\boldsymbol{g}} = \boldsymbol{g}_{\boldsymbol{p}} + \boldsymbol{J}_{\boldsymbol{g}(\boldsymbol{v}_{\boldsymbol{t-1}})}\Delta \boldsymbol{v} \\
&\widehat{\boldsymbol{g}}_{\boldsymbol{act}} = \boldsymbol{A}\widehat{\boldsymbol{g}} \\
&\widehat{\boldsymbol{g}}_{\boldsymbol{in}} = (\boldsymbol{I}_{n_g} - \boldsymbol{A})\widehat{\boldsymbol{g}} \\
&\widehat{\boldsymbol{g}}_{\boldsymbol{in}} \leq \boldsymbol{0} \\
&\boldsymbol{Q} = diag(g_{p,1} \quad \cdots \quad g_{p,n_g}) \\
&\boldsymbol{A} = diag(a_1 \quad \cdots \quad a_{n_g}): a_i = \begin{cases} 1 & g_{p,i} > 0 \\ 0 & g_{p,i} \leq 0 \end{cases} \\
&-\boldsymbol{\alpha} \leq \Delta \boldsymbol{v} \leq \boldsymbol{\alpha}
\end{aligned} \qquad (15)$$

where $\widehat{\boldsymbol{g}}$ and $\boldsymbol{g}_{\boldsymbol{p}} \in \mathbb{R}^{n_g}$ are the linear model-predicted and current (measured) plant operating point for all constraints. $\boldsymbol{J}_{\boldsymbol{g}(\boldsymbol{v}_{\boldsymbol{t-1}})} \in \mathbb{R}^{n_g \times n_v}$ is the Jacobian of the constraints with respect to the subset of process inputs used in the dMAIS step; this is constructed using the most recent directional plant perturbation results. Using the most recently calculated plant Jacobian, a local approximation of the constraint-input relationship is generated such that small input adjustments can be computed. This differs from a constraint adaptation scheme (e.g., Chachuat et al., 2009) since it uses a linear model with a satisfaction objective as opposed to a nonlinear model with an economic objective; moreover, the adjustment step is used to compliment the dMAIS problem defined above, which acts on an economic basis. The model-predicted constraint vector is partitioned into active and inactive constraints using the matrix $\boldsymbol{A} \in \mathbb{R}^{n_g \times n_g}$ and its identity matrix difference $\boldsymbol{I} - \boldsymbol{A}$ where $\boldsymbol{I}_{n_g} \in \mathbb{R}^{n_g \times n_g}$. $\boldsymbol{A}$ contains diagonal identity elements to indicate if the plant constraint $g_{p,i} \forall i \in \{1, \dots, n_g\}$ has been violated. Using the $\boldsymbol{A}$ matrix, the inactive constraint entries are set to zero in the vector $\widehat{\boldsymbol{g}}_{\boldsymbol{act}} \in \mathbb{R}^{n_g}$; conversely, the active constraint entries are set to zero in the vector $\widehat{\boldsymbol{g}}_{\boldsymbol{in}} \in \mathbb{R}^{n_g}$. Using the inactive constraint predictions $\widehat{\boldsymbol{g}}_{\boldsymbol{in}}$, the linearized model can be used such that they remain inactive using the inequality constraint in formulation (15). Moreover, the objective function in (15) features a minimization term for the active constraints $\widehat{\boldsymbol{g}}_{\boldsymbol{act}}$ whereby their predicted value is minimized; this serves to bring their value to zero as constraints in MA are posed such that the RHS is zero. This objective is weighted by a diagonal matrix of the constraint violation magnitudes for the active constraints $\boldsymbol{QA} \in \mathbb{R}^{n_g \times n_g}$; this way larger violations are prioritized over smaller violations. The decision variable for this problem is the vector of process inputs adjustments $\Delta \boldsymbol{v} \in \mathbb{R}^{n_v}$, which are bounded by the user-specified $\boldsymbol{\alpha} \in \mathbb{R}^{n_v}$ mentioned in the initialization section above. The adjustment bounds are designed to be small through the choice of $\boldsymbol{\alpha}$, thus requiring little computational or transient time. Also required in initialization are the maximum number of constraint adjustments a_{max}; this is imposed on the scheme such that there is little delay in returning to the dMAIS loop. Accordingly, $\boldsymbol{\alpha}$ and a_{max} are user defined but should be small (i.e., since they are assumed to be adjustments and not large changes). We note that the disturbance block is checked at every iteration of the dMAIS algorithm as shown in Figure 2. Whether or not a constraint violation is detected, disturbances must be checked to accommodate for their potential effect of suddenly changing the memberships in V and W. The timeliness of this check is ensured by capping the number of adjustment iterations at a_{max}.

While the operating point adjustment problem (15) focuses solely on constraint attenuation, the main dMAIS objective in equations (13) and (14) is still to minimize plant-model mismatch through its modifiers. The reason for the additional adjustments is to decrease potential constraint violations in the modifier refinement process whereby the plant-model mismatch is not accounted for to its full possible extent within the dMAIS paradigm. Through the adjustment subproblem (15), constraint-violating operating points may be abated quickly. Firstly, the measurement of $\boldsymbol{g_p}$ serves to localize problem (15) in the current constraint-space of the plant. This measurement is updated at every adjustment problem iteration such that the local linear prediction of constraints $\widehat{\boldsymbol{g}}$ begins at the correct state. Additionally, only the inputs contained within $\boldsymbol{v}$ are used for the constraint adjustment step as only the local plant gradients for these inputs are updated as part of the dMAIS algorithm. Despite no guarantee being available for whether violation will be avoided (this would require controllability of all states via all inputs); problem (15) uses readily available information via $\boldsymbol{g_p}$ and $\boldsymbol{J}_{\boldsymbol{g}(\boldsymbol{v}_{t-1})}$ as opposed to other constraint-feasibility approaches that require additional data be estimated from the system (e.g., Hessian matrices). Note that problem (15) constitutes a discrete time one-step-ahead linear-quadratic regulator whereby no control-move suppression terms are used, and the state matrix is an identity matrix. In principle, such a linear quadratic regulator is solvable for an explicit feedback law using the dynamic Ricatti equation; however, the inactivity constraints and absence of control move suppression term prohibits this for the system shown in equation (15).

The dMAIS algorithm is summarized as follows:

Algorithm 1: dMAIS operation	
	Initialize: define n_v, $\boldsymbol{\alpha}$, a_{max}, $\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t-1}} = \mathbf{0}$, $\boldsymbol{\varepsilon}^{\boldsymbol{f}}_{\boldsymbol{g},\boldsymbol{t-1}} = \mathbf{0}$, $\boldsymbol{\mu}^{\boldsymbol{f}}_{\boldsymbol{\phi},\boldsymbol{t-1}} = \mathbf{0}$.
1.	For $t = 0$: perform full MA and apply to plant.
2.	Are constraints being violated?
a.	Yes: $a = 0$, activate constraint adjustment, go to step 3.
b.	No: proceed to step 5.
	Operating point adjustments
3.	Solve problem (15) and apply to plant.
4.	$\boldsymbol{g_{viol}} < \mathbf{0}$ or $a > a_{max}$?
a.	Yes: Proceed to step 5.
b.	No: $a += 1$, return to step 3.
5.	New disturbance?
a.	Yes: $t += 1$, activate full MA, go to step 6.
b.	No: $t += 1$, activate dMAIS, go to step 9.
	Full adaptation
6.	Perturb n_u inputs.

7.	Estimate n_u modifiers and filter.
8.	Solve full MA problem using equations (2) and (3). Return to step 2.
	Directional adaptation with input selection
9.	Perturb n_v inputs.
10.	Estimate n_v modifiers and filter.
11.	Re-evaluate “modified costs” in equation (12) and refine modifiers in $\boldsymbol{v}$.
12.	Solve dMAIS using equations (13) and (14).
13.	Has the scheme converged to an operating point
a.	Yes: end.
b.	No: continue refining modifiers and return to step 2.

The benefit to the proposed dMAIS approach is twofold: firstly, using $n_v < n_u$ input modifiers result in a faster acting scheme that prioritizes economic modification; secondly, the adjustment step will enable iterates to be constraint abiding without any additional information (e.g., Hessian matrix). On the other hand, the adjustment step in the proposed dMAIS scheme is designed to act quickly and only take small steps. Accordingly, the system may not be able to close the constraint gap if the adjustment step begins far from the constraint as the number of adjustment iterations is limited in quantity and size. Crucially, the selection of n_v is not a trivial and an algorithm that leverages disturbance periodicity to determine the number of modification directions is presented in section 3.3.

We note that, dMAIS concepts can be used in tandem with previously proposed methods. Indeed, the modified cost metric introduced in this work can be used similarly to the sensitivity matrix in standard dMA to compute privileged directions (i.e., not limited to partial single-input derivatives). Moreover, the dual methods and the use transient measurements introduced by Costello et al. (2016) and François and Bonvin (2014), respectively, can also be incorporated into the dMAIS scheme proposed in this work.

3.2. dMAIS properties

The vector $\boldsymbol{v_t}$ of inputs used in the dMAIS approach can alternatively be represented by the block matrix $\boldsymbol{V_t} \in \mathbb{R}^{n_u \times n_v}$, i.e.:

$$\boldsymbol{V_t} = \begin{bmatrix} diag(u_1, \dots, u_{n_v}) \\ \mathbf{0}_{(n_u-n_v)\times n_v} \end{bmatrix} \tag{16}$$

where $\mathbf{0}_{(n_u-n_v)\times n_v} \in \mathbb{R}^{(n_u-n_v)\times n_v}$ denotes a zero matrix. Similarly, dMA defines its directions using the block matrix $\boldsymbol{U_t} \in \mathbb{R}^{n_u \times n_v}$, i.e.:

$$\boldsymbol{U_t} = [\boldsymbol{\delta u}_1 \quad \cdots \quad \boldsymbol{\delta u}_{n_v}] \tag{17}$$

where $(\boldsymbol{\delta u}_j \in \mathbb{R}^{n_u})\ \forall j \in \{1, \dots, n_v\}$ are the vectors containing input directions whereby a subset of the inputs elements is chosen for each direction. From equations (16) and (17), it is evident that $\boldsymbol{V_t}$ is a special case of $\boldsymbol{U_t}$ whereby only the diagonal elements in the top block are used. This represents an analogue to multivariable calculus whereby partial derivatives represent a special case of directional derivatives. Since dMAIS implies a special case of dMAIS, some properties of the latter can be applied to dMAIS.

Theorem 1: (Plant optimality for chosen input adaptations). Consider the dMAIS algorithm without measurement noise and perfect estimation of plant derivatives in n_v inputs. If the algorithm converges to the fixed point $(\boldsymbol{u}_\infty, \boldsymbol{\varepsilon}_{\boldsymbol{g},\infty}, \boldsymbol{\gamma}_{\boldsymbol{g},\infty}, \boldsymbol{\gamma}_{\boldsymbol{\phi},\infty})$, this corresponds to a KKT point of the modified optimization problem in equations (13) and (14), then $\boldsymbol{u}_\infty$ will be optimal for the plant in these n_v directions.

Proof: As shown above, dMAIS is a special case of dMA. Accordingly, see Costello et al. (2018), Theorem 3.1. ∎

An advantage of dMAIS is that inputs are chosen based on readily available plant data whereby the information necessary to compute the modified costs is found as part of the dMAIS algorithm during the perturbation step. Instead of using model sensitivities with respect to uncertain parameters to determine $\boldsymbol{U_t}$, the modified cost metric in equation (12) uses the cost gradient modifiers (i.e., the plant-model cost gradient error) to determine the inputs to which the cost is most sensitive (i.e., $\boldsymbol{V_t}$). These modifiers are multiplied by the latest-acquired input values to normalize their magnitude; thus, dMAIS chooses the directions of highest normalized input error. Note that Costello et al. (2016) normalize the sensitivity matrix with uncertain parameter ranges; however, this does not abide by assumption 2 (section 2). Moreover, the approach presented offers benefits with respect to the active approach proposed by Singhal et al. (2018); namely, it does not rely on parameter uncertainty being present or access to a probability density as stated in assumption 2 (section 2). With the approach presented in this work, we only consider the cost gradient sensitivities, which are equivalent to the Lagrangian gradient in the case of no active constraints. Only cost sensitivities are considered (as opposed to Lagrangian sensitivities) because the plant Lagrange multipliers cannot be readily measured.

3.3. Disturbance periodicity and the number of modification directions

The aforementioned weaknesses in the standard MA scheme can be seen most saliently in equations (7) and (8), which depend on the index $\forall j \in \{1, \dots, n_u\}$ and correspond to the perturbation block in Figure 1. As previously noted, these perturbations delay the operating point updates as they may be time-consuming. To analyze the refinement time that the MA scheme requires, we introduce a user-defined perturbation time τ (i.e., the required to perform a single perturbation), a system-defined settling time T (i.e., the time required to reach a new operating point upon modifier refinement), and an externally defined disturbance period ΔT (i.e., the time between subsequent disturbances). The use of different τ and T reflects the fact that small perturbations (of τ duration) may not require the same settling time as an operating point change (of T duration). This occurs as perturbations are meant to be small (i.e., a fraction of an input's value) while operating point changes are potentially large (i.e., a completely different set of input values). Accordingly, if n_u, τ, or T are large, the MA refinement procedure will be time-consuming. Furthermore, if ΔT is small, convergence to the true optimum may not occur before a new disturbance is imposed. That is, if the MA scheme requires n_{MA} iterations to converge to the optimum, the following inequality must hold if convergence is to occur:

$$n_{MA} n_u \tau + n_{MA} T \leq \Delta T \tag{18}$$

However, this inequality may not be fulfilled if n_u and τ are large, or ΔT small as mentioned previously. By treating equation (18) as an equality we can express the maximum number of MA iteration necessary to reach convergence as follows:

$$n_{MA} = \frac{\Delta T}{n_u \tau + T} \tag{19}$$

This ratio is not practically useful as many of these quantities are not known *a priori*; however, it serves for theoretical discussion of the MA schemes in periodic settings. In contrast to MA equation (19), the number of iterations to reach convergence for dMAIS is defined as follows:

$$n_{dMAIS} = \frac{\Delta T}{n_v \tau + T} \tag{20}$$

since $n_v < n_u$ perturbations are performed on most iterations, a larger number of dMAIS iterations may be performed (i.e., $n_{pMA} > n_{MA}$). This results in quicker modifier refinement, which is the working principle of dMAIS. These refinements will work towards the directional optimum given the chosen modification directions; as a full set of modifiers is not refined until convergence, the dMAIS scheme will not converge to the plant KKT points as noted for dMA (Costello et al., 2016). However, the directional optimum will certainly be better than a "do-nothing" case and convergence to this optimum may occur more quickly (i.e., within a given disturbance period).

Recalling equations (19) and (20), which quantify the number of MA and dMAIS iterations that a given scheme must perform to reach convergence, we propose a scheme-independent metric to assess the efficacy of various dMAIS and MA schemes on a given system. While the number of modified inputs is scheme-dependent and the settling time is not known *a priori*, thus rendering equations (19) and (20) impractical; they elucidate how the number of iterations of each scheme is dependent on the disturbance period (i.e., $n_{pMA} = f(\Delta T)$). Thus, for a given plant, the best number of inputs modified with respect to can be expressed as piecewise function of the disturbance period:

$$n_v = \begin{cases} 1 & \Delta T \leq \zeta_1 \\ \vdots & \zeta_k < \Delta T \leq \zeta_{k+1} \\ n_u & \Delta T > \zeta_{n_u} \end{cases} \tag{21}$$

where n_v is segmented into n_u regimes such that $n_v \in \{1, \dots, n_u\}$ can be determined based on prior operation of the system. ζ_j $\forall j \in \{1, \dots, n_u\}$ are the corresponding disturbance period boundaries that define the how many modifiers j are suitable for refinement. According to equation (21), the number of inputs to be modified for (thus perturbed) could be tuned using the disturbance frequency. Determination of the regime boundaries ζ_j can be performed through preliminary system runs, whereby data are collected for various frequencies such that equation (21) can be fully defined for a given process; however, this may be impractical.

A limitation of dMA is that the number of privileged directions must be pre-specified by the user such that $n_v \leq n_\theta$. With the dMAIS approach, we leverage the periodicity of the disturbances to determine the number of inputs for adaptation as a function of the disturbance period. The following algorithm enables the systematic determination of n_v for a given disturbance period ΔT under a performance metric PM and a minimization convention.

Algorithm 2: determination of n_v for a disturbance period ΔT	
	Initialize: define $n_v = 1$, $l = 1$, $\boldsymbol{\alpha}$, a_{max}, B, $\boldsymbol{\mu}^{f}_{g,t-1} = \mathbf{0}$, $\boldsymbol{\varepsilon}^{f}_{g,t-1} = \mathbf{0}$, $\boldsymbol{\mu}^{f}_{\phi,t-1} = \mathbf{0}$.
1.	Deploy dMAIS algorithm for n_v modifiers.
2.	B disturbance elapsed?

a.	Yes: Calculate PM for B previous disturbance period, go to step 3.
b.	No: $t \leftarrow t + \Delta t$, go to step 2.
3.	$l > 2$?
a.	Yes: l+= 1, go to step 4.
b.	No: l+= 1, go to step 1.
4.	$PM_l < PM_{l-1}$?
a.	Yes: n_v+= 1, go to step 1.
b.	No: $n_v = l - 1$, end.

Algorithm 2 initial assumes that only one modification direction is being used ($n_v = 1$). The user defines an allowable computational budget B along with all other dMAIS operational parameters. Once B disturbances have elapsed, R or P_{prod} can be computed for unconstrained and constrained systems, respectively. After the initial B disturbance periods, another input direction is assumed to be available; hence n_v+= 1. This allows for comparison between the previous B and the next B disturbance periods on the basis of a user-defined PM whereby a modification dimension is added until there is no significant improvement in the process cost (assuming a minimization convention). Examples of performance metrics are provided in the next section for constrained and unconstrained MA-operated systems.

4. Case studies

The proposed scheme was tested using two case studies: the Williams-Otto CSTR (Williams and Otto, 1960) and the forced circulation evaporator (Lee et al., 1989). The former case study explored dMAIS on a two-input system such that the effect of filtering and disturbance period can be isolated and assessed on an entirely economic basis. The latter case study provides a setting in which to test dMAIS on a three-input system with active constraints such that the effect of number of adapted inputs and constraint satisfaction can be quantified on process economics and throughput, respectively. Moreover, the evaporator also provides a constrained setting in which to assess the proposed dMAIS against standard dMA. While both test systems are low-dimensional (i.e., few inputs), and the dMAIS could have effects on the performance of large plants through dimensionality reduction, we remark that the applicability of the dimensionality reduction in the dMAIS as proposed herein is in modifier refinement speed. Accordingly, both case studies are subjected to a variety of high-frequency (measurable) disturbances to which they use varying degrees of dimensionality reduction as proposed in section 3.3. Through these case studies, the economic effect of rapid modifier action is observed clearly, without the need for many inputs. The respective optimization problems and simulated plant were implemented in the Pyomo (Hart et al., 2011) environment, where the IPOPT solver (Wächter and Biegler, 2006) was used; they were performed on an Intel core i7-4770 CPU @ 3.4 GHz. The code used to run these case studies can be found in https://git.uwaterloo.ca/ricardez_group/dmais.

As stated in the introduction, we aim to improve on the aggregate performance of MA across many disturbances; hence, showing any iteration is not very instructive. Instead, we introduce the following performance metrics (PM), which summarize aggregate process performance over time.

Accordingly, the cumulative process economics $R(\$)$ are calculated defined as follows:

$$R = \sum_{k=0}^{T_f} \phi_k \tag{22}$$

where $T_f(time)$ is the final scenario length and ϕ_k are the instantaneous process economics at time k. Additionally, the time operating at constraint violating points is used as a measure to directly quantify constraint violation. This is defined as follows:

$$t_{viol} = \sum_{\forall k \in \{0,\ldots,T_f-1\} | g_k > 0} \Delta t_k \tag{23}$$

where Δt_k are the sampling interval lengths; accordingly, the cumulative time at constraint violation over a test scenario is quantified. Furthermore, constraint violations influence the quantity of material processed (i.e., throughput); especially in cases where below-specification product may be produced. The cumulative mass of material process m $(mass)$ is defined as follows:

$$m = \sum_{\forall k \in \{0,\ldots,T_f\} | g_k \leq 0} m_k \tag{24}$$

where m_k is the instantaneous mass throughput at time k and the expression in equation (24) sums over constraint-satisfying product. Lastly, the cost per mass rate P_{prod} is defined using equations (22) and (24) as follows:

$$P_{prod} = \frac{R}{m} \tag{25}$$

The production metrics in equations (22)–(25) are computed *a posteriori* to the scenarios tested for each case study.

4.1. Williams-Otto CSTR

The CSTR proposed by Williams and Otto, which is depicted in Figure 3, serves as a benchmark for MA. Its small size and nonlinearity make it a good example to examine price variation as a function of the operating conditions.

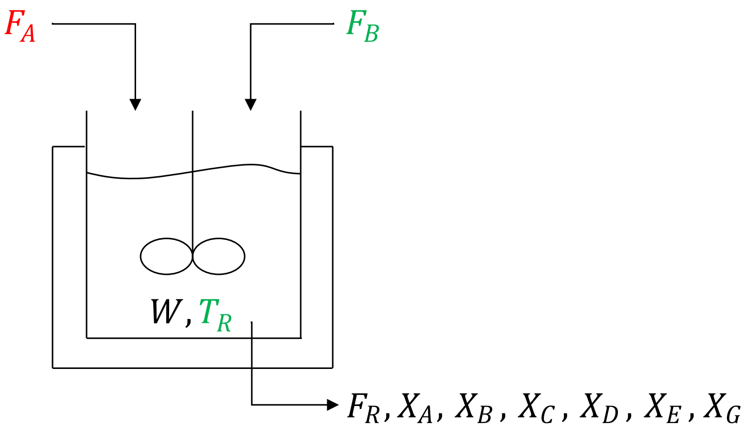


Figure 3: Williams-Otto CSTR. Disturbance variables shown in red and manipulated variables in green.

The system consists of chemical A/B as inputs, D/E as the desired products, and C/G as by-products; these undergo the three-reaction scheme:

$$A + B \overset{k_1}{\rightarrow} C: k_1 = (1.6599 \times 10^6) e^{\left(\frac{-6666.7}{T_R}\right)} \tag{26-1}$$

$$B + C \overset{k_2}{\rightarrow} D + E\text{: } k_2 = (7.2117 \times 10^8)e^{\left(\frac{-8333.3}{T_R}\right)} \tag{26-2}$$

$$C + D \overset{k_3}{\rightarrow} G\text{: } k_3 = (2.6745 \times 10^{12})e^{\left(\frac{-11111}{T_R}\right)} \tag{26-3}$$

where k_1, k_2, and $k_3(s^{-1})$ denote the reaction rate constant and $T_R(K)$ denotes the temperature of the well-mixed reaction. The three-reaction system corresponds to the following species dynamic material balances:

$$W\frac{dX_A}{dt} = F_A - F_R X_A - k_1 X_A X_B W \tag{26-4}$$

$$W\frac{dX_B}{dt} = F_B - F_R X_B - k_1 X_A X_B W - k_2 X_B X_C W \tag{26-5}$$

$$W\frac{dX_C}{dt} = -F_R X_C + 2k_1 X_A X_B W - 2k_2 X_B X_C W - k_3 X_C X_D W \tag{26-6}$$

$$W\frac{dX_D}{dt} = -F_R X_D + k_2 X_B X_C W - 0.5k_3 X_C X_D W \tag{26-7}$$

$$W\frac{dX_E}{dt} = -F_R X_E + k_2 X_B X_C W \tag{26-8}$$

$$W\frac{dX_G}{dt} = -F_R X_G + 1.5k_3 X_C X_D W \tag{26-9}$$

where $X_{\{A,B,C,D,E,G\}}(kg/kg)$ denote the molar fraction of each species. F_A and $F_B(kg/s)$ denote the inlet flowrates of A and B, respectively. $F_R(kg/s)$ denotes the outlet flowrate, which assumes an overall molar balance ($F_R = F_A + F_B$) with a constant holdup of $W = 2104.7\ kg$.

The model in equation (26) captures the complete species dynamics and represents the Williams-Otto plant. In addition to this plant model, a simplified steady-state model has also been formulated for the Williams-Otto CSTR. The abbreviated model omits species C and uses the follow two-reaction scheme to approximate the system:

$$A + 2B \overset{k_1}{\rightarrow} D + E\text{: } k_1 = (2.189 \times 10^8)e^{\left(\frac{-8077.6}{T_R}\right)} \tag{27-1}$$

$$A + B + D \overset{k_2}{\rightarrow} G\text{: } k_2 = (4.310 \times 10^{13})e^{\left(\frac{-12438}{T_R}\right)} \tag{27-2}$$

where all variables are defined as in equation (26) and the two-reaction scheme corresponds to the following steady-state material balances:

$$0 = F_A - F_R X_A - k_1 X_A {X_B}^2 W - k_2 X_A X_B X_D W \tag{27-3}$$

$$0 = F_B - F_R X_B - 2k_1 X_A {X_B}^2 W - k_2 X_A X_B X_D W \tag{27-4}$$

$$0 = -F_R X_D + k_1 X_A {X_B}^2 W - k_2 X_A X_B X_D W \tag{27-5}$$

$$0 = -F_R X_E + k_1 X_A {X_B}^2 W \tag{27-6}$$

$$0 = -F_R X_G - 3k_2 X_A X_B X_D W \tag{27-7}$$

The inputs to the system are $\boldsymbol{u} = [F_B \quad T_R]^T$ and the disturbance is $\boldsymbol{d} = [F_A]$. The nominal input and disturbance values for the system are $\boldsymbol{u_{nom}} = [6.1 \quad 366.15]^T$ and $\boldsymbol{d_{nom}} = [1.8]$. The inputs have the bounds $F_B \in [3,6]$ and $T_R \in$

$[343.15, 373.15]$ and the economic objective is to maximize the profit produced by the product species. This is denoted using the following minimization (note the negative to convert maximization to minimization) objective function:

$$-\phi = F_R(1143.38X_D + 25.92X_E) - 76.23F_A - 114.34F_B \qquad (28)$$

The mismatched model in equation (27) is deployed using the regular MA and dMAIS schemes shown in Figure 1 and Figure 2, respectively. Since only two inputs are available in this system, the dMAIS will only ever adapt with respect to one of them while the MA will adapt with respect to both. These competing schemes are evaluated on an economic basis using the cumulative profit function shown in equation (22) (i.e., $PM = R$). The disturbance variable is changed every period (ΔT) from the distribution $\boldsymbol{d} \sim [\mathcal{U}(0.3,3)]$**.** All necessary 0th order system information for MA is assumed to be measurable and sampled every $\Delta t = 3\ minutes = 1\ time\ interval$. The perturbation sizes for these inputs are $\boldsymbol{\delta} = 0.001\boldsymbol{u_t}$ (i.e., 0.1% of the current input value) and they are assumed to last $\tau = 50$ time intervals. Three test scenarios are performed, which feature forty disturbance realizations $\boldsymbol{d_l}\ \forall l \in \{1, \dots, B\}$, $B = 40$, each occurring every ΔT sampling intervals, such that the effect of the scheme can be analyzed over a long period of time and over a wide range of operating conditions.

Scenario 1 has $\Delta T = 250$. The filter matrices in equations (9)–(11) are assumed to all use an equivalent filter constant λ. This filter constant is varied for each simulation, which features a different random disturbance sequence for each filter run. This allows for the performance of the scheme to be assessed across a wide array of disturbance trajectories.

Scenario 2 sets $\Delta T = 250$ and varies the filter constant (λ). However, this scenario has the same disturbance sequence for all filters runs. This way the efficacy of the scheme with respect to λ can be extricated from the disturbance trajectory. This filter is important in the performance of the scheme as it affects the speed at which the modifiers are updated (and can thus inhibit the speed of dMAIS).

Scenario 3 has varying disturbance periods (ΔT), a filter constant of $\lambda = 0.01$, and the same disturbance sequence for each run. This extricates the effect of disturbance frequency on the scheme as it is designed to work best for increased frequencies.

Results from Scenario 1 are summarized in Table 1, where the dMAIS outperforms the standard MA scheme for small filter constants based on the total process revenue; this is reflected in larger process profits R since the W-O is a maximization problem. Aside from $\lambda = 0.01$, the benefit of the dMAIS scheme appears to be increasing as the modifiers are filtered less; this suggests that the filters indeed inhibit the speed at which the proposed scheme finds an economically preferable operating point. Furthermore, a break-even point between full MA and dMAIS occurs between $\lambda = 0.075$ and $\lambda = 0.1$ whereby full MA is best for higher filtering and dMAIS for lower. This likely occurs as increased filtering inhibits the ability of dMAIS to act quickly, thus eliminating its advantage over full MA. A conflating factor of this scenario is the random and varying disturbance sequence used for each filter run, which makes the improvement of the proposed scheme a function of the filter and the specific disturbance sequence; to extricate the former from the latter, scenario 2 keeps the same disturbance sequence for all filter runs.

Table 1: Results for all scenarios in the Williams-Otto case study. %I denotes the percent improvement (difference) in R of dMAIS with respect to MA.

Scenario 1	**Scenario 2**	**Scenario 3**

λ	R (\$) MA	R (\$) dMAIS	$\%I$	λ	R (\$) MA	R dMAIS	$\%I$	ΔT	R (\$) MA	R (\$) dMAIS	$\%I$
0.01	108,988	111,719	2.50	0.01	106,784	129,272	21.06	150	46,555	72,673	56.10
0.025	113,289	128,535	13.46	0.025	113,289	128,535	13.46	200	63,667	92,379	45.10
0.05	127,230	138,401	8.78	0.05	114,611	128,404	12.03	250	106,784	129,272	21.06
0.075	147,091	152,130	3.42	0.075	116,983	123,814	5.84	300	150,076	115,675	−23.26
0.1	198,767	190,434	−4.19	0.1	122,488	117,857	−3.78				

The results from Scenario 2 are shown in Table 1, whereby the trend in improvement of the dMAIS over the standard MA scheme is more clearly appreciable than in Scenario 1 owed to the equivalent disturbance sequence in all filter runs. This is also illustrated in the time domain by Figure 4, where the cost trajectories corresponding to the filter runs are displayed. As shown therein, the respective process revenues of dMAIS and MA diverge as time progresses. This is owed to the accretion of revenue over time and would continue further for longer scenarios. As the filter constant is increased, the revenue dynamics of the two implementations become increasingly similar whereby the dMAIS and standard MA show more overlap. As in Scenario 1, a break-even point occurs between $\lambda = 0.075$ and $\lambda = 0.1$ whereby the full MA becomes more favourable than the dMAIS as increased filtering inhibits convergence speed. This scenario illustrates the merit of allowing the dMAIS to adapt with respect to a single input. As exemplified by better performance for lower filter constants, Scenario 2 verifies the notion also observed in Scenario 1 that the advantage the dMAIS has over the standard MA is inhibited by aggressive filtering.

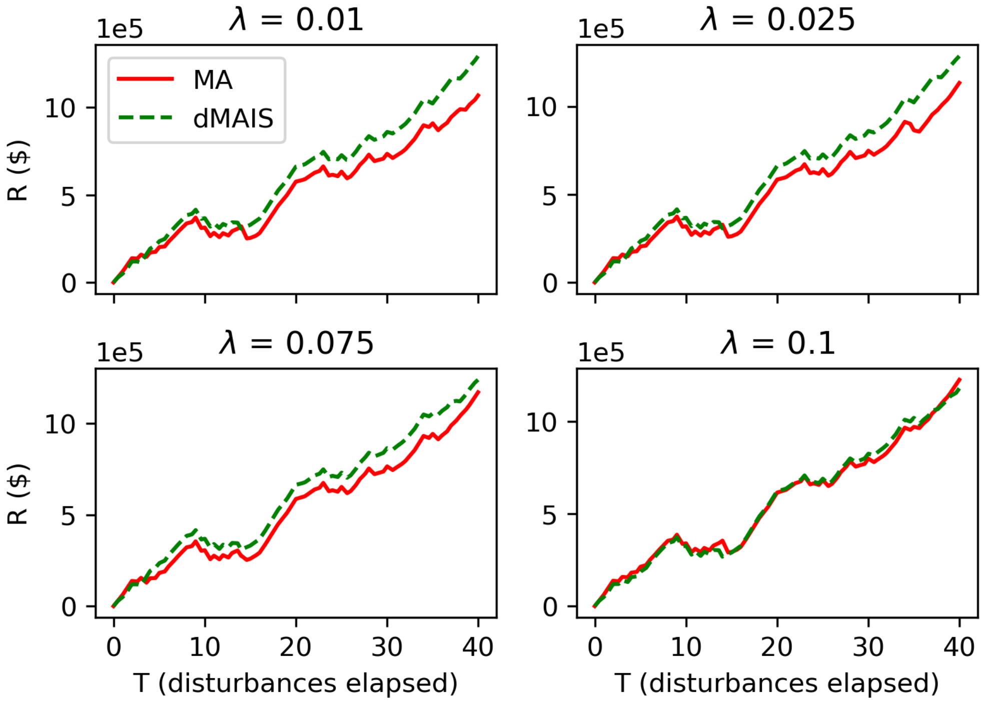


Figure 4: Profit accretion profiles for Williams-Otto case study, scenario 2, increasing filter constant.

The results from scenario 3 are also shown in Table 1, whereby a trend of increasing improvement of dMAIS over the standard MA is shown for runs with decreasing disturbance periods. This is also depicted transiently in Figure 5, whereby the revenue trajectories diverge increasingly as the disturbance period decreases (i.e., increased disturbance frequency). Note that the plots in Figure 5 are compressed/elongated to show the forty disturbance periods in the same range despite their varying period. As in the previous scenarios, a break-even point between the two schemes exists between $\Delta T = 250$ and $\Delta T = 300$. From this, we can conclude that the input-number regime for this system from equation (21) is as follows:

$$n_v = \begin{cases} 1 & \Delta T \leq 250 \\ n_u & \Delta T > 300 \end{cases} \tag{29}$$

Once the disturbance period becomes sufficiently large (i.e., infrequent disturbances), the dMAIS loses its competitive advantage of acting quickly as the standard MA has sufficient time to approach and benefit from economically superior operating points. Nevertheless, for short disturbance periods, the advantage can be significant (e.g., $\Delta T = 150$ with 56.1% cost improvement); this exemplifies the applicability of the scheme for constantly disturbed systems as proposed in the outset.

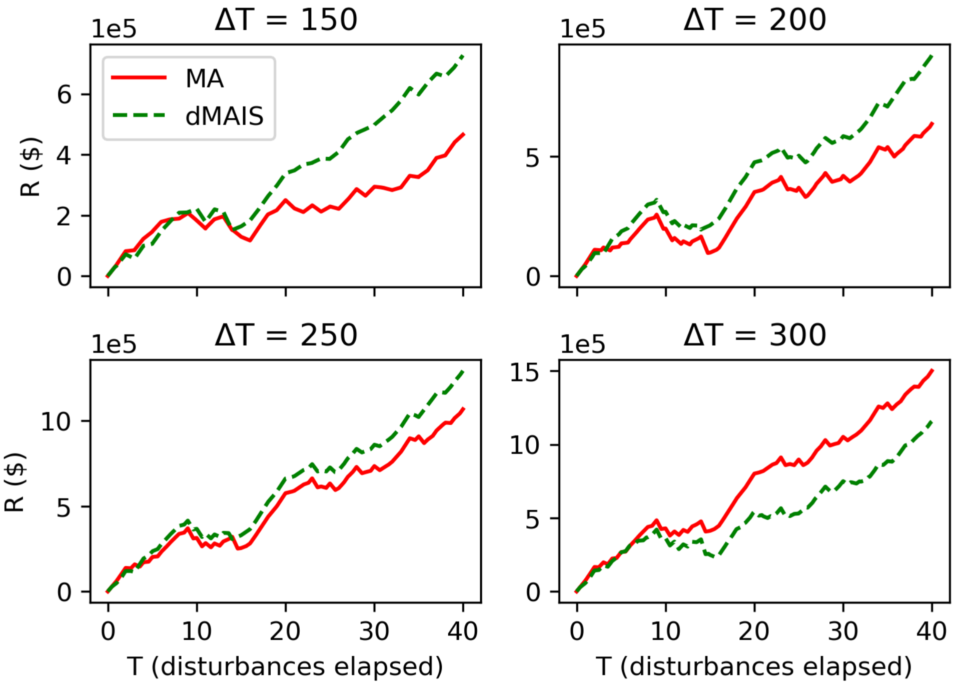


Figure 5: Profit accretion profiles for Williams-Otto case study, scenario 3, increasing disturbance periods

A couple conclusions can be made from the Williams-Otto case. Firstly, the filters are found to inhibit, or conversely incite, the dMAIS scheme to perform better than the MA scheme through quick action. As this is a tuning parameter, we conclude that dMAIS should use as little filtering as possible. Moreover, the disturbance period was found to affect the efficacy of the dMAIS scheme as quick adaptation is more suitable for quick disturbances.

While the Williams-Otto case-study explored herein is excellent as a benchmark as it has been used in multiple studies, it contains inherent features that leave some aspects of dMAIS unanswered. Firstly, it is a two-input system, which is the bare minimum requirement for dMAIS. While this number of inputs provides a simple way to assess the system, most industrial systems have more inputs. In this case study, only one of the two inputs is chosen for adaptation; in other systems, a subset (as opposed to only one) input can be chosen for this task. Moreover, the Williams-Otto optimization problem does not require any inequality constraints to be adapted; hence, the effect of the operating point adjustment step was not observed; these aspects will be addressed in the next case study.

4.2. Forced circulation evaporator

The forced circulation evaporator, depicted in Figure 6, is another nonlinear industrial system that has been used for multiple model-based control and optimization studies. This system provides a different perspective from the previous case study as its optima occur at an active constraint, making is a good setting in which to observe potential constraint violations.

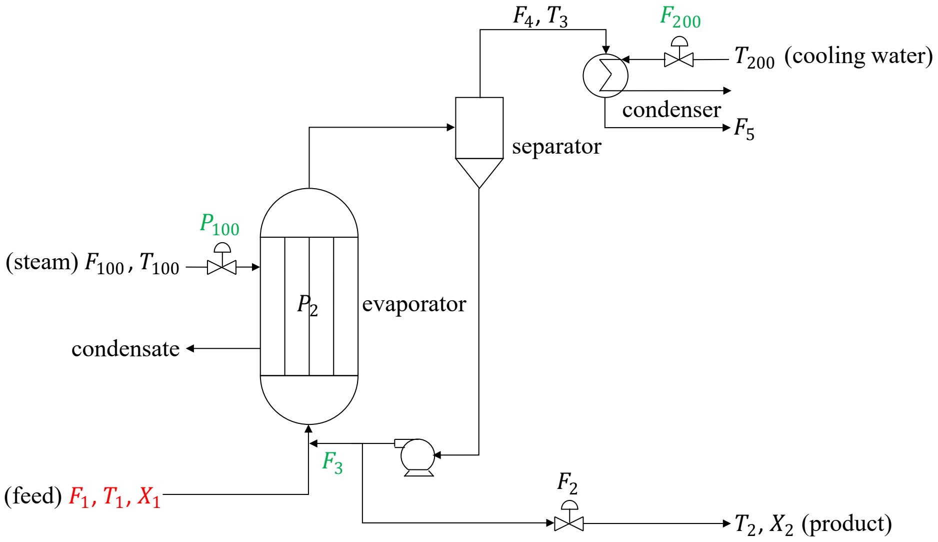


Figure 6: Forced circulation evaporator system. Disturbance variables shown in red and manipulated variables in green.

The dynamic evaporator system model consists of the following material balances:

$$H\frac{dX_2}{dt} = F_1X_1 - F_2X_2 \quad (30\text{-}1)$$

$$K\frac{dP_2}{dt} = F_4 - F_5 \quad (30\text{-}2)$$

$$F_2 = F_1 - F_4 \quad (30\text{-}3)$$

where $X_1(mol\%)$ and $X_2(mol\%)$ denote the feed and outlet product composition, respectively, and $P_2(kPa)$ is the evaporator pressure. F_1, F_2, F_4, and $F_5(kg/min)$ denote the feed, product, uncondensed by-product, and condensed by-product flowrates, respectively. $H = 20\,kg$ denotes the evaporator holdup and $K = 4\,kg/kPa$ denotes unit conversion constant. The energy balance over the evaporator is modelled as follows:

$$T_2 = 0.5616P_2 - 0.3126X_2 + 48.43 \quad (30\text{-}4)$$

$$T_3 = 0.507P_2 + 55 \quad (30\text{-}5)$$

$$F_4 = \frac{Q_{100} - F_1C_p(T_2 - T_1)}{\kappa} \quad (30\text{-}6)$$

where T_1, T_2, and $T_3(°C)$ denote the feed, product, and recycle temperatures, respectively. $Q_{100}(kW)$ denotes the evaporator heat duty, $C_p = 0.07(kW \cdot min)/(kg \cdot °C)$ denotes a constant evaporator fluid heat capacity, and $\kappa = 38.5\ (kW \cdot min)/kg$ denotes an evaporator fluid latent heat of evaporation. The energy balance over the steam jacket is modelled as follows:

$$T_{100} = 0.1538P_{100} + 90 \tag{30-7}$$

$$Q_{100} = UA_1(T_{100} - T_2) \tag{30-8}$$

$$UA_1 = 0.16(F_1 + F_3) \tag{30-9}$$

$$F_{100} = \frac{Q_{100}}{\kappa_s} \tag{30-10}$$

where $T_{100}(°C)$, $P_{100}(kPa)$, and $F_{100}(kg/min)$ denote the jacket inlet saturated steam temperature, pressure, and flowrate. $UA_1(kW/°C)$ denotes the jacket-to-evaporator heat transfer coefficient and $\kappa_s = 36.6(kW \cdot min)/kg$ denotes the latent heat of saturated steam. The condenser is modelled as follows:

$$Q_{200} = \frac{UA_2(T_3 - T_{200})}{1 + \frac{UA_2}{2C_pF_{200}}} \tag{30-11}$$

$$F_5 = \frac{Q_{200}}{\lambda} \tag{30-12}$$

where $Q_{200}(kW)$, $T_{200}(°C)$, and $F_{200}(kg/min)$ denote the condenser duty, temperature, and cooling water flowrate, respectively. $UA_2 = 6.84(kW/°C)$ denotes the condenser heat transfer coefficient.

Equation (30) represents the mechanistic (i.e., "perfect") evaporator model. For the purposes of this study, equation and parameter values were changed to introduce a plant-model mismatch. Accordingly, the mismatched model uses $\kappa = 35.5\ (kW \cdot min)/kg$ in equations (30-6) and (30-12), $\kappa_s = 34.6(kW \cdot min)/kg$ in equation (30-10), and substitutes equation (30-9) for the following:

$$UA_1 = 0.16F_3 \tag{31}$$

Notably, the product composition is subjected to the following constraint to ensure a sufficiently high-quality product is generated by the evaporator:

$$X_2 \geq 25\% \tag{32}$$

The disturbance and manipulated variables for the forced circulation evaporator system are $\boldsymbol{d} = [X_1 \quad F_1 \quad T_1 \quad T_{200}]^T$ and $\boldsymbol{u} = [P_{100} \quad F_{200} \quad F_3]^T$, respectively. The nominal disturbance and input values are $\boldsymbol{d_{nom}} = [5 \quad 10 \quad 40 \quad 25]^T$ and $\boldsymbol{u_{nom}} = [200 \quad 200 \quad 50]^T$. The inputs have the bounds $P_{100} \in [10{,}400]$, $F_{200} \in [10, 400]$, and $F_3 \in [1{,}100]$. The objective of this system is to minimize the cost expressed as follows:

$$\Phi = 0.1009(F_2 + F_3) + 60F_{200} + 60P_{100} \tag{33}$$

The disturbance variables are changed every period (ΔT) from individual uniform distributions that serve as multipliers for the nominal disturbance values i.e., $\boldsymbol{d} \sim [\mathcal{U}(0.8{,}1.2) \cdot 5 \quad \mathcal{U}(0.8{,}1.2) \cdot 10 \quad \mathcal{U}(0.8{,}1.2) \cdot 40 \quad \mathcal{U}(0.8{,}1.2) \cdot 25]^T$. All necessary 0th order system information for MA is assumed to be measurable and sampled every $\Delta t = 1\ minute = 1\ time\ interval$; moreover, a varying number of modifier directions are used for different scenarios in the case study.

The perturbation sizes for these inputs are $\boldsymbol{\delta} = 0.001\boldsymbol{u_t}$ (i.e., 0.1% of the current input value) and assumed to last $\tau = 300$ time intervals. Moreover, the system is limited to only ten adjustment iterations $j_{max} = 10$ of $\boldsymbol{\alpha} = 0.01\boldsymbol{v_{t-1}}$ such that the next operating point update is not delayed significantly.

A few scenarios were performed for this case study, which features ten disturbances $\boldsymbol{d_l}\ \forall l \in \{1, \dots, B\}$, $B = 10$, each occurring every ΔT sampling intervals. The disturbance period (ΔT), number of inputs modified with respect to (n_v), and scheme (i.e., full MA, standard MA, dMAIS) are varied such that the timing, degree, and type of modification can be analyzed. In addition to the full MA and dMAIS, a version of the dMAIS without the operating point adjustment step (blue block of Figure 2) is also deployed and denoted dMAIS(-); this scheme is impractical in practice but serves to observe the effect of the adjustment step for active constraints proposed in this scheme. A number after dMAIS denotes the number of modifiers being continually refined; for instance, dMAIS1(-) denotes that a single input is being modified with respect to and that the constraint adjustment scheme is not being deployed.

The data for this scenario is shown in the appendix Table A1. Figure 7 shows the cumulative cost calculated using equation (22) of the competing scheme under various disturbance periods. It should be noted first that the full MA is unable to ever perform a single iteration in the $\Delta T = 2000$ case; this occurs as performing n_u perturbations is too protracted and a new disturbance always occurs before they finish. Moreover, longer disturbance periods entail longer simulation times, thus increasing values of R with increasing ΔT, as shown in Figure 7. Nevertheless, as observed therein, the full MA outperforms the dMAIS and dMAIS(-) schemes on a cumulative cost minimization basis for all disturbance periods where it can perform an iteration. On this cumulative cost basis, there seems to be relatively little difference between dMAIS and dMAIS(-) as indicated by their nearly equivalent trajectories. However, this superficial interpretation does not consider constraint violations, thus the economic analysis should be adjusted to consider process throughput.

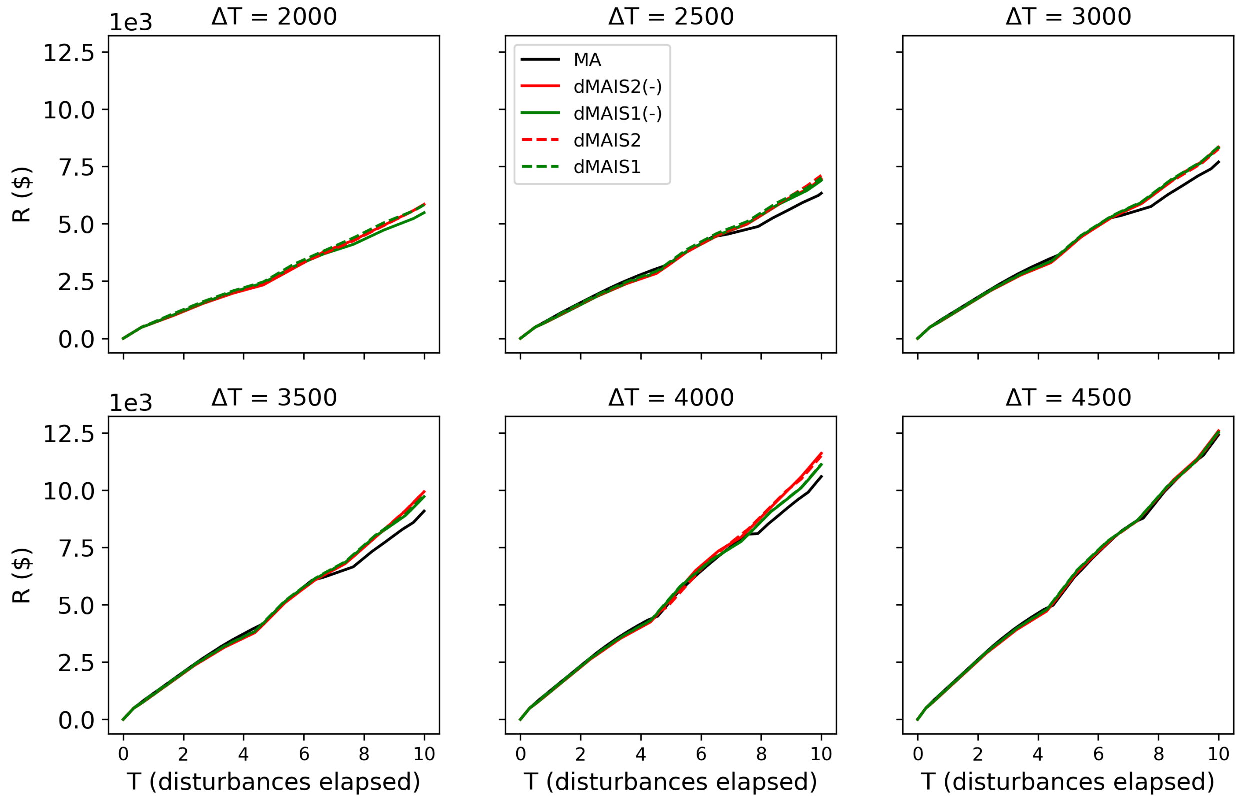


Figure 7: Cost accretion profiles for evaporator case study, increasing disturbance periods.

As outlined in the introduction, basic MA schemes do not guarantee constraint satisfaction during modifier refinement; moreover, as this study uses frequent periodic disturbances, the constraint satisfaction upon convergence property is not observed since convergence is not reached. Table 2 summarizes the cumulative time at constrain violation as defined in equation (23) while Figure 9 displays the throughput for all competing schemes as defined in equation (24). Additionally, the constrain trajectories for all scenarios can be found in Figure A1 (Appendix). In terms of times at constrain violation and throughput, the analysis favours the proposed scheme until the disturbance period is increased to $\Delta T = 4500$. Regardless of the disturbance period, the dMAIS schemes are shown to always outperform their dMAIS(-) counterparts on a constraint violation and mass processed basis. This is owed to the constraint adjustment step, which ensures that the pre-perturbation operating point abides with the product purity requirement in equation (32); thus, the dMAIS iterations produce above-specification product while their dMAIS(-) equivalents may not. This effect of the adjustment step is further evident when comparing dMAIS to full MA whereby the former also outperforms the latter on constraint violation and throughput bases. As illustrated in Table 2, the cumulative time at constraint violation as defined in equation (23) is highest for the full MA scheme for all scenario except where the disturbances are sufficiently spaced at $\Delta T = 4000$. With more frequent disturbances, even the dMAIS(-) without adjustments outperforms the full MA on constraint satisfaction; thus, the quick action given by the dMAIS alone is observed to have the effect of staying in constraint violating points for less time.

Table 2: Cumulative time (min) at constraint violation t_{viol} for evaporator case study, increasing disturbance periods

	$\Delta T = 2000$	$\Delta T = 2500$	$\Delta T = 3000$	$\Delta T = 3500$	$\Delta T = 4000$	$\Delta T = 4500$
MA	–	240	302	328	310	281
dMAIS2	148	176	233	237	249	286
dMAIS2(-)	165	207	251	255	255	320
dMAIS1	119	146	205	255	3222	349
dMAIS1(-)	167	209	254	303	364	411

Analyzing a subset of constraint trajectories from Figure A1 (Appendix) more closely, Figure 8 displays the results for the best (dMAIS1) scheme, its counterpart without the adjustment set (dMAIS1(-)), and the full MA scheme for $\Delta T = 2500$. In these trajectories, the result of the constraint adjustment step is more clearly appreciable. In several time instances (e.g., $T \sim 0.7, 4.6, 7.6$), the adjustment step is activated to bring the composition above the quality constraint. The effects of the adjustment step with respect to constraint satisfaction are accrued over time, thus generating the operational differences between dMAIS and dMAIS(-) schemes observed in in Table 2; these will continue to accrue as the process operation evolves in time.

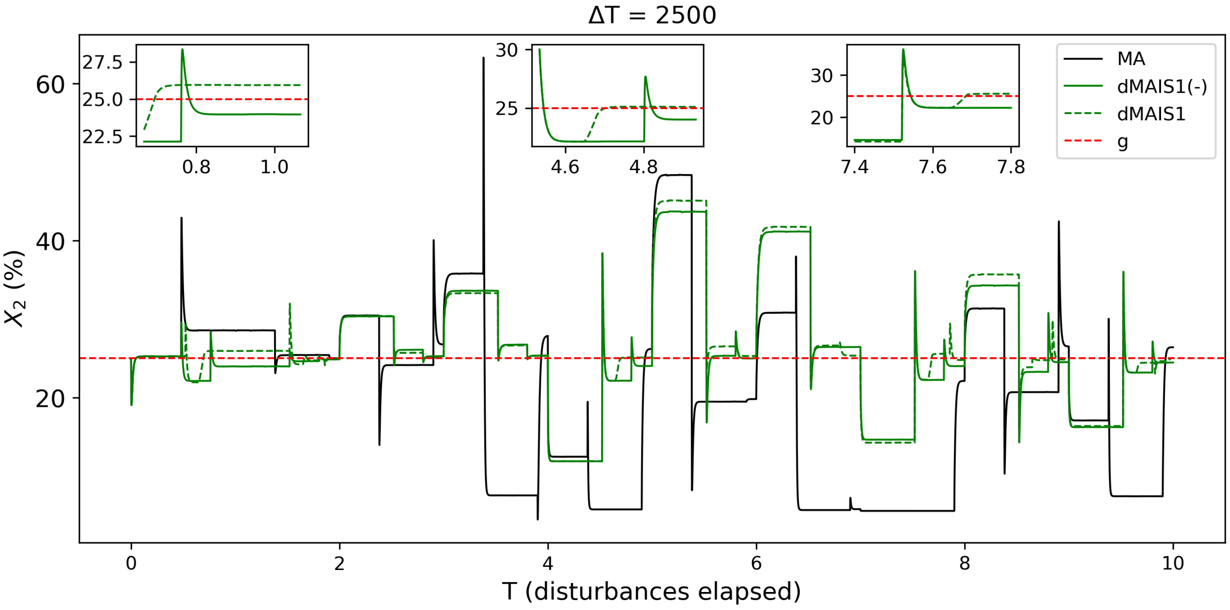


Figure 8: Product quality trajectory for $\Delta T = 2500$ scenario in evaporator case study

Figure 9 illustrates how the number of inputs modified with respect to (n_v) can impact the efficacy of the dMAIS scheme. For short disturbance periods (i.e., $\Delta T = 2000$, $\Delta T = 2500$, and $\Delta T = 3000$), the dMAIS1 (i.e., $n_v = 1$) scheme are best. This occurs as the disturbance happen frequently enough to require more iterations of the dMAIS that are facilitated by the dMAIS1 schemes. For intermediate disturbance periods (i.e., $\Delta T = 3500$ and $\Delta T = 4000$), the dMAIS2 (i.e., $n_v = 2$) scheme is the best. In this case, the disturbances happen less frequently as to allow for more iterations of the dMAIS schemes performed in dMAIS2; however, they still occur frequently enough as to not favour the full MA scheme. Furthermore, for long disturbance periods (i.e., $\Delta T = 4500$), the full MA scheme dominates as enough time between disturbances occurs for the full MA to arrive near the plant optima.

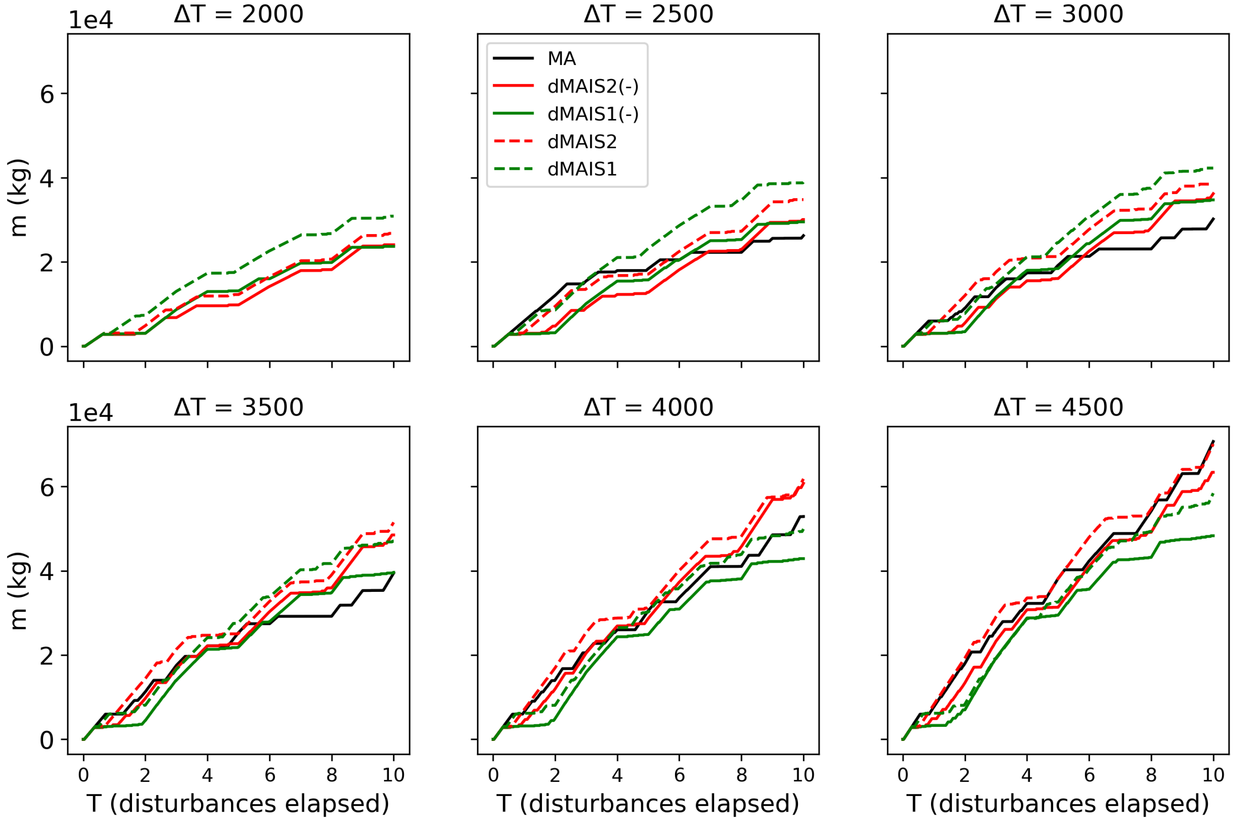


Figure 9: Material production profiles for evaporator case study, increasing disturbance periods

Figure 10 summarizes the aggregate effect of cost (Figure 7) and throughput (Figure 9) as defined in equation (25) (i.e., $PM = P_{prod}$). As with the throughput, a clear pattern emerges whereby the dMAIS schemes are superior to the dMAIS(-) schemes, which are superior to the full MA scheme. Thus, for the evaporator case, the following regimes are established for the number of inputs modified with respect to:

$$n_v = \begin{cases} 1 & \Delta T \leq 3000 \\ 2 & 3500 < \Delta T \leq 4000 \\ n_u & \Delta T > 4000 \end{cases} \tag{34}$$

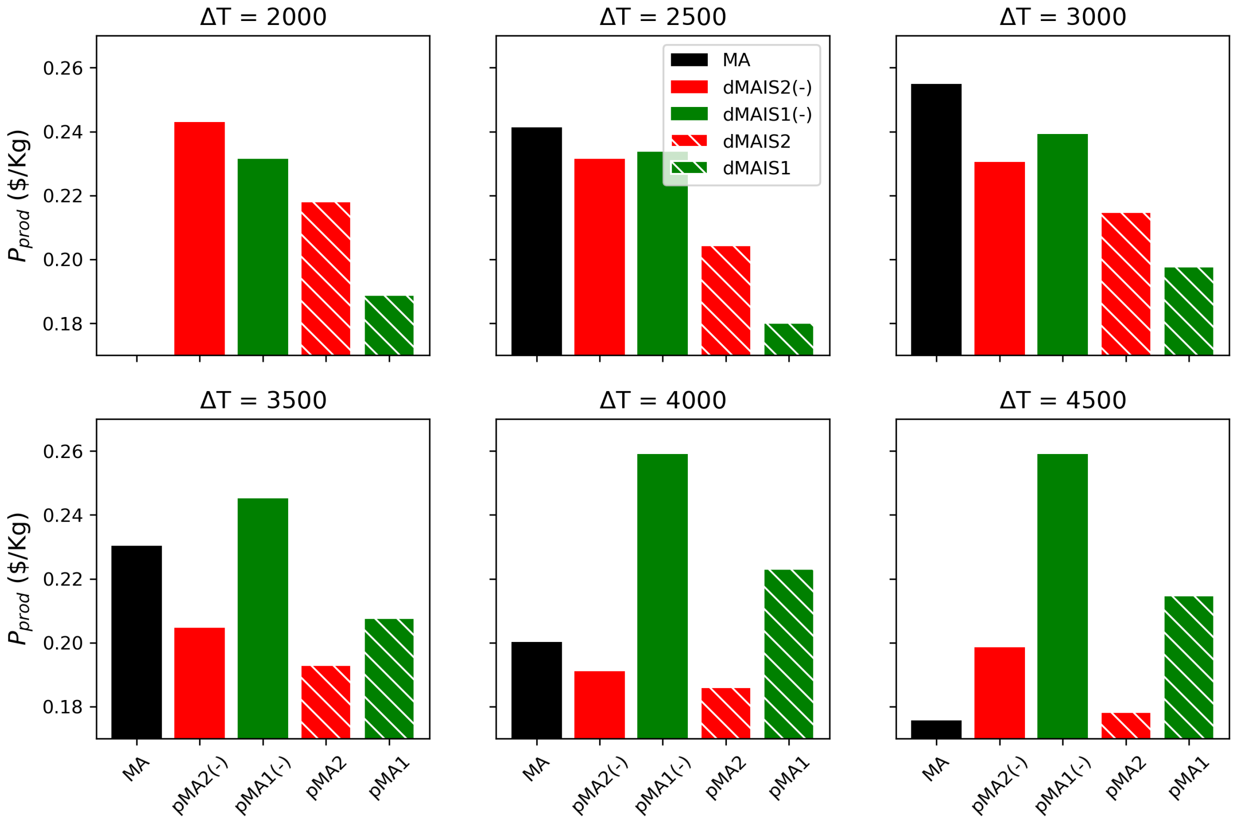


Figure 10: Cost per unit weight for evaporator case study, increasing disturbance periods

In contrast to the William-Otto scenario, the evaporator exemplifies the potential economic importance of producing constraint-adhering product as affected by increased production. Moreover, we observe the effect of different number of input modifications more concretely and its relationship to the disturbance period.

While dMAIS was shown to outperform the traditional MA scheme in the case of frequent periodic disturbances, many MA variants could have similar benefits to dMAIS under these conditions; one such variant is dMA (Costello et al., 2016). As noted previously, dMA can choose multi-input directions for adaptation but requires the one-time local computation of a parameter/input sensitivity matrix. While dMAIS can only use single-input adaptation directions, sensitivities are computed locally; thus, a potential tradeoff is present between the schemes, making them apt for comparison.

After determining the suitable number of modifier directions (i.e., $n_v = 1$) using algorithm 2 for a disturbance period of ΔT =2000, several dMA scenarios were performed for the same ten-disturbance sequence was imposed on the dMAIS. The results from the dMA approach can be found in Table 3 whereby each scenario differs in the point around which the model sensitivity matrix is identified (as noted in Costello et al., 2016; this matrix is required for determining the privileged dMA directions $\boldsymbol{U_t}$.

Table 3: Performance of dMA scenarios with sensitivity matrix calculated at different operating point, $n_v = 1$, and $\Delta T = 2000$.

Scenario	**Sensitivity matrix point**	$\boldsymbol{R}$ ($)	$\boldsymbol{m(kg)}$	$\boldsymbol{P_{prod}}$ ($\boldsymbol{\$/kg}$)
1	$\boldsymbol{d} = [6.0 \quad 9.8 \quad 37.9 \quad 23.2]^T$ $\boldsymbol{u} = [350 \quad 200 \quad 40]^T$	6926.08	38404.97	0.180

2	$\boldsymbol{d} = [4.1 \quad 12.7 \quad 37.1 \quad 23.\,5]^T$ $\boldsymbol{u} = [350 \quad 200 \quad 50]^T$	6314.30	24860.15	0.254
3	$\boldsymbol{d} = [5.1 \quad 9.7 \quad 51.0 \quad 20.9]^T$ $\boldsymbol{u} = [166 \quad 183 \quad 98]^T$	7004.76	38474.18	0.182
4	$\boldsymbol{d} = [5.1 \quad 12.9 \quad 37.0 \quad 23.2]^T$ $\boldsymbol{u} = [390 \quad 200 \quad 50]^T$	6724.96	33030.24	0.204
5	$\boldsymbol{d} = [5.2 \quad 8.8 \quad 49.8 \quad 24]^T$ $\boldsymbol{u} = [148 \quad 133 \quad 95]^T$	7024.16	38483.14	0.182
6	$\boldsymbol{d} = [5.6 \quad 8.4 \quad 43.3 \quad 24.5]^T$ $\boldsymbol{u} = [335 \quad 80 \quad 32]^T$	6418.26	28182.64	0.228
7	$\boldsymbol{d} = [4.1 \quad 9.6 \quad 46.7 \quad 22.8]^T$ $\boldsymbol{u} = [324 \quad 150 \quad 54]^T$	6928.07	36818.15	0.188
8	$\boldsymbol{d} = [5.8 \quad 8.8 \quad 54.0 \quad 28.3]^T$ $\boldsymbol{u} = [218 \quad 126 \quad 41]^T$	6477.56	30165.92	0.21
dMA average	–	6727.27	33552.42	0.204
dMAIS	–	5827.37	30852.88	0.189

As shown in Table 3, and Figure A2 in the appendix, the performance of the dMA scheme on the evaporator is highly dependent on which point the sensitivity matrix is computed through the direction it chooses. Compared to the dMAIS, which is not dependent on this matrix, the dMA can perform moderately better (e.g., scenario 1; ~4.8% improvement) or significantly worse (e.g., scenario 2; ~34.4% deterioration). Note that dMAIS does not rely on a parameter distribution being known *a priori*, thus this assumption is alleviated by the proposed approach. Moreover, the potential variability in performance owed to the sensitivity matrix point is abated by using dMAIS. On aggregate, dMAIS outperforms dMA (~7.4% improvement in using dMAIS over the average in dMA), as shown in Table 3. Conversely, the dMA is shown to be able to outperform dMAIS if the sensitivity matrix is computed at an adequate point; thus, there is a tradeoff in the two schemes between average and variability in performance. We note that superiority of dMAIS over dMA is not necessarily generalizable and may be dependent on the case study under consideration. As discussed above, this is mostly owed to the use of multi-input directions compared to the ability to update directions online.

5. Conclusion and future works

MA is a commonly used method to abate model uncertainty, but its gradient estimation step can cause delays in the update of operating points. Accordingly, the dMAIS scheme presented herein only modifies with respect to a subset of the inputs chosen using the modified cost metric. This subset is refined as the dMAIS scheme progresses such that the modifications are chosen to have the largest effect on the process economics. Additionally, dMAIS employs an operating point adjustment step, which drives constraint-violating systems into constraint adhering regions prior to the perturbation step. Using the dMAIS scheme, constraint satisfaction is improved as the modifiers are being refined. The proposed scheme was deployed on the Williams-Otto plant where it was found to be superior to the full MA for small disturbance periods and small filter constants; thus, leveraging modifier refinement speed to its economic advantage. Moreover, the dMAIS scheme was deployed for an evaporator case study with active constraints whereby it was shown to increase material throughput through decreased constraint violation compared to the full MA. Increased throughput was shown to also result in improved process economics. The evaporator case also exemplified how the best number of modifiers is dependent on the disturbance period such that different numbers of modifiers can be used in different disturbance regimes. With respect to standard dMA, dMAIS was found to lead to an average performance improvement in the evaporator owed to its lack of dependence on the initial sensitivity matrix. Conversely, some dMA scenarios were found to outperform dMAIS if the initial sensitivity matrix was computed around certain operating points; thus, there is a tradeoff in robustness and performance between the two methods.

The dMAIS scheme proposed in this work have been implemented using the traditional perturbation method; however, gradients acquisition can be made more efficient through dual MA. Future works will also investigate the joint use of dMAIS and dual MA, which could lead to further benefits in speeding up modifier refinement. The dMAIS scheme can be inhibited by filtering, this limits its applicability to low noise environments; thus, an alternative noise abatement scheme must be proposed to make the scheme suited to noisy measurements (e.g., Patrón and Ricardez-Sandoval, 2022b). Furthermore, the scheme proposed herein was only tested in systems whereby preliminary runs may be performed for tuning; this may not be achievable or desirable in all systems. Accordingly, online tuning and tuning budget sizing for dMAIS requires further attention. Lastly, the case studies presented in this work were selected such that they provide clear illustrations of the benefits and limitations of the proposed scheme. However, industrial plants are usually more complex; future works will thus deploy dMAIS in high-dimensional constrained chemical plants (i.e., those involving many inputs and constraints).

Acknowledgements

The authors would like to acknowledge the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial support.

Nomenclature

dMAIS /MA schemes		**Subscripts and superscripts**	
Percent improvement	$\%\boldsymbol{I}$	Active constraints	act
Matrix of ones	$\boldsymbol{1}$	Filtered quantity	f
Maximum adjustment iterations	a_{max}	Constraint quantity	g
Active constraint indicator matrix	$\boldsymbol{A}$	Index for constraints	i
Computational budget	B	Inactive constraints	in
Disturbances	$\boldsymbol{d}$	Index for inputs	j
Process model	$\boldsymbol{f}$	Index for time	k
Plant	$\boldsymbol{f_p}$	Index for disturbances elapsed	l
Constraints	$\boldsymbol{g}$	Lower bound	lb
Identity matrix	$\boldsymbol{I}$	Model quantity	m
Jacobian matrix	$\boldsymbol{J}$	Modified formulation	MA
Mass of material processed	m	Nominal quantity	nom
Number of	n	Ordered vector	ord
Number of MA iterations to convergence	n_{MA}	Process (sampled) quantity	p
Number of dMAIS iterations to convergence	n_{dMAIS}	Perturbed quantity	$pert$
Process economics per unit mass	P_{prod}	Directionally modified formulation with input selection	$dMAIS$
Constraint violation matrix	$\boldsymbol{Q}$	Current time	t
Total process economics	R	Upper bound	ub
Cumulative time violating constraints	t_{viol}	Objective function quantity	ϕ
Input vector	$\boldsymbol{u}$	Upon convergence	∞
Ordered set of inputs	U		
dMA direction matrix	$\boldsymbol{U}$	**Williams-Otto plant**	
dMAIS modified input vector	$\boldsymbol{v}$	Mass flowrate of streams $\{A,B,R\}$	$F_{\{A,B,R\}}$
Set of inputs used for modification	V	Rate of reactions $\{1,2,3\}$	$k_{\{1,2,3\}}$
dMAIS direction matrix	$\boldsymbol{V}$	Mass holdup	W
dMAIS inputs not used for modification	$\boldsymbol{w}$	Mass fraction of species $\{A,B,C,D,E,G\}$	$X_{\{A,B,C,D,E,G\}}$
Set of inputs not used for modification	W		
States	$\boldsymbol{x}$	**Evaporator plant**	
Adjustment bounds	$\boldsymbol{\alpha}$	Fluid heat capacity	C_p
Perturbation size	δ	Molar flowrate of streams $\{1,2,3,4,5,100,200\}$	$F_{\{1,2,3,4,5,100,200\}}$
Sampling interval	Δt	Pressure of streams $\{2,100\}$	$F_{\{2,100\}}$
Disturbance period	ΔT	Evaporator heat duty	Q_{100}
0th order modifiers	$\boldsymbol{\varepsilon}$	Evaporator holdup	H
Boundary for number of updated modifiers	$\boldsymbol{\zeta}$	Unit conversion constant	K
Filter matrices	$\boldsymbol{\lambda}$	Condenser heat duty	Q_{200}
1st order modifiers	$\boldsymbol{\mu}$	Temperature of streams $\{1,2,3,100,200\}$	$T_{\{1,2,3,100,200\}}$
1st order modifiers refined in dMAIS	$\boldsymbol{\gamma}$	Jacket-evaporator heat transfer coefficient	UA_1
Perturbation time	τ	Condenser heat transfer coefficient	UA_2
Operating point change transient time	T	Mole percent of species $\{1,2\}$	$X_{\{1,2\}}$
Economic objective function	ϕ	Fluid latent heat of evaporation	κ
1st order modifiers not refined in dMAIS	$\boldsymbol{\omega}$	Steam latent heat of evaporation	κ_s

References

Bottari, A., Zumoffen, D.A.R., Marchetti, A.G., 2020. Economic Control Structure Selection for Two-Layered Real-Time Optimization Systems. *Ind. Eng. Chem. Res.* 59(49), 21413–21428. https://doi.org/10.1021/acs.iecr.0c02591.

Bunin, G.A., François, G., Srinivasan, B., Bonvin, D., 2011. Input Filter Design for Feasibility in Constraint-Adaptation Schemes. *IFAC. Proc. Vol*. 44(1), 5585–5590. https://doi.org/10.3182/20110828-6-IT-1002.02937.

Chachuat, B., Srinivasan, B., Bonvin, D., 2009. Adaptation strategies for real-time optimization. *Comput. Chem. Eng.* 33(10), 1557–1567. https://doi.org/10.1016/j.compchemeng.2009.04.014.

Cao, S., Rhinehart, R.R., 1995. An efficient method for on-line identification of steady state. *J. Process Control* 5(6), 363–374. https://doi.org/10.1016/0959-1524(95)00009-F.

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. https://doi.org/10.1021/ie00069a034.

Costello, S., François, G., Bonvin, D., 2016. A Directional Modifier-Adaptation Algorithm for Real-Time Optimization. *J. Process Control* 39, 64–76. https://doi.org/10.1016/j.jprocont.2015.11.008.

Darby, M.L., Nikolaou, M., Jones, J., Nicholson, D., 2011. RTO: An overview and assessment of current practice. *J. Process Control* 21(6), 874–884. https://doi.org/10.1016/j.jprocont.2011.03.009.

de Avila Ferreira, T., François, G., Marchetti, Bonvin, D., 2017. Use of Transient Measurements for Static Real-Time Optimization. *IFAC-PapersOnLine*, 50(1), 5737–5742. https://doi.org/10.1016/j.ifacol.2017.08.1130

Forbes, J.F., Marlin, T.E., 1994. Model Accuracy for Economic Optimizing Controllers: The Bias Update Case. *Comput. Ind. Eng. Chem. Res.* 33(8), 1919–1929. https://doi.org/10.1021/ie00032a006

Forbes, J.F., Marlin, T.E., 1996. Design cost: a systematic approach to technology selection for model-based real-time optimization systems. *Comput. Chem. Eng.*, 20(6–7), 717–734. https://doi.org/10.1016/0098-1354(95)00205-7.

François, G., Bonvin, D., 2014. Use of Transient Measurements for the Optimization of Steady-State Performance via Modifier Adaptation. *Ind. Eng. Chem. Res.*, 53(13), 5148–5159. https://doi.org/10.1021/ie401392s.

Gao, W., Engell, S., 2005. Iterative set-point optimization of batch chromatography. *Comput. Chem. Eng.* 29(6), 1401–1409. https://doi.org/10.1016/j.compchemeng.2005.02.035.

Gao, W., Wenzel, S., Engell, S., 2016. A reliable modifier-adaptation strategy for real-time optimization. *Comput. Chem. Eng.* 91, 318–328. https://doi.org/10.1016/j.compchemeng.2016.03.019.

Hart, W., Watson, J., Woodruff, D., 2011. Pyomo: modeling and solving mathematical programs in Python. *Math. Program Comput.* 3(3), 219–260. https://doi.org/10.1007/s12532-011-0026-8.

Hou, Y., Wu, J., Chen, Y., 2016. Online Steady State Detection Based on Rao-Blackwellized Sequential Monte Carlo. *Qual. Reliab. Eng. Int.,* 32(8), 2667–2683. https://doi.org/10.1002/qre.2067.

HSL. A collection of Fortran codes for large scale scientific computation. http://www.hsl.rl.ac.uk.

Jiang, T., Chen, B., He, X., Stuart, P., 2003. Application of steady-state detection method based on wavelet transform. *Comput. Chem. Eng.*, 27(4), 569–578. https://doi.org/10.1016/S0098-1354(02)00235-1.

Kornaros, M., Dokianakis, S.N., Lyberatos, G., 2010. Partial Nitrification/Denitrification Can Be Attributed to the Slow Response of Nitrite Oxidizing Bacteria to Periodic Anoxic Disturbances. *Environ. Sci. Technol.*, 44(19), 7245–7253. https://doi.org/10.1021/es100564j.

Lee, P.L., Newell, R.B., Sullivan, G.R., 1989. Generic Model Control – A Case Study. *Can. J. Chem. Eng*. 67, 478–484. https://doi.org/10.1002/cjce.5450670320.

Mansour, M., Ellis, J.E., 2003. Comparison of methods for estimating real process derivatives in on-line optimization. *Appl. Math. Model.* 27(4), 275–291. https://doi.org/10.1016/S0307-904X(02)00124-5.

Marchetti, A.G., 2022. Feasibility in real-time optimization under model uncertainty. The use of Lipschitz bounds. *Comput. Chem. Eng.* 168, 108057. https://doi.org/10.1016/j.compchemeng.2022.108057.

Marchetti, A., Chachuat, B., Bonvin, D., 2009. Modifier-Adaptation Methodology for Real-Time Optimization. *Ind. Eng. Chem. Res.* 48(13), 6022–6033. https://doi.org/10.1021/ie801352x.

Marchetti, A., Chachuat, B., Bonvin, D., 2010. A dual modifier-adaptation approach for real-time optimization. *J. Process Control* 20(9), 1027–1037. https://doi.org/10.1016/j.jprocont.2010.06.006.

Marchetti, A.G., de Avila Ferreira, T., Costello, S., Bonvin, D, 2020. Modifier Adaptation as a Feedback Control Scheme. *Ind. Eng. Chem. Res.*, 59(6), 2261–2274. https://doi.org/10.1021/acs.iecr.9b04501.

Marchetti, A.G., François, G., Faulwasser, T., Bonvin, D., 2016. Modifier Adaptation for Real-Time Optimization–Methods and Applications. *Processes* 4(4), 55. https://doi.org/10.3390/pr4040055.

Marchetti, A.G., Faulwasser, T., Bonvin, D., 2017a. A feasible-side globally convergent modifier-adaptation scheme. *J. Process Control* 54, 38–46. https://doi.org/10.1016/j.jprocont.2017.02.013.

Marchetti, A.G., Singhal, M., Faulwasser, T., Bonvin, D., 2017b. Modifier adaptation with guaranteed feasibility in the presence of gradient uncertainty. *Comput. Chem. Eng.* 98, 61–69. https://doi.org/10.1016/j.compchemeng.2016.11.027.

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. https://doi.org/10.1021/acs.iecr.5b01946.

del Rio Chanona, E.A., Petsagkourakis, P., Bradford, E., Alves Graciano, J.E., Chachuat, B., 2021. Real-time optimization meets Bayesian optimization and derivative-free optimization: A tale of modifier adaptation. *Comput. Chem. Eng.*, 147, 107249. https://doi.org/10.1016/j.compchemeng.2021.107249.

Pan, Y., Lee, J.H., 2003. Identification and Control of Processes with Periodic Operations or Disturbances. *Ind. Eng. Chem. Res.* 42(9), 1938–1947. https://doi.org/10.1021/ie020313y.

Patrón, G.D., Ricardez-Sandoval, L., 2022a. An integrated real-time optimization, control, and estimation scheme for post-combustion CO_2 capture. *Appl. Energy*, 308, 118302. https://doi.org/10.1016/j.apenergy.2021.118302.

Patrón, G.D., Ricardez-Sandoval, L., 2022b. Low Variance Parameter Estimation Approach for Real-Time Optimization of Noisy Process Systems. *Ind. Eng. Chem. Res.*, 61(45), 16780–16798. https://doi.org/10.1021/acs.iecr.2c02897.

Pawlowski, A., Guzmán, J.L., Rodríguez, F., Berenguel, M., Normey-Rico, J.E., 2011. Predictive Control with Disturbance Forecasting for Greenhouse Diurnal Temperature Control. *IFAC. Proc. Vol*. 44(1), 1779–1784. https://doi.org/10.3182/20110828-6-IT-1002.00857.

Rodrigues, D., Marchetti, A.G., Bonvin, D., 2022. On improving the efficiency of modifier adaptation via directional information. *Comput. Chem. Eng.*, 164, 107867. https://doi.org/10.1016/j.compchemeng.2022.107867.

Singhal, M., Marchetti, A.G., Faulwasser, T., Bonvin, D., 2018. Active directional modifier adaptation for real-time optimization. *Comput. Chem. Eng.*, 115, 246–261. https://doi.org/10.1016/j.compchemeng.2018.02.016.

Tian, Y., Du, W., Qian, F., 2013. Fault Detection and Diagnosis for Non-Gaussian Processes with Periodic Disturbance Based on AMRA-ICA. *Ind. Eng. Chem. Res.* 52(34), 12082–12107. https://doi.org/10.1021/ie400712h.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program*., 106, 25−57. https://doi.org/10.1007/s10107-004-0559-y.

Williams, T.J., Otto, R.E., 1960. A generalized chemical process model for the investigation of computer control. *IEEE Trans. Commun*. 79(5), 458–473. 10.1109/TCE.1960.6367296.

Yip, W.S., Marlin, T.E., 2004. The effect of model fidelity on real-time optimization performance. *Comput. Chem. Eng.*, 28(1–2), 267–280. https://doi.org/10.1016/S0098-1354(03)00164-9.

Zhang, Y., Monder, D.J., Forbes, J.F., 2002. Real-time optimization under parametric uncertainty: a probability constrained approach. *J. Process Control* 12(3), 373–389. https://doi.org/10.1016/S0959-1524(01)00047-6.

Appendix

Table A1: Data for the evaporator scenario.

	Mean processing cost $\bar{M}(\$/kg)$					Mass processed $m \times 10^3(kg)$				
T	MA	dMAIS 2	dMAIS 2(-)	dMAIS 1	dMAIS 1(-)	MA	dMAIS 2	dMAIS 2(-)	dMAIS 1	dMAIS 1(-)
2000	N/A	654	728	566	566	N/A	481	535	473	617
2500	724	613	694	540	540	524	601	695	590	778
3000	764	644	691	593	593	603	723	769	694	845
3500	690	579	614	622	622	789	969	1,029	792	940
4000	600	558	573	669	669	1,057	1,215	1,236	857	998
4500	526	535	595	644	644	1,413	1,267	1,401	966	1,168

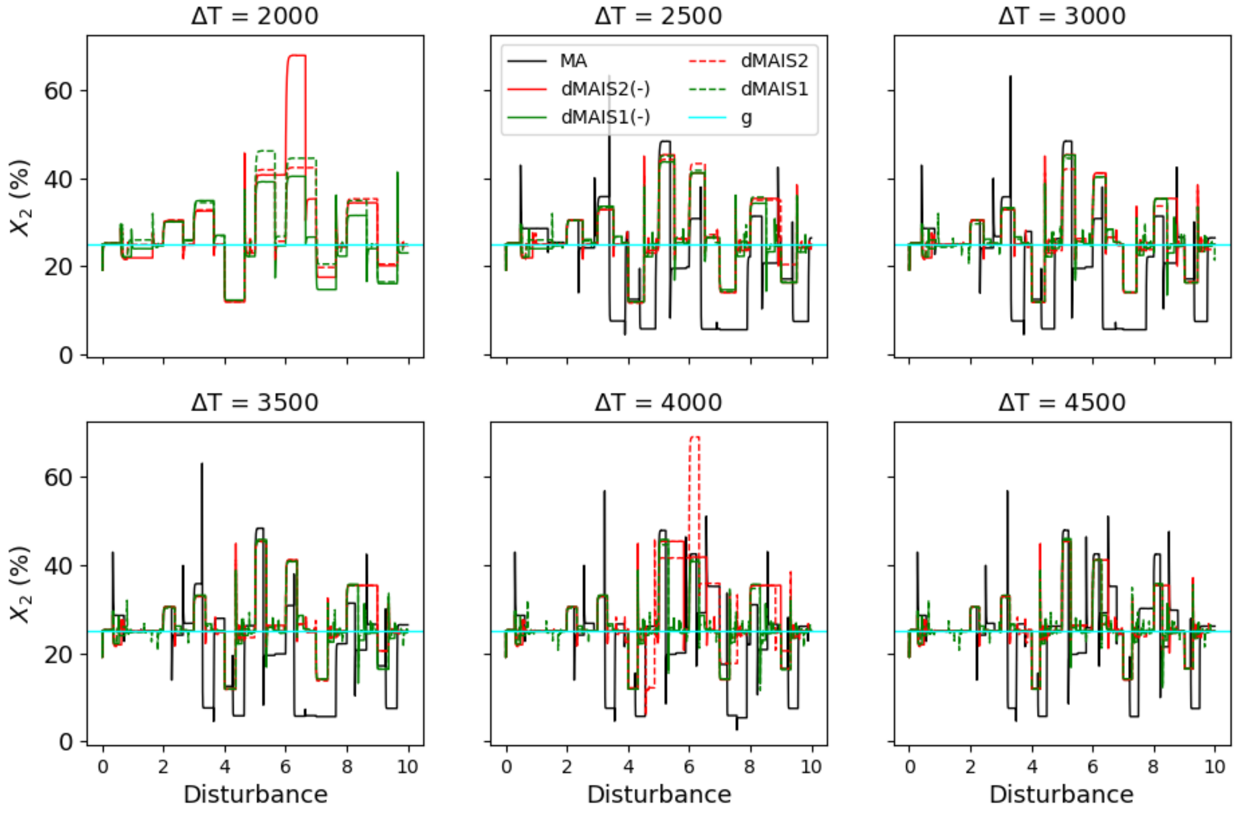


Figure A1: Constraint trajectories for evaporator case, increasing disturbance periods

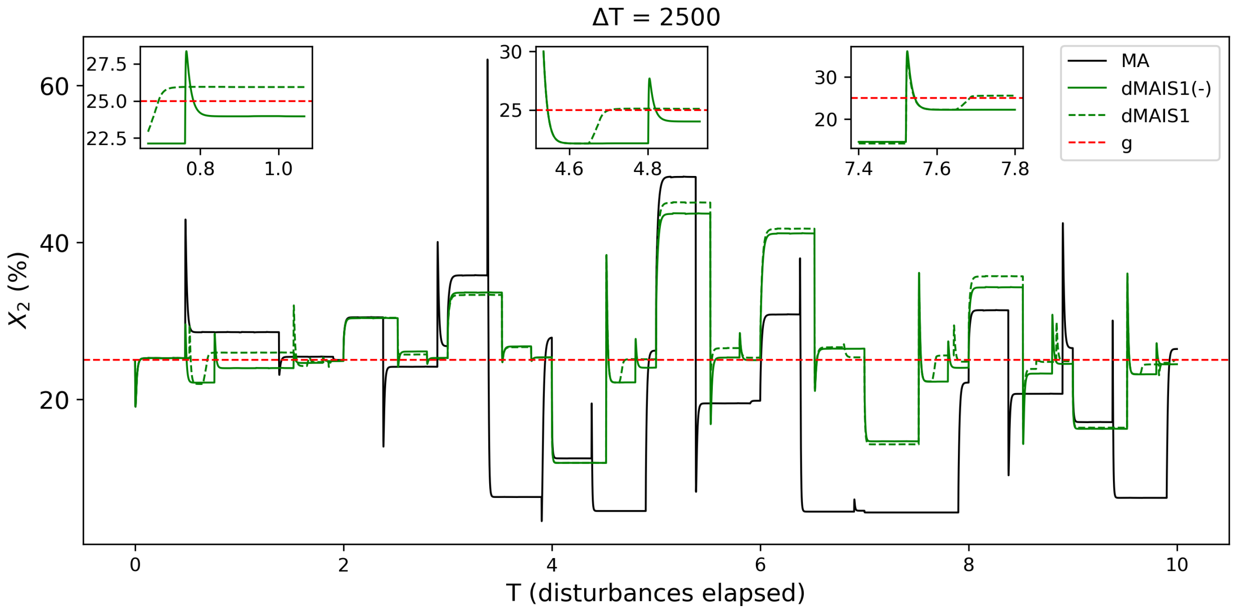


Figure A2: Profit rate (top) and constraint (bottom) for the evaporator case study under dMAIS, dMA (scenario 1), and dMA (scenario 2)