Internal Modifier Adaptation for the Optimization of Large-Scale Plants with Inaccurate Models.

Citation for published version:
Papasavvas, A & Francois, G 2019, 'Internal Modifier Adaptation for the Optimization of Large-Scale Plants with Inaccurate Models.', *Industrial & Engineering Chemistry Research*, vol. 58, no. 30, pp. 13568-13582. https://doi.org/10.1021/acs.iecr.9b00246

Digital Object Identifier (DOI):
10.1021/acs.iecr.9b00246

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Industrial & Engineering Chemistry Research

Internal Modifier Adaptation for the Optimization of Large-Scale Plants with Inaccurate Models

Aris Papasavvas* and Gregory Francois*

School of Engineering, Institute for Material and Processes,

The University of Edinburgh, Edinburgh EH93FB, Scotland, United Kingdom

E-mail: A.Papasavvas@sms.ed.ac.uk; Gregory.Francois@ed.ac.uk

Abstract

Modifier adaptation (MA) methods are iterative model-based real-time optimization (RTO) methods with the proven ability to reach, upon converge, the unknown optimal steady-state operating conditions of a plant despite plant-model mismatch and disturbances. So far, MA has been applied to small-scale but never – to the best of the authors' knowledge – to large-scale systems, the optimization of which being, in practice, a very difficult engineering problem. While standard MA uses plant measurements of the cost and constraints only, in this article, a new MA approach is proposed, namely *Internal Modifier Adaptation (IMA)*, which allows *the use of all available plant measurements* leading to corrections at the level of the inner structure of the model. This article also provides a mathematical proof that IMA preserves the property of MA methods to reach the optimal inputs of the plant upon convergence. The application and the benefits of the proposed method are illustrated through two large-scale simulated case studies: (i) a steel-making plant, and (ii) the Tennessee Eastman challenge problem.

Introduction

Industrial processes are operated via the manipulation of input variables. A "driving force" is required to take educated decisions and perform a systematic update of the inputs with the potential to maximize performances and enforce the satisfaction of operational constraints. Process optimization methods can be classified into two categories depending on the driving force. On the one hand, methods like evolutionary techniques such as steepest-decent methods, heuristic search methods (e.g. Nelder-Mead[1]), or evolutionary optimization[2], use past and current plant measurements for choosing the next set of inputs. On the other hand model-based methods make an explicit use of the available model of the plant. They are thus more suited to complex and constrained optimization problems, e.g., when the number of inputs grows large, provided the model is suitable for optimization, i.e., the optimization problem can be solved efficiently, in a reasonable amount of time, with limited risks of failure. However, the fact that the available models are often inaccurate generally leads to suboptimal operation and constraints violation.

When it comes to large-scale systems, the available model is often available in the form of a network of interconnected unit models, developed using dedicated software, that mimic the structure of the plant. The trouble is that plant-model mismatch in any of the unit models typically affects the whole network with the possibility of being amplified. This happens when there is recycling, i.e., when the outputs of a unit model i have an impact on another or several other unit models, which affect, in turn, the inputs of the unit model i. This intuitive idea has been confirmed[3] through a very simple example, whereby the aggregation of unit models into a single model leads to the amplification of the uncertainty of each unit model, making the overall model more uncertain than its sub-parts. The most obvious way to avoid this is to use accurate models, but this indeed also goes with caveats and can become inappropriate, even if accurate models are available, as it tends to increase the complexity of the resulting large-scale non-linear program (NLP). As a result, the numerical solver may not converge in a reasonable time, or fail to find a solution. Because real-time optimization (RTO) methods are capable of using inaccurate models, one solution would be the development of RTO tailored to large-scale interconnected systems, which would avoid the amplification of uncertainties.

RTO methods combine the use of an inaccurate model and of plant measurements to improve the performances of the plant, combining the pros of evolutionary and of model-based optimization techniques. Modifier adaptation (MA[4]) is one such method with the proven ability to reach, upon converge, the unknown plant optimal steady-state operating conditions despite plant-model mismatch and disturbances. MA uses *some* plant measurements to add input-affine corrections to the prediction of the cost and constraints, which corresponds to the minimal modifications enabling optimality upon convergence, which are sufficient to correct the way the modified model predicts the conditions of optimality of the plant. This method is gaining interest in the RTO research community and many extensions has been suggested to alleviate four of its most typical limitations (a-d).

Firstly, (a) ways to obtain accurate estimations of the steady-state plant gradients from noisy measurements have been studied over the past few years. One option is to combine MA with quadratic approximation approaches used in derivative-free optimization methods, as suggested in Gao et al. (2016)[5]. Alternatively, Gaussian regression techniques[6], trained with plant measurements of the cost and constraints can be used to reduce plant-model mismatch. Also, transient measurements can be used to infer steady-state plant gradients[7–10]. On the other hand, Nested Modifier Adaptation (NMA[11,12]) avoids these estimations by using a nested architecture with a gradient-free optimization algorithm to update the modifiers. Next, (b) model adequacy[13] is a condition the model must satisfy to enable convergence of MA to the plant plant optimal inputs. But this condition must be met by the model at the unknown plant optimal inputs. Methods to either enforce[14], or increase the chances[15–17], that model adequacy holds have been recently developed. Next, (c) guarantees for safe, i.e. with feasible plant iterates, convergence to the plant optimum have to been looked for. It has been proposed either to add penalty terms to the cost and constraints functions of the model[18], to limit the step size between iterates[19], or to implement exponential filtering[20]. Last, (d) the scalability of MA methods to large-scale non-linear plants remains an open issue. A first attempt resulted into the so-called distributed MA framework (DMA[21,22]), whereby it is proposed to apply MA to interconnected subsystems aiming at maintaining their cost and constraints private, or to cases whereby centralized computing units are not available. DMA uses *some* plant measurements, i.e., the interconnection variables between the subsystems together with measurements of the cost and constraints of each subsystem, to iteratively modify the available model and ultimately reach the optimal steady-state of the interconnected plant.

In this article, it is proposed to merge the ideas behind both Output Modifier Adaptation (MAy[4]) and DMA, into a new MA algorithm: *Internal Modifier Adaptation* (IMA). It is shown that IMA enables to use *all* available plant measurements in the decision-making process. Methods for computing the modifier terms are also provided and IMA is proven to preserve the desired property of MA schemes, i.e., to reach a KKT point of the plant upon convergence.

After a short review of some conceptual aspects of MA methods, a new way to interpret RTO problems is proposed in Section 3. Next, in Section 4, two new MA algorithms are proposed, analyzed and compared. Section 5 illustrates the concepts presented in Section 4 through two simulated case studies: (i) a steel-making plant and (ii) the Tennessee Eastman challenge process. Finally, Section 6 concludes the paper. Supporting information provides the reader with the mathematical proof of Theorem 1 and with details about the case studies of Section 5.

A Short Review of the State-of-the-Art in Modifier Adaptation Methods

In this section, once the optimization problem is formulated, the conceptual ideas behind most MA methods are shortly reviewed. It is shown that each MA approach can indeed be distinguished by the way the decision-making problem is formulated, i.e., the definitions of the model, the plant, the cost, the constraints and the level at which measurements-based corrections are performed. Block-oriented descriptions are proposed to illustrate how these methods use plant measurements and implement corrections, as an attempt to improve the insight of the reader.

Optimization problem

Hereafter, the subscript $(.)_p$ indicates that a quantity is related to the plant. The problem of finding the optimal operating conditions of the plant can be formulated mathematically as a nonlinear program (NLP):

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

where $\boldsymbol{u} \in \mathbb{R}^{n_u}$ are the input variables, $\boldsymbol{y}_p \in \mathbb{R}^{n_y}$ are the plant measured outputs, $\boldsymbol{F}_p(\boldsymbol{u})$ is the mapping between $\boldsymbol{u}$ and $\boldsymbol{y}_p$, $\phi \in \mathbb{R}$ is the cost function, $\boldsymbol{g} \in \mathbb{R}^{n_g}$ is the vector of constraint functions, and $\boldsymbol{q}_p \in \mathbb{R}^{n_q}$ is the subset of the measured variables $\boldsymbol{y}_p$ affecting the cost and constraints functions[*1].

In practice $\boldsymbol{F}_p(\boldsymbol{u})$ is not perfectly known, and only an approximate model $\boldsymbol{F}(\boldsymbol{u})$ of the

[*1] $\boldsymbol{q}_p$ is introduced and used in this article in order to clearly highlight that *not all* measured variables always affect the cost or constraints.

input-output mapping is available. With $\boldsymbol{F}(\boldsymbol{u})$, the solution to Problem (1) can be approached by solving the following NLP:

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

Due to plant-model mismatch, generally $\boldsymbol{u}^\star \neq \boldsymbol{u}_p^\star$, hence the need for RTO methods. This stems from the fact that cost and constraints values and gradients, which are the KKT elements[23] of any optimization problem, are not the same for the model and the plant.

Modifier Adaptation (MA)

The main idea behind MA is to use plant measurements to modify the cost and constraint functions in such a way that the KKT elements of the model match those of the plant at each iteration, and thus also upon convergence. Said differently: denoting $X(\boldsymbol{u}) := \{\ \Phi(\boldsymbol{u})$, $G_1(\boldsymbol{u})$, $\ldots$, $G_{n_g}(\boldsymbol{u})\}$, then for each iteration k, the aim is to have:

$$\{X(\boldsymbol{u}_k), \nabla_u X(\boldsymbol{u}_k)\} = \{X_p(\boldsymbol{u}_k), \nabla_u X_p(\boldsymbol{u}_k)\}, \tag{3}$$

where $\nabla_u(\cdot)$ denotes the gradient operator w.r.t. $\boldsymbol{u}$. To enforce (3) at each k, MA suggests that the modeled cost and constraints functions are augmented by the addition of affine-in-input functions, i.e., $\forall X = \{\Phi,\ G_1,\ \ldots,\ G_{n_g}\}$:

$$X_{m,k}(\boldsymbol{u}) = X(\boldsymbol{u}) + \alpha_{X,k}(\boldsymbol{u}),$$

where:

$$\alpha_{X,k}(\boldsymbol{u}) := \varepsilon_{X,k} + \boldsymbol{\lambda}_{X,k}^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k),$$

with $\varepsilon_{X,k} \in \mathbb{R}$ and $\boldsymbol{\lambda}_{X,k} \in \mathbb{R}^{n_u}$ being the zeroth- and first-order cost and constraint modifiers, respectively:

$$\begin{aligned}
\varepsilon_{X,k} &:= X_p(\boldsymbol{u}_k) - X(\boldsymbol{u}_k), \\
\boldsymbol{\lambda}_{X,k} &:= \nabla_u X_p(\boldsymbol{u}_k) - \nabla_u X(\boldsymbol{u}_k).
\end{aligned}$$

Thus, the following modified model-based optimization problem is solved:

$$\begin{aligned}
\boldsymbol{u}_{k+1}^{\star} := \arg\min_{\boldsymbol{u}} \quad & \Phi_{m,k}(\boldsymbol{u}) \\
\text{s.t.} \quad & \boldsymbol{G}_{m,k}(\boldsymbol{u}) \leq \boldsymbol{0},
\end{aligned} \tag{4}$$

and the next operating point $\boldsymbol{u}_{k+1}$ is generally determined by applying a first-order filter:

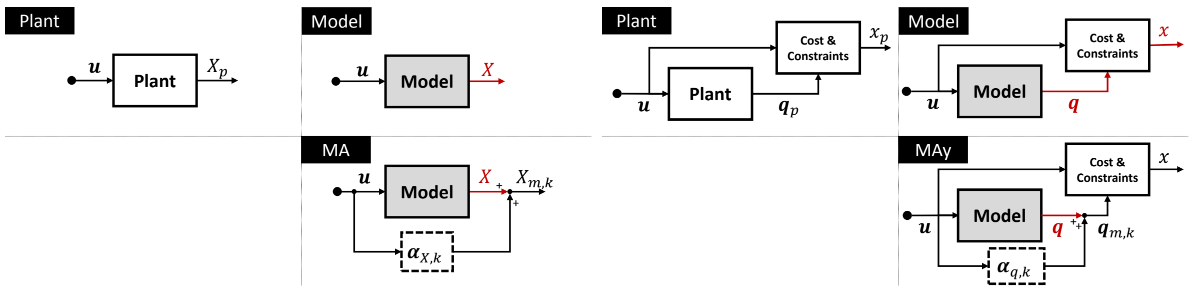
$$\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + \boldsymbol{K}(\boldsymbol{u}_{k+1}^{\star} - \boldsymbol{u}_k), \tag{5}$$

where $\boldsymbol{K} \in \mathbb{R}^{n_u \times n_u}$ is a gain matrix, typically diagonal, with diagonal elements $K_i \in (0, 1]$, $\forall i \in [1, n_g]$. $\boldsymbol{u}_{k+1}$ is uniformly applied to the plant until steady state is reached and the whole procedure is repeated until convergence.

Figure 1a illustrates by means of a block-oriented description how these corrections are performed . With MA, the decision-making process can be interpreted as follows:

- **The plant** is viewed as a set of mappings between the manipulated variables $\boldsymbol{u} \in \mathbb{R}^{n_u}$ and the plant cost and constraints at steady state $X_p(\boldsymbol{u})$.
- **The model** is an approximation of the plant mappings, providing estimates $X(\boldsymbol{u})$ of $X_p(\boldsymbol{u})$.
- **Iterative input-affine modifications** are performed to correct the way the modified model predicts the KKT elements of the plant, leading to plant inputs update.

Note that *hereafter, gray boxes will be used for the model in order to distinguish it from* *the plant (white boxes), illustrating thus plant-model mismatch. Red arrows and fonts are for uncertain or uncorrected variables, while black is used for accurate or corrected variables. Clouds depict networks and double vertical lines concatenate or de-concatenate vectors.*



(a) MA. (b) MAy.

Figure 1: Block-oriented description of MA and MAy.

This modification of the cost and constraints functions enforces the matching of the affine properties of both the modified model-based and the plant optimization problem at $\boldsymbol{u}_k$. This is how the conditions of optimality of the model and the plant are reconciled, since this leads upon convergence to the equivalence of the (in)existence of feasible descent directions for the modified model and for the plant. Hence, upon convergence of MA to $\boldsymbol{u}_\infty$, no feasible descent directions exist for the modified model and thus for the plant, which is how $\boldsymbol{u}_\infty$ can be easily shown to be a KKT point for the plant.

In summary, one key message of MA is that *one way to guarantee plant optimality upon convergence of a RTO method is by enforcing the affine properties of the (modified) model to match those of the plant at $\boldsymbol{u}_k, \forall k$*, an observation that has been suggested[4] but never explicitly stated. Note that the generalization of this message is the topic of Theorem 2, as will be seen later. From this viewpoint, it is clear that MA performs the minimal number of corrections to enable optimality upon convergence, i.e., since only the values and gradients of the cost and constraints are reconciled, which is sufficient to allow the matching of the affine properties of the optimization problem. However, we argue next that performing more corrections can be beneficial.

Output Modifier Adaptation (MAy)

MAy differs from standard MA in that instead of directly correcting the way the KKT elements are predicted by the augmented model, modifications are performed to correct the predictions of $\boldsymbol{q}_p$. Before going further, the following assumption is required[*2]:

Assumption 1 *The cost and constraints functions are known functions of the decision variables $\boldsymbol{u}$ and $\boldsymbol{q}_p$ the measured outputs affecting them, i.e., respectively they can be written as $\phi(\boldsymbol{u}, \boldsymbol{q}_p)$ and $\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{q}_p)$.*

As depicted in Figure 1b, provided Assumption 1 holds, with MAy the decision-making process can be interpreted as follows:

- **The plant** is viewed as a mapping between the decision variables $\boldsymbol{u}$ and $\boldsymbol{q}_p$, the latter being the measured variables used for the calculation of the cost and constraints functions $x(\boldsymbol{u}, \boldsymbol{q}_p) := \{\phi(\boldsymbol{u}, \boldsymbol{q}_p), g_1(\boldsymbol{u}, \boldsymbol{q}_p), \ldots, g_{n_g}(\boldsymbol{u}, \boldsymbol{q}_p)\}$.

- **The model** is an approximation of the plant mapping providing $\boldsymbol{q}$, i.e., the estimates of $\boldsymbol{q}_p$.

- **Iterative input-affine modifications** are performed to correct the way the modified model predicts $\boldsymbol{q}_p$.

MAy modifies the model predictions $\boldsymbol{q}(\boldsymbol{u})$, such that: $\boldsymbol{q}_{m,k}(\boldsymbol{u}) := \boldsymbol{q}(\boldsymbol{u}) + \boldsymbol{\alpha}_{q,k}(\boldsymbol{u})$, with $\boldsymbol{\alpha}_{q,k}(\boldsymbol{u}) := \boldsymbol{\varepsilon}_{q,k} + \boldsymbol{\lambda}_{q,k}^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k)$, where $\{\boldsymbol{\varepsilon}_{q,k}, \boldsymbol{\lambda}_{q,k}\}$ are respectively the zeroth- and first-order modifiers. Since $\boldsymbol{q}_{m,k}$ affects the way the values and gradients of the cost and constraints are calculated, this leads to an indirect correction of the affine properties of the modified model-based optimization problem, while providing additional improvements (e.g., better convergence properties) compared to MA, thanks to deeper corrections of the model[15].

[*2]This assumption does not imply that the model is assumed to be structurally correct, but that the cost and constraints are correctly predicted when the values of $\boldsymbol{q}_p$ are accurately predicted. Structural and parametric plant-model mismatch are still present between $\boldsymbol{F}(\boldsymbol{u}) \neq \boldsymbol{F}_p(\boldsymbol{u})$ and generally $\boldsymbol{q} \neq \boldsymbol{q}_p$.

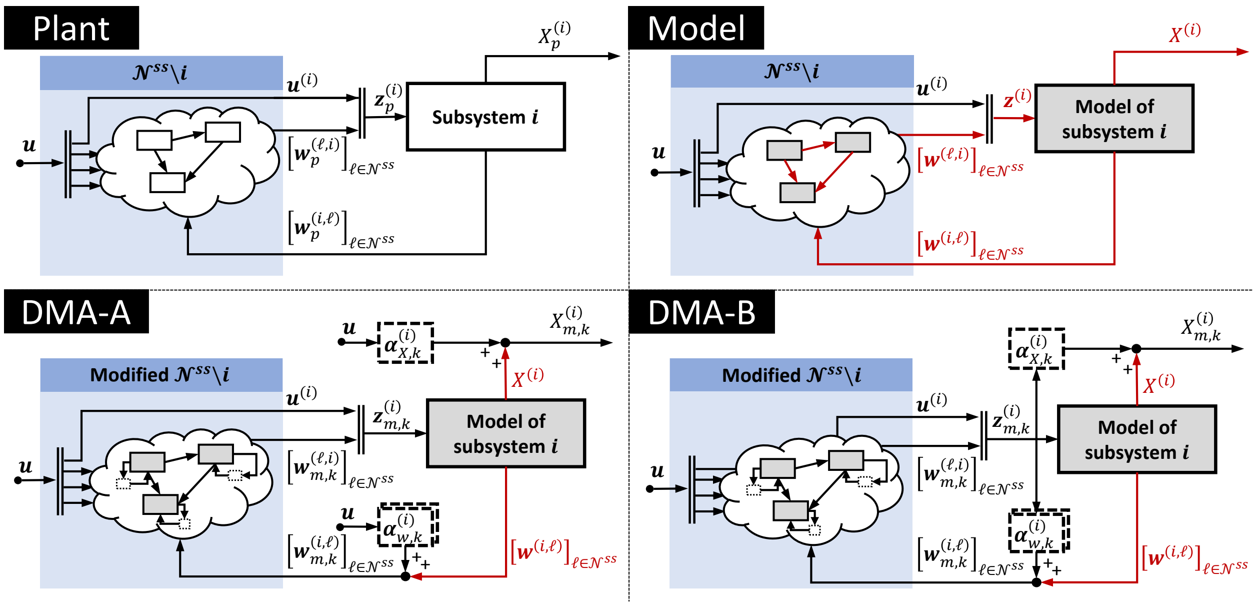

Plant
Model
DMA-A
DMA-B
$\mathcal{N}^{ss}\setminus i$
Modified $\mathcal{N}^{ss}\setminus i$
u
$u^{(i)}$
$z_p^{(i)}$
Subsystem i
$X_p^{(i)}$
$[w_p^{(\ell,i)}]_{\ell\in\mathcal{N}^{ss}}$
$[w_p^{(i,\ell)}]_{\ell\in\mathcal{N}^{ss}}$
$z^{(i)}$
Model of subsystem i
$X^{(i)}$
$[w^{(\ell,i)}]_{\ell\in\mathcal{N}^{ss}}$
$[w^{(i,\ell)}]_{\ell\in\mathcal{N}^{ss}}$
$\alpha_{X,k}^{(i)}$
$X_{m,k}^{(i)}$
$z_{m,k}^{(i)}$
$[w_{m,k}^{(\ell,i)}]_{\ell\in\mathcal{N}^{ss}}$
$[w_{m,k}^{(i,\ell)}]_{\ell\in\mathcal{N}^{ss}}$
$\alpha_{w,k}^{(i)}$

Figure 2: Block-oriented description of DMA.

Distributed Modifier Adaptation (DMA)

MA has been recently improved to account for the specificities of large-scale highly interconnected plants, with the associated difficulties, such as high numbers of units or when units cannot (or does not want to) share specific information (such as its own cost or constraints) with the others. With the so-called Distributed Modifier Adaptation (DMA), the whole plant is seen as a network of interconnected subsystems, which mimics the structure of the plant, i.e., each element of the network is typically of one unit (or of a small set of units), connected to the others via interconnection variables. This formulation can also be advantageous for distributing the computational load[21] of the model-based optimization problem (4), while allowing the subsystems to keep their cost and constraints private[22].

As depicted in Figure 2, with DMA the decision-making problem can be interpreted as follows:

- **The plant** is viewed as a network $\mathcal{N}^{ss} := \{1, \ldots, n_{ss}\}$ of n_{ss} interconnected subsystems, e.g., compressors stations interconnected with pipes[24].

  Each subsystem, indexed by i, can be seen as a mapping between its own inputs $z_p^{(i)}$

and outputs, which are both cost and constraints $X_p^{(i)}\ (\boldsymbol{z}_p^{(i)}) := \{\Phi_p^{(i)}(\boldsymbol{z}_p^{(i)}),\ G_{p,1}^{(i)}(\boldsymbol{z}_p^{(i)})$ $, \dots, G_{p,n_{g,i}}^{(i)}(\boldsymbol{z}_p^{(i)})\}$, and the aggregated (concatenated) output-interconnection variables $\left[\boldsymbol{w}_p^{(i,\ell)}\right]_{\ell\in\mathcal{N}^{ss}}$, i.e. $\forall \ell \in \mathcal{N}^{ss}$:

$$\left[\boldsymbol{w}_p^{(i,\ell)}\right]_{\ell\in\mathcal{N}^{ss}} := \left[\boldsymbol{w}_p^{(i,1)\mathsf{T}}, \dots, \boldsymbol{w}_p^{(i,n_{ss})\mathsf{T}}\right]^{\mathsf{T}}. \tag{6}$$

Note that this notation is used throughout this article for aggregated vectors. The inputs of a subsystem i, $\boldsymbol{z}_p^{(i)} \in \mathbb{R}^{n_{z,i}}$, are the manipulated variables, $\boldsymbol{u}_p^{(i)} \in \mathbb{R}^{n_{u,i}}$ and the aggregated input-interconnection variables, $\left[\boldsymbol{w}_p^{(\ell,i)}\right]_{\ell\in\mathcal{N}^{ss}}$, which are indeed the outputs of the subsystems $\ell \in \mathcal{N}^{sm}$ affecting subsystem i. With DMA, the superscript $(\cdot)^{(\ell,i)}$ refers to as the interconnection variables from the subsystem ℓ to the subsystem i, also a superscript $(\cdot)^{(i)}$ refers to a variable, or function, of subsystem i.

- **The models** returns approximations $X^{(i)}(\boldsymbol{z}^{(i)}) := \{\Phi^{(i)}(\boldsymbol{z}^{(i)}), G_1^{(i)}(\boldsymbol{z}^{(i)}), \dots, G_{n_{g,i}}^{(i)}(\boldsymbol{z}^{(i)})\}$ and $\left[\boldsymbol{w}^{(i,\ell)}\right]_{\ell\in\mathcal{N}^{ss}}$ of the plant variables $X_p^{(i)}(\boldsymbol{z}_p^{(i)})$ and $\left[\boldsymbol{w}_p^{(i,\ell)}\right]_{\ell\in\mathcal{N}^{ss}}$, respectively.

- **Iterative input-affine modifications** are performed to correct the way the modified model predicts the KKT elements and the interconnection variables of the plant.

To implement input-affine corrections to a DMA-network, two main methods – hereafter referred to as DMA-A[21] and DMA-B[22] – have been proposed, which imply the iterative identification of virtual affine functions $\{\alpha_{X,k}^{(i)}(\boldsymbol{u}), \alpha_{w,k}^{(i)}(\boldsymbol{u})\}$ or $\{\alpha_{X,k}^{(i)}(\boldsymbol{z}_m^{(i)}), \alpha_{w,k}^{(i)}(\boldsymbol{z}_m^{(i)})\}$, respectively. Figure 2 illustrates the way these two methods are implemented. The main difference between the two is that with DMA-A, affine functions of $\boldsymbol{u}$ are used to implement corrections at the level of the subsystems, while affine functions of $\boldsymbol{z}^{(i)}$ are considered with DMA-B instead. The latter allows DMA-B to compute modifiers locally, and potentially to keep the costs and constraints of each subsystem private[22].

The representation of the plant as a network of physical units is indeed what enables DMA to use the plant measured interconnection variables in the optimization framework, in addition to the measurements or estimates of the KKT elements that standard MA uses. Notice that the use of interconnection variables has also been reported to lead to improvements in terms of convergence speed[21], although investigations have not been pushed further.

An alternative to mimicking the plant structure

As seen, the affine-in-input corrections of the cost and constraints prevents MA to converge anywhere else than to KKT point of the plant. Also, it has been seen that (i) input-affine corrections of the measured outputs (MAy) and (ii) input-affine corrections of the interconnection variables (DMA) also leads to a correction of the prediction of the KKT elements. In addition, both MAy and DMA often result in better (i.e., faster and safer) convergence properties. Additionally, DMA allows the distribution (and potentially reduction) of the computational load while providing the different agents of the plant with privacy whenever necessary or compulsory.

To combine the pros of (i) and (ii), we propose to merge DMA and MAy into a single approach, namely *Internal Modifier Adaptation* (IMA). This merging impacts how the problem is interpreted since DMA treats the plant as a network of subsystems, while MAy considers the plant as as single system and implement corrections at the level of the measured outputs.

With IMA, the plant is seen as *two* interconnected networks. The first is the network of interconnected sub-models that are mappings between, on one hand, their input interconnection *and* manipulated variables, and, on the other hand, their output interconnection *and* output variables affecting the cost and constraints. The second network consists of sub-costs and sub-constraints that are mappings from the output variables that affect cost and constraints of the sub-models to the corresponding costs and constraints.

By doing so, the structure (to be built) of the model is at the core of the definition of the structure of the problem. In other words, instead of mimicking the organization of the plant like with DMA, the specificities of the model structure and of its contributing equations are exploited, together with the list of available plant measurements when representing the plant and dividing it into smaller interconnected subsystems.

Model pre-processing

To obtain the aforementioned representation – with two interconnected networks – of the plant, the following model pre-processing procedure is proposed and illustrated by means of the following mathematical example.

Example 1 *(The Un-Processed Model) Consider the steady-state model of a single unit plant with seven process variables* $\{v_1, \ldots, v_7\}$*. The manipulated variables are* $\{v_3, v_4\}$ *and* $\{v_1, v_5, v_7\}$ *are measured on the plant. All variables are connected by the five equations (a)-(e), and (f) provides the cost. Since this is a single physical unit, one can state without further analysis that DMA and MA are identical for this problem, with all equations (a-f) taken as a whole.*

$$0 = v_1 + v_2 + v_5, \tag{a}$$

$$0 = v_2 + v_3, \tag{b}$$

$$0 = v_4 + v_6, \tag{c}$$

$$0 = v_5 + v_6, \tag{d}$$

$$0 = v_5 + v_6 + v_7, \tag{e}$$

$$\phi = v_1^2 + v_7^2. \tag{f}$$

This model approximates the single-unit plant that is defined by the following steady-state equations (unknown a priori):

$$0 = v_{p,2} + v_3 + v_{p,5}, \tag{a_p}$$

$$0 = v_{p,2} + v_3, \tag{b_p}$$

$$0 = v_3 + v_{p,5} + v_{p,7}, \tag{c$_p$}$$

$$0 = v_{p,1} + v_4 + v_{p,6}, \tag{d$_p$}$$

$$0 = v_4 + v_{p,6}, . \tag{e$_p$}$$

Assumption 1 holds, thus the cost $\phi_p = v_{p,1}^2 + v_{p,7}^2$ is computed as for the model, i.e. with equation (f).

Pre-processing procedure

The model pre-processing is done in three steps:

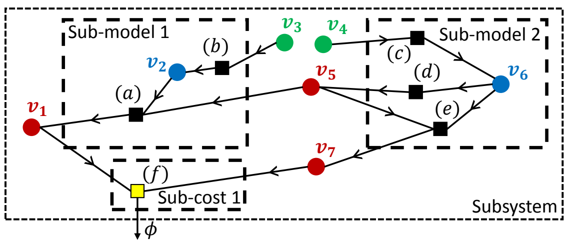
Step 1: (Variables Classification) Model variables are classically classified as either manipulated variables $\boldsymbol{u} \in \mathbb{R}^{n_u}$, measured output variables $\boldsymbol{y} \in \mathbb{R}^{n_y}$, or states.

Step 2: (Network Clustering) The network generated by the equations linking these variables is analyzed and clustered into sub-models, sub-costs, and sub-constraints. *Sub-models* are defined as the equations which connect states (only) to measured and/or manipulated variables. Similarly, *sub-costs and sub-constraints* correspond to equations allowing the calculation of costs or constraints.

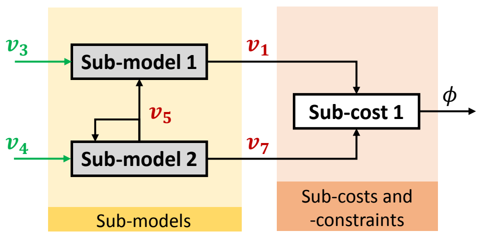
Step 3: (Building the Directed Network) The variables that are not part of any sub-model are defined as interconnection variables since they connect sub-models. Each of these variables must be computed from one, and only one, sub-model. Therefore, these variables are denoted as output-interconnection variables of the sub-model that computes them. For the other sub-models impacted by these variables, they are input-interconnection variables. Doing this for each interconnection variables results in a directed network (or directed graph, or digraph[25]).

The application of this procedure to Example 1 is given hereafter:

Example 2 *(Model Pre-Processing) The application of the model pre-processing to the model of Example 1 is illustrated in Figure 3. Step 1 is illustrated by the colors that are given to the different variables, green for the manipulated variables $\boldsymbol{u} = [v_3,\, v_4]^\mathsf{T}$, red for the measured* *variables $\boldsymbol{y} = [v_1,\, v_5,\, v_7]^\mathsf{T}$, and blue for the states $[v_2,\, v_6]^\mathsf{T}$. The equations (a-e) are depicted using black squares. The cost function (f) is depicted with a yellow square to highlight that it is a known function of measured and manipulated variables. Step 2 is represented in Figure 3a with dashed boxes ("sub-model 1", "sub-model 2", and "sub-cost 1"). The dotted box "subsystem" illustrates here that DMA would have treated the problem as a single subsystem since, as mentioned before, all equations represent the behavior of a single unit. Finally, by applying step 3 to Figure 3a, Figure 3b is obtained, where v_1 and $\{v_5, v_7\}$ are computed with sub-models* 1 *and* 2*, respectively, illustrating the flow of information through the network of sub-systems, -costs and -constraints. This flow of information is also represented in Figure 3a with the arrows on the connection links.*



(a) Steps 1-2 (3 with the arrows on the links).



(b) Step 3.

Figure 3: Application of the model pre-processing to Example 1.

Definitions and Notations

The model structure resulting from the application of the model pre-processing procedure leads to a network $\mathcal{N}^{sm} := \{1, \ldots, n_{sm}\}$ of n_{sm} sub-models interconnected with a network $\mathcal{N}^{sc} := \{1, \ldots, n_{sc}\}$ of n_{sc} sub-costs or sub-constraints.

For the sub-models, a similar notation to DMA is used. The inputs $\boldsymbol{z}^{(i)}$ of a sub-model $i \in \mathcal{N}^{sm}$ result from the concatenation of $\boldsymbol{u}^{(i)} \in \mathbb{R}^{n_{u,i}}$ (the manipulated variables affecting i), and $\left[\boldsymbol{w}^{(\ell,i)}\right]_{\ell\in\mathcal{N}^{sm}}$ (the aggregated input-interconnection variables $\boldsymbol{w}^{(\ell,i)}$ connecting the sub-models $\ell \in \mathcal{N}^{sm}$ to i). The superscript $(\cdot)^{(i)}$ is introduced to refer to as sub-model i, and $(\cdot)^{(\ell,i)}$, to refer to as the directed interconnection from sub-model $\ell \in \mathcal{N}^{sm}$ to sub-model

$i \in \mathcal{N}^{sm}$. The outputs of sub-model i are denoted $\boldsymbol{y}^{(i)} \in \mathbb{R}^{n_{y,i}}$. The outputs in $\boldsymbol{y}^{(i)}$ that are used in the network $\mathcal{N}^{sc}$ by a sub-cost or -constraint j are denoted $\boldsymbol{q}^{[i,j]}$, and the outputs in $\boldsymbol{y}^{(i)}$ that are used in $\mathcal{N}^{sm}$ by a sub-model ℓ are denoted $\boldsymbol{w}^{(i,\ell)}$, the output-interconnection variables of i. The superscript $(\cdot)^{[i,j]}$ refers to the directed interconnection from a sub-model $i \in \mathcal{N}^{sm}$ to a sub-cost, or -constrain, $j \in \mathcal{N}^{sc}$. The mapping between $\boldsymbol{z}^{(i)}$ and $\boldsymbol{y}^{(i)}$ is denoted $\boldsymbol{f}^{(i)}$, i.e., $\boldsymbol{y}^{(i)} = \boldsymbol{f}^{(i)}(\boldsymbol{z}^{(i)})$. These notations are summarized in Figure 4 and Table 1. Finally, the two following assumptions are performed:

Assumption 2 (Sub-models) *Each sub-model $i \in \mathcal{N}^{sm}$ is such that $\forall \boldsymbol{z}^{(i)}$, the input-output mapping $\boldsymbol{f}^{(i)}(\boldsymbol{z}^{(i)})$ is injective, i.e., for any $\boldsymbol{z}^{(i)}$ there is only one $\boldsymbol{y}^{(i)}$ such that $\boldsymbol{y}^{(i)} = \boldsymbol{f}^{(i)}(\boldsymbol{z}^{(i)})$.*

Assumption 3 (Uniqueness of the Network Solution) *For one $\boldsymbol{u}_k$, there is only one $\boldsymbol{y}$ solving the model, i.e., only one $\boldsymbol{y} := [\boldsymbol{y}^{(1)\mathsf{T}}, \dots, \boldsymbol{y}^{(n_{sm})\mathsf{T}}]^{\mathsf{T}}$ such that $\boldsymbol{y}^{(i)} = \boldsymbol{f}^{(i)}(\boldsymbol{z}^{(i)})$, $\forall i \in \mathcal{N}^{sm}$.*

For sub-costs and -constraints: a sub-cost or -constraint $j \in \mathcal{N}^{sc}$, has two sets of inputs: $\boldsymbol{u}^{[i]}$, the subset of $\boldsymbol{u}$ affecting j, and $\left[\boldsymbol{q}^{[i,j]}\right]_{i\in\mathcal{N}^{sm}}$, the aggregated vector of all sub-models outputs affecting j. The notation $(\cdot)^{[j]}$ is used to refer to the sub-cost or -constraint j. The output of a sub-cost or -constraint is either a cost or a constraint. Figure 5 and Table 1 summarize these notations. Generalizing to all $j \in \mathcal{N}^{sc}$ leads to the following definitions:

$$\phi^{[j]}\left(\boldsymbol{u}^{[j]}, \left[\boldsymbol{q}^{[i,j]}\right]_{i\in\mathcal{N}^{sm}}\right) := \begin{cases} x^{[j]}, & \text{if } j \text{ is a sub-cost,} \\ 0, & \text{otherwise.} \end{cases}$$

$$g^{[j]}\left(\boldsymbol{u}^{[j]}, \left[\boldsymbol{q}^{[i,j]}\right]_{i\in\mathcal{N}^{sm}}\right) := \begin{cases} x^{[j]}, & \text{if } j \text{ is a sub-constraint,} \\ \emptyset, & \text{otherwise.} \end{cases}$$

The aggregated cost and constraints are defined as:

$$\phi(\boldsymbol{u}) := \sum_{j\in\mathcal{N}^{sc}} \phi^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big), \tag{7}$$

$$\boldsymbol{g}(\boldsymbol{u}) := \Big[g^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big)\Big]_{j\in\mathcal{N}^{sc}}, \tag{8}$$

respectively. Finally, the following assumption is performed:

Assumption 4 *The functions $x^{[j]}$, $\forall j \in \mathcal{N}^{sc}$, are twice continuously differentiable ($\mathcal{C}^2$) w.r.t. their inputs $\boldsymbol{u}^{[j]}$ and $\big[\boldsymbol{q}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}$.*

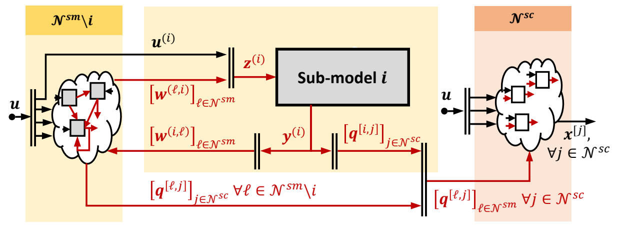

$\mathcal{N}^{sm}\backslash i$
$\boldsymbol{u}^{(i)}$
$\boldsymbol{z}^{(i)}$
Sub-model i
$\mathcal{N}^{sc}$
$\boldsymbol{u}$
$[\boldsymbol{w}^{(\ell,i)}]_{\ell\in\mathcal{N}^{sm}}$
$[\boldsymbol{w}^{(i,\ell)}]_{\ell\in\mathcal{N}^{sm}}$
$\boldsymbol{y}^{(i)}$
$[\boldsymbol{q}^{[i,j]}]_{j\in\mathcal{N}^{sc}}$
$\boldsymbol{x}^{[j]}, \forall j \in \mathcal{N}^{sc}$
$[\boldsymbol{q}^{[\ell,j]}]_{j\in\mathcal{N}^{sc}}\ \forall \ell \in \mathcal{N}^{sm}\backslash i$
$[\boldsymbol{q}^{[\ell,j]}]_{\ell\in\mathcal{N}^{sm}}\ \forall j \in \mathcal{N}^{sc}$

Figure 4: Interactions of a sub-model i with the networks $\mathcal{N}^{sm}$ and $\mathcal{N}^{sc}$.

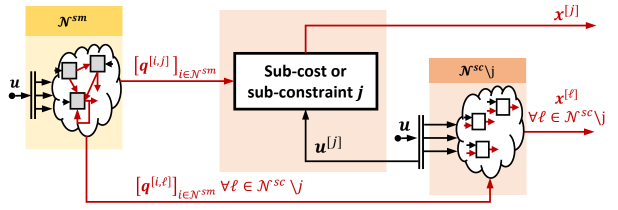

$\mathcal{N}^{sm}$
$\boldsymbol{x}^{[j]}$
$[\boldsymbol{q}^{[i,j]}]_{i\in\mathcal{N}^{sm}}$
Sub-cost or sub-constraint j
$\mathcal{N}^{sc}\backslash j$
$\boldsymbol{u}$
$\boldsymbol{x}^{[\ell]}$
$\forall \ell \in \mathcal{N}^{sc}\backslash j$
$\boldsymbol{u}^{[j]}$
$[\boldsymbol{q}^{[i,\ell]}]_{i\in\mathcal{N}^{sm}}\ \forall \ell \in \mathcal{N}^{sc}\backslash j$

Figure 5: Interactions of a sub-cost, or sub-constraint, j with the network $\mathcal{N}^{sm}$.

To illustrate the definitions of these notations example is continued next.

Example 3 *(Notations) According to Figure 3b, the outputs the two sub-models are $y^{(1)} := v_1$ and $\boldsymbol{y}^{(2)} := [v_5, v_7]^\mathsf{T}$. The interconnection variables are $w^{(2,1)} = v_5$, and $w^{(2,2)} = v_5$. The output variables affecting the cost are $q^{[1,1]} = v_1$ and $q^{[2,1]} = v_7$. The manipulated variables affecting the sub-models 1 and 2 are $u^{(1)} = v_3$, and $u^{(2)} = v_4$, respectively. The inputs of sub-models 1 and 2 are $\boldsymbol{z}^{(1)} = [u^{(1)},\, w^{(2,1)}]^\mathsf{T}$ and $\boldsymbol{z}^{(2)} = [u^{(2)},\, w^{(2,2)}]^\mathsf{T}$, respectively. Finally, the mappings $\boldsymbol{f}^{(1)}$ and $\boldsymbol{f}^{(2)}$ are the model equations $\{(a),(b)\}$ and $\{(c),(d),(e)\}$, respectively.*

Symbols	
n_{sm}	number of sub-models
$\mathcal{N}^{sm}$	network of n_{sm} sub-models
n_{sc}	number of sub-costs and -constraints
$\mathcal{N}^{sc}$	network of n_{sc} sub-costs and -constraints
$\boldsymbol{u}$	manipulated variables
$\boldsymbol{u}^{(i)}$	manipulated variables affecting sub-model i
$\boldsymbol{u}^{[j]}$	manipulated variables affecting sub-cost j
$\boldsymbol{w}^{(\ell,i)}$	interconnection variables linking sub-model ℓ to sub-model i
$\boldsymbol{z}^{(i)}$	inputs of sub-model i (composed of $\boldsymbol{u}^{(i)}$ and $\boldsymbol{w}^{(\ell,i)}$)
$\boldsymbol{q}^{[i,j]}$	outputs of sub-model i affecting sub-cost j
$\boldsymbol{y}^{(i)}$	outputs of sub-model i
$\boldsymbol{f}^{(i)}$	inputs-outputs mapping of sub-model i
$g^{[j]}$	sub-constraint j
$\boldsymbol{g}$	aggregated constraint (see Equation (8))
$\phi^{[j]}$	sub-cost j
ϕ	aggregated cost (see Equation (7))
$x^{[j]}$	sub-cost or sub-constraint j
$\boldsymbol{\varepsilon}_{y,k}^{(i)}$	zeroth-order modifier for IMA
$\boldsymbol{\lambda}_{y,k}^{(i)}$	first-order modifier for IMA
$\boldsymbol{\alpha}_{y,k}^{(i)}$	affine function correcting sub-model i at the k-th iteration (see Equation (14))
$\mathcal{A}_{y,k}$	set of all affine corrections $\boldsymbol{\alpha}_{y,k}^{(i)}$, i.e, $\mathcal{A}_{y,k} := \{\boldsymbol{\alpha}_{y,k}^{(1)}, \ldots, \boldsymbol{\alpha}_{y,k}^{(n_{sm})}\}$
Subscripts and superscripts	
$(\cdot)^{(i)}$	variables or functions related to sub-model i
$(\cdot)^{(\ell,i)}$	variables connecting sub-model ℓ to sub-model i
$(\cdot)^{[j]}$	variables or functions related to sub-cost or -constraint j
$(\cdot)^{[i,j]}$	variables connecting sub-model i to sub-cost or -constraint j
$(\cdot)_p$	variables or functions related to the plant
$(\cdot)_{m,k}$	variables or functions modified at iteration k
$[(\cdot)]_{i\in\mathcal{N}^{sm}}$	concatenation (according to Equation (6))
$(\cdot)\|_{\boldsymbol{u}_k}$	variables or functions evaluated at $\boldsymbol{u}_k$

Table 1: Summary of the notations used for IMA.

Representation of the Plant

Now that the pre-processed model and its notations have been introduced, we consider a plant represented, without loss of generality (when Assumption 1 holds), as in Figure 6: here, the plant is seen as a mapping $\boldsymbol{F}_p$ between its inputs $\boldsymbol{u}$ and its outputs $\boldsymbol{y}_p$, which can easily be reformulated to match the structure of the pre-processed model as n_{sm} mappings $\boldsymbol{F}_p^{(i)}, \forall i \in \mathcal{N}^{sm}$, from $\boldsymbol{u}$ to the corresponding sub-plant outputs $\boldsymbol{y}_p^{(i)}$. These outputs, like those of the model, are composed of interconnection variables $\boldsymbol{w}_p^{(i,\ell)}$, $\forall \ell \in \mathcal{N}^{sm}$, and variables affecting the sub-costs and constraints $\boldsymbol{q}_p^{[i,j]}$, $\forall i \in \mathcal{N}^{sc}$. Going further and representing the plant as in Figure 7 implicitly relies on the fact that the modeled network $\mathcal{N}^{sm}$ is a "consistent" representation of the plant, which is unfortunately not always possible.

As illustrated in Figure 8 and in the following Example, the existence of the plant mappings $\boldsymbol{F}_p^{(i)}, \forall i \in \mathcal{N}^{sm}$, does not always imply the existence of correlations (mappings) $\widetilde{\boldsymbol{f}}_p^{(i)}, \forall i \in \mathcal{N}^{sm}$ between $\boldsymbol{z}_p^{(i)}$ and $\boldsymbol{y}_p^{(i)}$, hence the introduction of $\widetilde{\boldsymbol{f}}_p^{(i)}$.

Example 4 *(Model Consistency) We now assume that the single unit plant is known, i.e. the equations (a_p)-(e_p) are known, and apply the pre-processing procedure to the equations (a_p)-(e_p). This would lead to the three sub-plants illustrated in Figure 9a. As shown before, the model (a)-(e) predicts the correlation (mapping) $v_1 = \boldsymbol{f}^{(1)}(v_3, v_5)$. But this correlation does not hold for the plant, as shown in Figure 9a since v_1 is independent of both v_3 and v_5 for the plant. So, the corresponding correlation (mapping) $\widetilde{\boldsymbol{f}}_p^{(1)}$ does not exist for the plant, i.e., variations of v_3 or v_5 do not imply variations of v_1, and v_1 can vary when v_3 and v_5 are constant. Notice that the plant mismatch in this illustrative example is quite extreme since the model correlation $v_1 = \boldsymbol{f}^{(1)}(v_3, v_5)$ does not even include a sub-set of the plant correlations. Such plant-model mismatch should rarely occur in practice, since it is expected that the available model predicts, at least, a sub-set of the plant correlations between known process variables. But, plant-model inconsistency (according to the definition of Assumption 5) could indeed happen more often in practice. A simple example is when the plant has more inputs than its model. When it happens, a variation of the plant outputs could be observed*

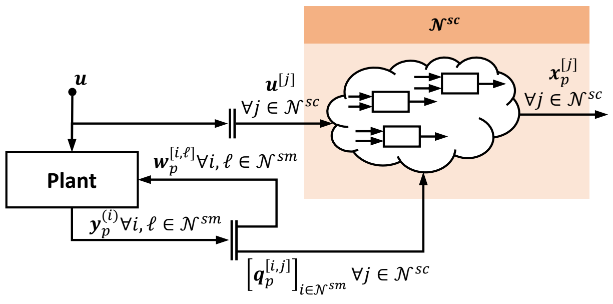

$\mathcal{N}^{sc}$
$\boldsymbol{u}$
$\boldsymbol{u}^{[j]}$
$\forall j \in \mathcal{N}^{sc}$
$\boldsymbol{x}_p^{[j]}$
$\forall j \in \mathcal{N}^{sc}$
Plant
$\boldsymbol{w}_p^{[i,\ell]} \forall i, \ell \in \mathcal{N}^{sm}$
$\boldsymbol{y}_p^{(i)} \forall i, \ell \in \mathcal{N}^{sm}$
$\left[\boldsymbol{q}_p^{[i,j]}\right]_{i \in \mathcal{N}^{sm}} \forall j \in \mathcal{N}^{sc}$

Figure 6: Block-oriented representation of the plant.

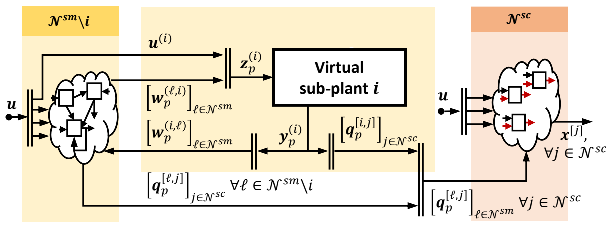

$\mathcal{N}^{sm} \backslash i$
$\mathcal{N}^{sc}$
$\boldsymbol{u}^{(i)}$
$\boldsymbol{z}_p^{(i)}$
Virtual sub-plant i
$\boldsymbol{u}$
$\left[\boldsymbol{w}_p^{(\ell,i)}\right]_{\ell \in \mathcal{N}^{sm}}$
$\left[\boldsymbol{w}_p^{(i,\ell)}\right]_{\ell \in \mathcal{N}^{sm}}$
$\boldsymbol{y}_p^{(i)}$
$\left[\boldsymbol{q}_p^{[i,j]}\right]_{j \in \mathcal{N}^{sc}}$
$\boldsymbol{u}$
$\boldsymbol{x}^{[j]}$, $\forall j \in \mathcal{N}^{sc}$
$\left[\boldsymbol{q}_p^{[\ell,j]}\right]_{j \in \mathcal{N}^{sc}} \forall \ell \in \mathcal{N}^{sm} \backslash i$
$\left[\boldsymbol{q}_p^{[\ell,j]}\right]_{\ell \in \mathcal{N}^{sm}} \forall j \in \mathcal{N}^{sc}$

Figure 7: Block-oriented representation of the plant *if the model consistency criterion holds.*

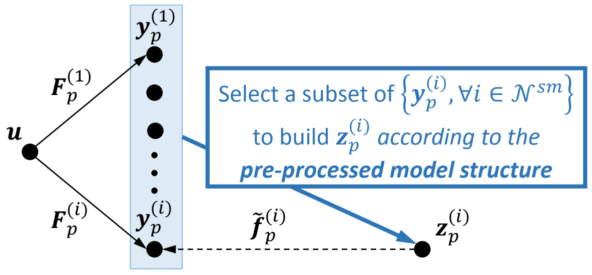
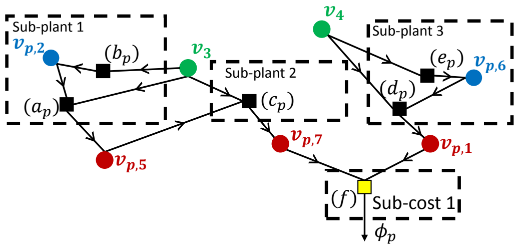
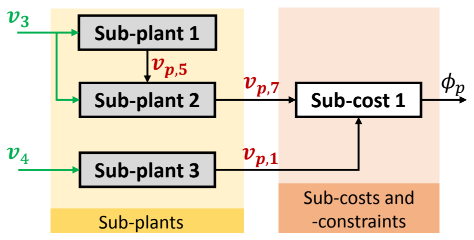

$\boldsymbol{y}_p^{(1)}$
$\boldsymbol{F}_p^{(1)}$
$\boldsymbol{u}$
Select a subset of $\left\{\boldsymbol{y}_p^{(i)}, \forall i \in \mathcal{N}^{sm}\right\}$ to build $\boldsymbol{z}_p^{(i)}$ according to the pre-processed model structure
$\boldsymbol{F}_p^{(i)}$
$\boldsymbol{y}_p^{(i)}$
$\widetilde{\boldsymbol{f}}_p^{(i)}$
$\boldsymbol{z}_p^{(i)}$

Figure 8: The mappings $\boldsymbol{F}_p^{(i)}$ between $\boldsymbol{u}$ and $\boldsymbol{y}_p^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, for the plant are based on obvious correlation relationships. However, the correlation mappings $\widetilde{\boldsymbol{f}}_p^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, do not necessarily exist on the plant since based on the pre-processed model structure.

without any corresponding variation of the modeled inputs, if, e.g. this variation is due to the unmodeled inputs.



(a) Steps 1-2 (3 with the arrows on the links).



(b) Step 3.

Figure 9: Application of the model pre-processing to the plant of Example 4.

To avoid this situation, in this article, the following model consistency assumption is performed.

Assumption 5 (Model Consistency) *The mappings $\widetilde{\boldsymbol{f}}_p^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, exist, i.e., all $\boldsymbol{z}_p^{(i)}$ and $\boldsymbol{y}_p^{(i)}$ are correlated (at steady-state).*

Finally, one also assume the uniqueness of the plant solution:

Assumption 6 (Uniqueness of the plant solution) *For any $\boldsymbol{u}_k$, there is only one $\boldsymbol{y}_p^{(i)}$ for the plant, $\forall i$, that satisfies $\boldsymbol{y}_p^{(i)} = \boldsymbol{F}_p^{(i)}(\boldsymbol{u}_k)$.*

Summary

In summary, with IMA:

- **The model** is a network of sub-models connected by *variables that are known (i.e. modeled with certainty) or measured on the plant.* Sub-model inputs are denoted $\boldsymbol{z}^{(i)}$, and contain both manipulated variables and interconnection variables. Their outputs $\boldsymbol{y}^{(i)}$ are measured on the plant and they either affect the sub-costs and -constraints, or are interconnection variables, or both. Assumptions 2 and 3 are satisfied.

- **The plant** is a mapping between the decision variables $\boldsymbol{u}$ and all the measured variables $\boldsymbol{y}_p$. Assumptions 5[*3] and 6 are satisfied.

- **The cost and constraints functions** are viewed as a network of sub-costs and -constraints that are known $\mathcal{C}^2$ functions of measured or manipulated variables, i.e., they satisfy Assumptions 1 and 4.

Remark 1 *The IMA model pre-processing procedure allows the automatic construction of the "best" possible model representation, i.e., allowing the use of all the plant measurements. As a result, the higher the number of available plant measurements, the larger the expected number of potential sub-models. Also, it must be kept in mind that the sub-models interconnection variables are not necessarily the process variables that connect the "physical" units in the plant. As illustrated with Example 1, a single-unit-plant model can be represented as two or more interconnected sub-models where* the interconnection variables are the measured variables on the plant. *In fact, the "worst" case would be when the number of plant measurements is reduced to its minimum value for MA. When it happens, IMA reduces to MAy. Generally speaking, one could expect, at the very least, to obtain sub-models corresponding to groups of physical units, i.e., to get as much sub-models as subsystems that would be obtained by applying DMA. For the latter, since the model is separated into sub-models, -costs, and -constraints but not into subsystems, one can still expect a better use of the measurements.*

[*3]As it will be seen later, two different implementations of IMA are proposed in this article and assumption 5 is indeed only required for one.

Optimization Problem

With the introduced notations summarized in Table 1, the plant optimization Problem (1) can be reformulated, without loss of generality, as the following NLP[*4]:

$$
\begin{aligned}
\boldsymbol{u}_p^\star := \arg\min_{\boldsymbol{u}} \quad & \sum_{j\in\mathcal{N}^{sc}} \phi^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}_p^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big) \\
\text{s.t.} \quad & \Big[g^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}_p^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big)\Big]_{j\in\mathcal{N}^{sc}} \leq \boldsymbol{0}, \\
& \boldsymbol{y}_p^{(i)} = \boldsymbol{F}_p^{(i)}(\boldsymbol{u}), \quad \forall i \in \mathcal{N}^{sm},
\end{aligned}
\tag{9}
$$

where $\boldsymbol{u}_p^\star$ is the plant optimum. Since the mappings $\boldsymbol{F}_p^{(i)}, \forall i \in \mathcal{N}^{sm}$, are unknown but approximated by $\boldsymbol{f}^{(i)}, \forall i \in \mathcal{N}^{sm}$, $\boldsymbol{u}_p^\star$ can be approached by solving the following model-based optimization problem:

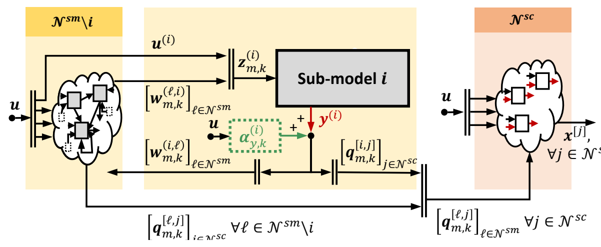
$$
\begin{aligned}
\boldsymbol{u}^\star := \arg\min_{\boldsymbol{u}} \quad & \sum_{j\in\mathcal{N}^{sc}} \phi^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big) \\
\text{s.t.} \quad & \Big[g^{[j]}\Big(\boldsymbol{u}^{[j]},\, \big[\boldsymbol{q}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big)\Big]_{j\in\mathcal{N}^{sc}} \leq \boldsymbol{0}, \\
& \boldsymbol{y}^{(i)} = \boldsymbol{f}^{(i)}(\boldsymbol{z}^{(i)}), \qquad \forall i \in \mathcal{N}^{sm}.
\end{aligned}
\tag{10}
$$

Modifier Adaptation with all available real-time plant measurements

This section is organized as follows: two new RTO methods (IMA-A and -B) that combine elements of MAy and DMA are introduced first. Because these methods are MA methods, modifiers are required and methods for their determination are presented next. Then, the IMA-A and -B algorithms are detailed and analyzed.

[*4]For the sake of space, it is not explicitly specified in (9) that $\boldsymbol{q}_p^{[i,j]}$, $\forall i \in \mathcal{N}^{sm}$, $\forall j \in \mathcal{N}^{sc}$, are composed of elements of $\boldsymbol{y}_p^{(i)}$, $\forall i \in \mathcal{N}^{sm}$. The same simplification is made for Problems (10) and (17) where $\boldsymbol{q}^{[i,j]}$ and $\boldsymbol{z}^{(i)}$ are composed of elements of $\boldsymbol{y}^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, $\forall j \in \mathcal{N}^{sc}$.

Internal Modifier Adaptation (IMA)


(a) IMA-A.

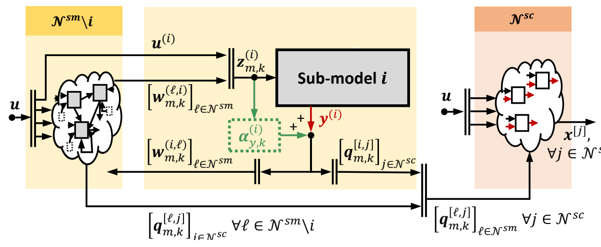

(b) IMA-B.

Figure 10: Block-oriented description of IMA-A and -B. The corrections are shown in green.

The main idea behind IMA is to simultaneously correct the measured variables affecting the cost and constraints $\boldsymbol{q}^{[i,j]}$, $\forall i \in \mathcal{N}^{sm}$, $\forall j \in \mathcal{N}^{sc}$, and the interconnection variables $\boldsymbol{w}^{(i,\ell)}$, $\forall i, \ell \in \mathcal{N}^{sm}$, by augmenting *all* the modeled and measured variables $\boldsymbol{y}^{(i)}, \forall i \in \mathcal{N}^{sm}$, with the following input-affine correction terms $\boldsymbol{\alpha}_{y,k}^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, $\forall k \in \mathbb{N}$ to iteratively modify the mapping functions $\boldsymbol{f}^{(i)}$ such that:

$$\boldsymbol{y}_{m,k}^{(i)} := \boldsymbol{f}_{m,k}^{(i)}\big(\boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}, \mathcal{A}_{y,k})\big),$$

with

$$\boldsymbol{f}_{m,k}^{(i)}\big(\boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}, \mathcal{A}_{y,k})\big) := \boldsymbol{f}^{(i)}\big(\boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}, \mathcal{A}_{y,k})\big) + \boldsymbol{\alpha}_{y,k}^{(i)}, \tag{11}$$

so that:

$$\boldsymbol{f}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k} = \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k}, \tag{12}$$

$$\nabla_u \boldsymbol{f}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k} = \nabla_u \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k}, \tag{13}$$

where $\boldsymbol{\alpha}_{y,k}^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, are either (i) affine functions of $\boldsymbol{u}$, or (ii) affine functions of $\boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}, \mathcal{A}_{y,k})$, and $\mathcal{A}_{y,k} := \{\boldsymbol{\alpha}_{y,k}^{(1)}, \ldots, \boldsymbol{\alpha}_{y,k}^{(n_{sm})}\}$. Case (i) corresponds to IMA-A, while case (ii) is IMA-B, as illustrated in Figures 10a and 10b, respectively. This distinction follows the same logics as for DMA, which has been classified into DMA-A or -B depending on whether affine corrections are in $\boldsymbol{u}$ or $\boldsymbol{z}_{m,k}^{(i)}$ (see Figure 2).

Finally, the functions $\boldsymbol{\alpha}_{y,k}^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, read:

$$\boldsymbol{\alpha}_{y,k}^{(i)} := \begin{cases} \boldsymbol{\varepsilon}_{y,k}^{(i)} + (\boldsymbol{\lambda}_{y,k}^{(i)})^{\mathsf{T}}(\boldsymbol{u} - \boldsymbol{u}_k) & \text{if IMA-A,} \\ \boldsymbol{\varepsilon}_{y,k}^{(i)} + (\boldsymbol{\lambda}_{y,k}^{(i)})^{\mathsf{T}}(\boldsymbol{z}_{m,k}^{(i)} - \boldsymbol{z}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k}) & \text{if IMA-B,} \end{cases} \tag{14}$$

where $\boldsymbol{\varepsilon}_{y,k}^{(i)}$ and $\boldsymbol{\lambda}_{y,k}^{(i)}$ are respectively the zeroth- and first-order modifiers for the sub-model i, with the following simplifications of the notations:

$$\boldsymbol{z}_{m,k}^{(i)} \triangleq \boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}, \mathcal{A}_{y,k}),$$
$$\boldsymbol{z}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k} \triangleq \boldsymbol{z}_{m,k}^{(i)}(\boldsymbol{u}_k, \mathcal{A}_{y,k}\big|_{\boldsymbol{u}_k}).$$

As depicted in Figure 10a and 10b, modifying the outputs of a sub-model i affects its output interconnection variables $\left[\boldsymbol{w}_{m,k}^{(i,\ell)}\right]_{\ell \in \mathcal{N}^{sm}}$, and, in turn, its input interconnection variables $\left[\boldsymbol{w}_{m,k}^{(\ell,i)}\right]_{\ell \in \mathcal{N}^{sm}}$. So, affine corrections on any sub-model i affect all the other sub-models, and vice-versa. Thus, all input-affine corrections $\boldsymbol{\alpha}_{y,k}^{(i)}$ must be computed simultaneously, as seen in Equation (11), where $\boldsymbol{z}_{m,k}^{(i)}$ (with subscript $(\cdot)_m$) appears in the unmodified function $\boldsymbol{f}^{(i)}$. In practice, this could require the solving of a large system of nonlinear equations, which can be computationally expensive, especially for large-scale plants. In the next subsection, efficient methods are proposed to mitigate this issue.

Modifiers computation for IMA-A and IMA-B

Theorem 1 *(Efficient method for computing the modifiers): Assume that Assumptions 1-6*[*5] *hold and that the modified model does not have multiple solutions. The set of modifiers* $\{\boldsymbol{\varepsilon}_{y,k}^{(i)}, \boldsymbol{\lambda}_{y,k}^{(i)}\}$ *enabling* (12)-(13) *is unique and satisfies the following equations:*

$$\boldsymbol{\varepsilon}_{y,k}^{(i)} = \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k} - \boldsymbol{f}^{(i)}(\boldsymbol{z}_p^{(i)})\big|_{\boldsymbol{u}_k}, \tag{15}$$

$$\boldsymbol{\lambda}_{y,k}^{(i)} = \begin{cases} \nabla_u \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k} - \nabla_{z^{(i)}} \boldsymbol{f}^{(i)}\big|_{\boldsymbol{z}_p^{(i)}|_{\boldsymbol{u}_k}} \nabla_u \boldsymbol{z}_p^{(i)}\big|_{\boldsymbol{u}_k}, & \textit{if IMA-A,} \\ \nabla_{z^{(i)}} \widetilde{\boldsymbol{f}}_p^{(i)}\big|_{\boldsymbol{z}_p^{(i)}|_{\boldsymbol{u}_k}} - \nabla_{z^{(i)}} \boldsymbol{f}^{(i)}\big|_{\boldsymbol{z}_p^{(i)}|_{\boldsymbol{u}_k}} & \\ \qquad \cdot \nabla_u \boldsymbol{z}_p^{(i)}\big|_{\boldsymbol{u}_k} \cdot \left(\nabla_u \boldsymbol{z}_p^{(i)}\big|_{\boldsymbol{u}_k}\right)^+, & \textit{if IMA-B.} \end{cases} \tag{16}$$

Proof. The proof can be found in the supporting information. □

Theorem 1 shows that the computation of the modifiers associated to a sub-model i uses only measurements of the virtual sub-plant i of the real plant. Therefore, they can be computed in a distributed manner at the level of the virtual sub-models, and, ultimately, the modified model-based optimization Problem (17), discussed in the next subsection, could be distributed, similarly to what is suggested in Milosavljevic et al. (2017)[22] for DMA-B. IMA algorithms are now summarized.

IMA Algorithms

Internal Modifier Adaptation, versions A and B (IMA-A and IMA-B)

Initialization. Provide $\boldsymbol{u}_0$. Choose $\boldsymbol{K} = K\boldsymbol{I}_{n_u}$ with $K \in (0, 1]$, and opt for IMA-A or IMA-B.

[*5]Note that Assumption 5 is only required for IMA-B. On the other hand, IMA-A remains applicable even if Assumption 5 is not met using the first line of Equation (16).

```
for $k = 0 \to \infty$

    1. Apply the inputs $\boldsymbol{u}_k$ to the plant and wait until steady state.

    2. Use all plant measurements to estimate, $\forall i \in \mathcal{N}^{sm}$:

        • $\boldsymbol{F}_p^{(i)}|_{\boldsymbol{u}_k}$, $\nabla_u \boldsymbol{F}_p^{(i)}|_{\boldsymbol{u}_k}$, and $\nabla_u \boldsymbol{z}_p^{(i)}|_{\boldsymbol{u}_k}$, for IMA-A,

        • or $\boldsymbol{F}_p^{(i)}|_{\boldsymbol{u}_k}$, $\nabla_u \widetilde{\boldsymbol{f}}_p^{(i)}|_{\boldsymbol{u}_k}$, and $\nabla_u \boldsymbol{z}_p^{(i)}|_{\boldsymbol{u}_k}$, for IMA-B,

       using, e.g., data from perturbed operating points in the neighborhood of $\boldsymbol{u}_k$.

    3. Evaluate the modifiers (15)-(16).

    4. Compute $\boldsymbol{u}_{k+1}^\star$ by solving the following modified model-based optimization problem:

       $\boldsymbol{u}_{k+1}^\star := \arg\min_{\boldsymbol{u}} \sum_{j\in\mathcal{N}^{sc}} \phi^{[j]}\Big(\boldsymbol{u}^{[j]}, \big[\boldsymbol{q}_{m,k}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big)$    (17)
           s.t. $\Big[g^{[j]}\Big(\boldsymbol{u}^{[j]}, \big[\boldsymbol{q}_{m,k}^{[i,j]}\big]_{i\in\mathcal{N}^{sm}}\Big)\Big]_{j\in\mathcal{N}^{sc}} \leq \boldsymbol{0}$,
                $\boldsymbol{y}_{m,k}^{(i)} = \boldsymbol{f}_{m,k}^{(i)}(\boldsymbol{z}_{m,k}^{(i)}), \qquad \forall i \in \mathcal{N}^{sm}$.

    5. Determine the next operating point $\boldsymbol{u}_{k+1}$ by applying a first order filter:

       $\boldsymbol{u}_{k+1} := \boldsymbol{u}_k + \boldsymbol{K}(\boldsymbol{u}_{k+1}^\star - \boldsymbol{u}_k)$.    (18)

end
```

In practice, plant measurements are corrupted with measurement noise and oscillations around steady state can be observed. For the sake of simplicity and because this article is mainly methodological, the following assumption is performed.

Assumption 7 *The output values and gradients are perfectly known for the plant at each RTO iteration*[*6]*.*

Remark 2 *As mentioned in introduction, it might be difficult in practice to estimate the plant gradients from plant measurements. Solutions exist to mitigate this issue*[5–12] *and most could be adapted to match the IMA framework. However, this is beyond the scope of this study.*

Optimality Upon Convergence

Now that the algorithms and the way they can be implemented in practice have been presented, it remains to prove that IMA-A and -B can only converge to the true plant optimum just like other MA approaches. To do so, we start first by stating a necessary condition for any RTO method to guarantee convergence to the plant optimum:

Theorem 2 (RTO convergence ⇒ KKT matching)
If a given RTO algorithm iteratively modifies the cost and constraint functions of a model such that, at $\boldsymbol{u}_k$*, for each iteration* k*, the values and gradients of the cost and constraints of the modified model match the values and gradients of the plant at* $\boldsymbol{u}_k$*, i.e., if:*

$$\phi_{m,k}\big|_{\boldsymbol{u}_k} = \phi_p\big|_{\boldsymbol{u}_k}, \qquad \boldsymbol{g}_{m,k}\big|_{\boldsymbol{u}_k} = \boldsymbol{g}_p\big|_{\boldsymbol{u}_k}, \tag{19}$$

$$\nabla_u\phi_{m,k}\big|_{\boldsymbol{u}_k} = \nabla_u\phi_p\big|_{\boldsymbol{u}_k}, \qquad \nabla_u\boldsymbol{g}_{m,k}\big|_{\boldsymbol{u}_k} = \nabla_u\boldsymbol{g}_p\big|_{\boldsymbol{u}_k}, \tag{20}$$

then, the only fixed point of this RTO algorithm is a KKT point for the plant.

Proof. This proof is straightforward and is indeed a generalization of Theorem 1 in Marchetti et al. (2009)[4], Proposition 1 in Milosavljevic et al. (2017)[22], and Theorem 2 in Papasavvas et al. (2018)[15]. □

[*6]Note that perfect gradients of the cost and constraints for the plant are also required with MA or MAy for similar methodological analyses.

Next, with the following Lemma, we state and prove that the modification of the model of IMA leads to affine corrections of the sub-cost and -constraint functions, i.e., that IMA-A and -B satisfy the conditions (19)-(20) of Theorem 2:

Lemma 1 *Consider $x^{[j]}$, $\forall j \in \mathcal{N}^{sc}$, known functions of $\boldsymbol{u}^{[j]}$ and of the modified functions $\boldsymbol{f}_{m,k}^{(i)} \in \mathcal{C}^2$, that approximate the functions $\boldsymbol{F}_p^{(i)}, \forall i \in \mathcal{N}^{sm}$. If, $\forall k \in \mathbb{N}$ and $\forall i \in \mathcal{N}^{sm}$ the equalities:*

$$\boldsymbol{f}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k} = \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k}, \qquad \forall i \in \mathcal{N}^{sm}, \tag{21}$$

$$\nabla_u \boldsymbol{f}_{m,k}^{(i)}\big|_{\boldsymbol{u}_k} = \nabla_u \boldsymbol{F}_p^{(i)}\big|_{\boldsymbol{u}_k}, \qquad \forall i \in \mathcal{N}^{sm}, \tag{22}$$

hold, then, $\forall j \in \mathcal{N}^{sc}$, the following equalities hold:

$$x^{[j]}(\boldsymbol{u}^{[j]}, \boldsymbol{f}_{m,k}^{(1)}, \dots, \boldsymbol{f}_{m,k}^{(n_{sm})})\big|_{\boldsymbol{u}_k} = x^{[j]}(\boldsymbol{u}^{[j]}, \boldsymbol{F}_p^{(1)}, \dots, \boldsymbol{F}_p^{(n_{sm})})\big|_{\boldsymbol{u}_k}, \tag{23}$$

$$\nabla_u x^{[j]}(\boldsymbol{u}^{[j]}, \boldsymbol{f}_{m,k}^{(1)}, \dots, \boldsymbol{f}_{m,k}^{(n_{sm})})\big|_{\boldsymbol{u}_k} = \nabla_u x^{[j]}(\boldsymbol{u}^{[j]}, \boldsymbol{F}_p^{(1)}, \dots, \boldsymbol{F}_p^{(n_{sm})})\big|_{\boldsymbol{u}_k}. \tag{24}$$

Proof. The proof is straightforward, and is indeed a generalization of Lemma 1 in Papasavvas et al. (2018)[15]. □

Finally, by combining Theorem 2 and Lemma 1, the following Theorem can be stated:

Theorem 3 (IMA convergence ⇒ KKT matching) *If assumptions 1-7 hold*[*7] *and if the input sequence $\{\boldsymbol{u}_k, \forall k \in \mathbb{N}\}$ generated by IMA converges to $\boldsymbol{u}_\infty := \lim\limits_{k\to\infty} \boldsymbol{u}_k$ with $\boldsymbol{u}_\infty$ a KKT point of the modified optimization Problem* (17)*, then $\boldsymbol{u}_\infty$ is also a KKT point for the plant Problem* (9)*.*

Proof. According to Theorem 1, (12) and (13) hold for the modified model at $\boldsymbol{u}_k$. Since the sub-costs and -constraints are known functions of the functions $\boldsymbol{f}_{m,k}^{(i)}$, $\forall i \in \mathcal{N}^{sm}$, which are corrected in an input-affine manner, Lemma 1 allows to state that $\{x_{m,k}^{[j]}\big|_{\boldsymbol{u}_k}, \nabla_u x_{m,k}^{[j]}\big|_{\boldsymbol{u}_k}\}$ $= \{x_p^{[j]}\big|_{\boldsymbol{u}_k}, \nabla_u x_p^{[j]}\big|_{\boldsymbol{u}_k}\}$, $\forall j \in \mathcal{N}^{sc}$. The aggregate constraints of the modified model $\boldsymbol{g}_{m,k}$ being

[*7]Note that Assumption 5 is not required for IMA-A.

the concatenation of $x_{m,k}^{[j]}$, with $j \in \mathcal{N}^{sc}$ (see Problem (9)), they are therefore also corrected in an input-affine manner, at $\boldsymbol{u}_k$. The same observation can be made for the aggregated cost of the modified model $\phi_{m,k}$. Theorem 2 can thus be used to finalize the proof that only KKT points of the plant can be fixed points for both IMA algorithms. $\square$

Remark 3 *So far, we have shown that zeroth- and first-order corrections are performed for MA (and MAy), DMA and IMA. Yet, one of the main advantages of IMA is that it also implement higher-order corrections that are inherited from MAy*[15] *and DMA*[21]*. IMA shares the same higher-order correction than MAy mainly because the outputs affecting the sub-costs and -constraints are corrected similarly. For DMA, the observed – yet not studied – higher-order corrections*[21] *are prone to stem from the attenuation of the propagation of uncertainty through the aggregated model, thanks to the correction of the interconnection variables. Because this is also the case with IMA, the same improvements are expected. For instance, when all sub-models are linear – which does not imply that the aggregated model is linear (it will typically not be linear, e.g., in the presence of recycling)–, then IMA-B corrects the model perfectly in a single RTO iteration. IMA-B being the only approach where sub-models are corrected locally, it is clear than only IMA-B will enable a one-step convergence in this case. This property is straightforward to prove (similarly to the equivalent theorem for MAy in Papasavvas et al. (2018)*[15]*), but this analysis is omitted here for the sake of space.*

Illustrative Examples

IMA-A and -B have been applied in simulation to two different case studies of growing complexity. The first sub-section deals with a steel-making process of moderate scale, which still incorporates enough units to be more challenging than a small-scale plant. Because several units are linear, this first case study will mainly highlight the advantages of IMA-B over IMA-A discussed in Remark 3, while allowing the reader to better understand the proposed approach to modeling and to the clustering of model equations. The second case study is of bigger scale and corresponds indeed to the Tennessee Eastman challenge problem.

Steel-making process

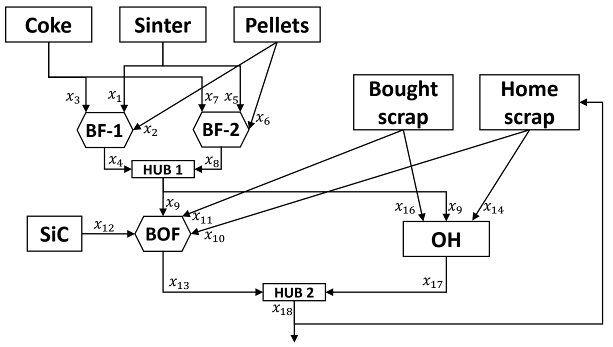


Figure 11: A unit based description of the steel plant.

We consider the steel-making plant depicted in Figure 11 taken from Ray et al. (1973)[26]. There are two blast furnaces (BF-1 and BF-2), one oxygen furnace (BOF), one open hearth shop (OH) and a storage tank for scraps. Also, in the description of the plant, two virtual hubs (Hub-1 and Hub-2) have been introduced for the distribution of material between the different equipments (see Figure 12). BF-1 and BF-2 consume sinter, pellets and coke to produce hot metal, which is distributed between the BOF and the OH through the Hub-1. The BOF uses hot metal with silicon carbide (SiC) and home and bought scrap to produce crude steel. Similarly, OH converts hot metal and scraps into crude steel. Finally, crude steel produced by both the BOF and the OH is either sold or stored via Hub-2.

The ten manipulated variables for the plant are $\boldsymbol{u} = [u_1, \ldots, u_{10}]^\mathsf{T}$, which are in order: u_1: sinter consumption in BF-1 (t/y); u_2 pellets consumption in BF-1 (t/y); u_3: sinter consumption in BF-2 (t/y); u_4: pellets consumption in BF-2 (t/y); u_5: the ratio of produced hot metal sent to BOF; u_6, bought scraps consumption in BOF (t/y); u_7: SiC consumption in BOF (t/y); u_8: bought scrap consumption (t/y); u_9: ratio of crude steel sent to the reserve (t/y); u_{10}: ratio of reserve scrap for BOF. These manipulated variables are indicated in green in Figure 12.

Twelve mass flow rates [t/y] are measured and are denoted in red in Figure 12 that are: (i to iv) the coke consumptions ($y_1^{(1)}, y_1^{(2)}$) and the hot metal production ($y_2^{(1)}, y_2^{(2)}$) of the two BF; (v) hot metal sent to the BOF ($y_1^{(3)}$), and (vi) hot metal sent to OH ($y_2^{(3)}$); (vii and viii) the crude steel produced by the BOF ($y_1^{(4)}$) and by the OH ($y_1^{(5)}$); (ix) the sold crude steel ($y_1^{(6)}$), (x) the crude steel stored in the reserve ($y_2^{(6)}$); (xi-xii) the crude steel extracted from the reserve and sent to the BOF ($y_1^{(7)}$) and to the OH ($y_2^{(7)}$).

The available model is such that plant-model mismatch is of parametric nature and plant and model parameters are listed at the bottom left of Figure 12.

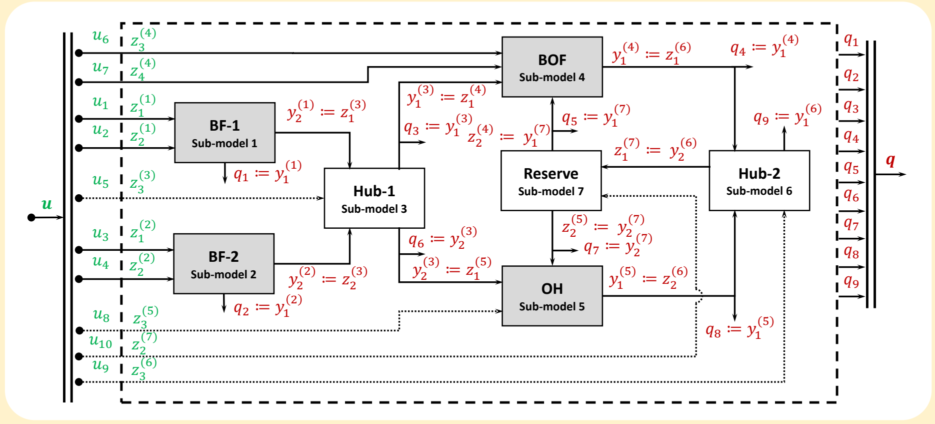
The pre-processing procedure has been applied to the available model, the corresponding results being depicted in Figure 12. The resulting network of sub-models is similar to the plant structure, with sub-models being mostly the models of the physical units, and so to the network that would be obtained with DMA. The difference with DMA lies here in the further separation between sub-models and the sub-costs and -constraints obtained with IMA.

There are four sub-costs and six sub-constraints, detailed Figure 12, which correspond to operating costs, minimum and maximum production capacities, fixed ratios between materials, and to the production target, respectively.

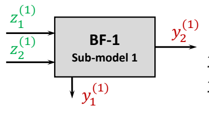
Figure 13 summarizes the simulation results for MA, IMA-A, and IMA-B. All methods have been initialized at the conservative inputs $\boldsymbol{u}_0 = [1.7,\ 0.3,\ 0.5,\ 0.2,\ 1,\ 1,\ 0.06,\ 0.1,\ 0.3,\ 0]^\mathsf{T}$. For the sake of clarity, only the aggregated cost, the five sub-constraints, u_9, and u_{10} are plotted.

As seen, the three methods converge to the true plant optimum. While it takes two iterations for MA and IMA-A to converge, IMA-B leads to convergence in a single iteration, which is a direct illustration of Remark 3. For this case study, it turns out that all sub-models are linear, and that the only nonlinearity stems from the recycling between the reserve, the BOF, and the OH, which is only related to the inputs u_9, and u_{10}. This is indeed the reason why the other inputs have not been plotted, since all methods lead to exactly the same profiles for $\{u_1, \ldots, u_8\}$.

The detailed sub-models network of the Steel-making process:

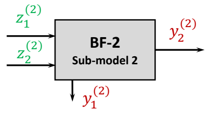


where:



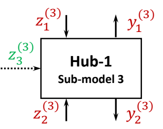
$z_1^{(1)}$: sinter flow to BF-1 [t/y]
$z_2^{(1)}$: pellets flow to BF-1 [t/y]
$y_1^{(1)}$: coke consumption [t/y]
$y_2^{(1)}$: hot metal to Hub-1 [t/y]

$$y_1^{(1)} = y_2^{(1)}(\theta_1^{(1)}(1+\theta_2^{(1)}e^{\theta_3^{(1)}(z_1^{(1)}/z_2^{(1)})}) + \theta_4^{(1)}(\theta_5^{(1)} - y_2^{(1)})^2)$$
$$y_2^{(1)} = 0.715z_1^{(1)} + 0.91z_2^{(1)}$$



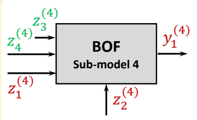
$z_1^{(2)}$: sinter flow to BF-2 [t/y]
$z_2^{(2)}$: pellets flow to BF-2 [t/y]
$y_1^{(2)}$: coke consumption [t/y]
$y_2^{(2)}$: hot metal to Hub-1 [t/y]

$$y_1^{(2)} = y_2^{(2)}(\theta_1^{(2)}(1+\theta_2^{(2)}e^{\theta_3^{(2)}(z_1^{(2)}/z_2^{(2)})}) + \theta_4^{(2)}(\theta_5^{(2)} - y_2^{(2)})^2)$$
$$y_2^{(2)} = 0.715z_1^{(2)} + 0.91z_2^{(2)}$$



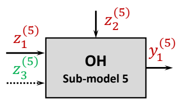
$z_1^{(3)}$: hot metal from BF-1 [t/y]
$z_2^{(3)}$: hot metal from BF-2 [t/y]
$z_3^{(3)}$: % for BOF [-]
$y_1^{(3)}$: hot metal to BOF [t/y]
$y_2^{(3)}$: hot metal to OH [t/y]

$$y_1^{(3)} = z_3^{(1)}(z_1^{(1)} + z_2^{(1)}) \qquad y_2^{(3)} = (1 - z_3^{(1)})(z_1^{(1)} + z_2^{(1)})$$



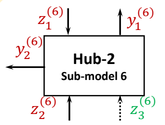
$z_1^{(4)}$: hot metal from Hub-1[t/y]
$z_2^{(4)}$: home scrap [t/y]
$z_3^{(4)}$: bought scrap [t/y]
$z_4^{(4)}$: bought SiC [t/y]
$y_1^{(4)}$: crude steel production [t/y]

$$y_1^{(4)} = \theta_1^{(4)}z_1^{(4)} + \theta_2^{(4)}(z_2^{(4)} + z_3^{(4)})$$



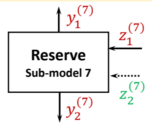
$z_1^{(5)}$: hot metal from Hub-1 [t/y]
$z_2^{(5)}$: home scrap [t/y]
$z_3^{(5)}$: bought scrap [t/y]
$y_1^{(5)}$: crude steel production [t/y]

$$y_1^{(5)} = \theta_1^{(5)}z_1^{(5)} + \theta_2^{(5)}(z_2^{(5)} + z_3^{(5)})$$



$z_1^{(6)}$: crude steel from BOF [t/y]
$z_2^{(6)}$: crude steel from OH [t/y]
$z_3^{(6)}$: recycling ratio [-]
$y_1^{(6)}$: crude steel to market [t/y]
$y_2^{(6)}$: crude steel to reserve [t/y]

$$y_1^{(6)} = (1 - z_3^{(6)})(z_1^{(6)} + z_2^{(6)}) \qquad y_2^{(6)} = z_3^{(6)}(z_1^{(6)} + z_2^{(6)})$$



$z_1^{(7)}$: crude steel from Hub-2 [t/y]
$z_2^{(7)}$: % to BOF [-]
$y_1^{(7)}$: home scrap to BOF [t/y]
$y_2^{(7)}$: home scrap to OH [t/y]

$$y_1^{(7)} = z_2^{(7)}z_1^{(7)} \qquad y_2^{(7)} = (1 - z_2^{(7)})z_1^{(7)}$$

The sub-costs and -constraints:

$$\phi^{[1]} = 21u_1 + 30u_2 + 25q_1 + 7.2 \times 10^6,$$
$$\phi^{[2]} = 21u_3 + 30u_4 + 25q_2 + 5.0 \times 10^6,$$
$$\phi^{[3]} = 15q_4 + 180u_7 + 35u_6 + 10.5 \times 10^6,$$
$$g^{[4]} = q_4 - 3.5,$$
$$g^{[5]} = 4(q_5 + u_6 - 12u_7) - q_3,$$
$$g^{[6]} = u_7 - 24q_3,$$
$$\phi^{[7]} = 26q_8 + 35u_8 + 3 \times 10^6,$$
$$g^{[8]} = (q_6 + 1.33(q_7 + u_8)) - 2,$$
$$g^{[9]} = 1.1 - q_9,$$
$$g^{[10]} = q_9 - 1.1.$$

$$u_1 \in [0, 1.7], \quad u_3 \in [0, 0.84],$$
$$u_2 \in [(0.7 - 0.715u_1)/0.91, (1.6 - 0.715u_1)/0.91],$$
$$u_4 \in [(0.4 - 0.715u_3)/0.91, (0.8 - 0.715u_3)/0.91],$$
$$u_5, u_8, u_9, u_{10} \in [0, 1],$$
$$u_6 \in [0, 4], \quad u_7 \in [0, 0.1].$$

Parameters of the plant and the model for (a) BF-1, (b) BF-2, (c) BOF, ad (d) OH:

	$\theta_1^{(1)}$	$\theta_2^{(1)}$	$\theta_3^{(1)}$	$\theta_4^{(1)}$	$\theta_5^{(1)}$
Plant	0.4	0.5	-0.71	0.3	1
Model	0.2	0.7	-0.6	0.2	1.5

(a) BF-1

	$\theta_1^{(4)}$	$\theta_2^{(4)}$
Plant	0.9	0.9
Model	0.84	0.84

(c) BOF

	$\theta_1^{(2)}$	$\theta_2^{(2)}$	$\theta_3^{(2)}$	$\theta_4^{(2)}$	$\theta_5^{(2)}$
Plant	0.5	0.4	-0.7	0.35	0.5
Model	0.4	0.5	-0.5	0.1	0.8

(b) BF-2

	$\theta_1^{(5)}$	$\theta_2^{(5)}$
Plant	0.92	0.92
Model	0.9	0.89

(d) OH

Figure 12: Sub-plants and sub-models used for the steel-making process.

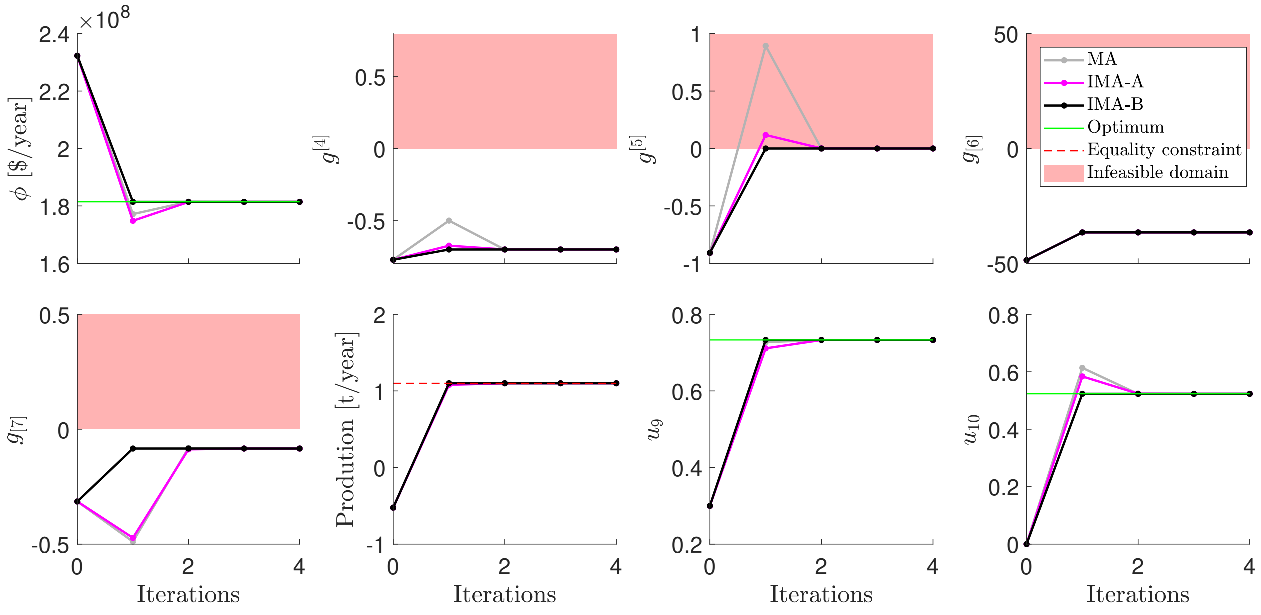


Figure 13: **Steel-making process:** Simulation results.

Although one iteration seems to be a marginal difference, it has to be kept in mind that not only one iteration means one steady-state to steady-state iteration, i.e., as much time as it takes to the slowest unit to reach steady state, but also, because plant gradients might have to be estimated using, e.g., finite differences, as many additional iterations to steady-state that are required to estimate plant gradients, here with an input vector of dimension 10.

Also, MA and IMA-A lead to a violation of the constraint $g^{[5]}$, which is the reason why the costs (top left of Figure 13) of MA and IMA-A achieve a lower value than the plant optimal cost, at the infeasible iteration. This violation being much larger with MA than with IMA-A, it is clear that IMA-A outperforms MA thanks to the additional corrections discussed by the end of the previous section. Since IMA-B prevents constraint violations, it clearly outperforms both MA and IMA-A.

The Tennessee Eastman Challenge Problem

The Tennessee Eastman plant (TE) is simulated using a benchmark FORTRAN code[27]. The plant is known to be open-loop unstable and is equipped with a decentralized control[28]

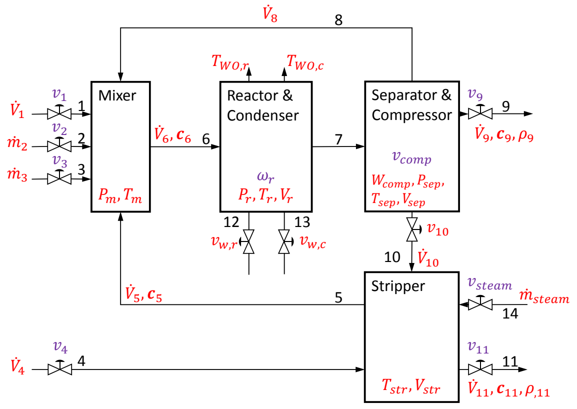


Figure 14: Unit based description of the Tennessee Eastman process. The variables manipulated by the controllers as well as the measured ones are in violet and red, respectively. More details about notations are available in the supporting information.

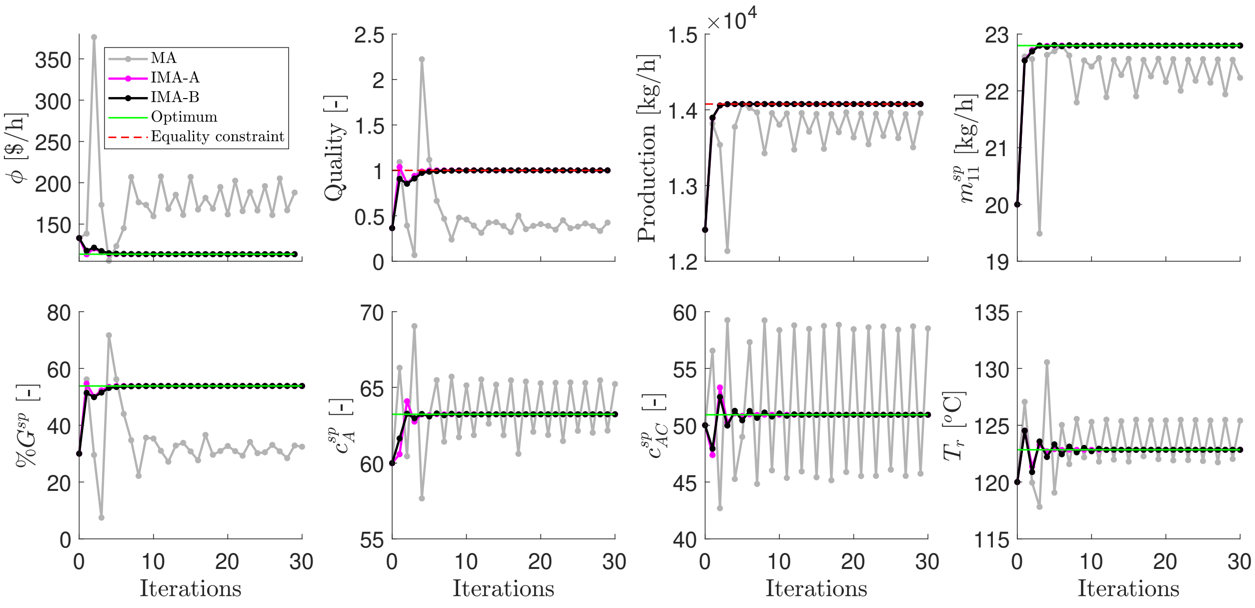


Figure 15: **TE:** Simulation results.

scheme.

The model at hand is a simplified version of a model available in the literature[29] sharing the same inputs as the plant of Bathelt et al. (2015)[27]. Nothing more than the plant information that was considered available in the Tennessee Eastman challenge problem statement[30], is considered available in this article, e.g. for modeling purposes, that is:

- Physical properties of the different components A, B, C, D, E, F, G, H, at 100 ^oC.
- An intentionnaly simplified set of reactions (as in Ricker et al. (1995)[29]) to serve as the model and to implement structural plant-model mismatch (i.e., the simulated reality of the FORTRAN code considers more reactions):

$$\begin{aligned} A_{(g)} + C_{(g)} + D_{(g)} &\rightarrow G_{(\ell)}, \\ A_{(g)} + C_{(g)} + E_{(g)} &\rightarrow H_{(\ell)}, \\ 1/3A_{(g)} + D_{(g)} + 1/3E_{(g)} &\rightarrow F_{(\ell)}. \end{aligned}$$

- The compositions and temperatures of the inlet streams to the plant (i.e., streams 1, 2, 3, and 4 in Figure 14).
- The sub-costs and -constraints of the plant,
- The measured variables. Densities ρ_9 and ρ_{11} of the flows 9 and 11 are also known[*8].

The only marginal differences with Ricker et al. (1995)[29] are in order:

- The compositions of the streams 1, 2, 3, 4 of Downs et al. (1993)[30] are used,
- The temperature T_6 is not fixed. Instead, F_6 F_9, $\dot{m}_{steam}$, P_m, T_{str}, and $c_{E,6}$ are fixed[*9],

[*8]These measurements are assumed to be available so that costs are functions of manipulated and measured variables only, for the reasons discussed in Section 3. Unlike in Ricker et al. (1995)[31], actual cost and constraints values are used. Also, the steady-state concentrations and volumetric flowrates of Stream 5, even though not directly measured, can be easily inferred from the other steady-state measurements at the mixer.

[*9]Details about these variables are available in the supporting information.

- The stripper is assumed to be ideal, i.e., it perfectly separates G and H from the other species,
- The quantity of each component in each equipment is not modeled to infer the liquid volumes V_r, V_{sep} and V_{str}. Instead, these volumes are directly manipulated through appropriate controller set-points.

The degrees of freedom for the controlled TE plant are indeed the set-points of the controllers, which are in order: $u_1 := \dot{m}_{11}^{sp}$ (mass flowrate of Stream 11 (kg/h)); $u_2 := c_{G,11}^{\%,sp}$ (molar concentration of G in Stream 11 (%)); $u_3 := c_A^{sp}$ (amount of A relative the amount of $A + C$ in stream 6 (%)); $u_4 := c_{AC}^{sp}$ (amount of $A + C$ in stream 6 (%)); $u_5 := T_r^{sp}$ (reactor temperature (^oC)) $u_6 := P_r^{sp}$ (reactor pressure (kPa)); $u_7 := V_r^{\%,sp}$ (reactor level (%)); $u_8 := V_{sep}^{\%,sp}$ (separator level (%)); $u_9 := V_{str}^{\%,sp}$ (stripper level (%)); $u_{10} := v_{steam}$ (steam valve position (%)); $u_{11} := \omega_r$ (reactor agitation speed (%)).

According to Ricker et al. (1995)[31], u_6 to u_{11} can be fixed, and so are they in this article. More precisely, the optimal reactor liquid level and pressure are kept at their lower and upper bounds, respectively. To avoid constraint violations during the transients to steady state - something that has been observed during dynamical simulations -, the values of P_r^{sp} and $V_r^{\%,sp}$ are set to 2800 kPa and 65%, slightly backed-off from their "real" limit values of 2895 kPa and 50%. Also, $V_{sep}^{\%,sp}$ and $V_{str}^{\%,sp}$ having negligible effects on the operating conditions of the plant, they are fixed to 50%, again far enough from their bounds. The energy consumed for stirring being not taken into account, the agitator speed is set to 100%. Finally, the steam valve position is set at 1%, since it has been proved[31] that steam has not effect on the steady state and costs money.

The model pre-processing procedure has been applied and the resulting networks ($\mathcal{N}^{sm}$ and $\mathcal{N}^{sc}$), with detailed equations, are given in the supporting information. Notice that, contrary to the first case study, the structure of $\mathcal{N}^{sm}$ does not mimic the structure of the plant because *not all* the process variables that connect the reactor, the condenser, the separator, the compressor, and the stripper are measured. Hence the difference between Figure 14 (which is a units-based description of the open-loop TE) and Figure S1 (in supporting information), which depicts $\mathcal{N}^{sm}$, and in turn between the clustering obtained with IMA and the unit-based description that DMA would have led to.

Only the first of the six operating modes[30] is considered. No disturbances are considered, since the purpose of this case study is to illustrate the implementation of IMA. Therefore, the model-based optimization problem considers four sub-costs and twelve sub-constraints, detailed in the supporting information. Basically, the liquid levels in the reactor, the separator and in the striper are bounded. The pressure and temperature of the reactor are upper bounded. The ratio between the flowrates of G and H in stream 11 is 50/50 to satisfy quality requirements. There is an equality constraint on the production rate, i.e., the mass flowrate of G and H in stream 11 has to be equal to 14076 kg/h. The aggregated cost is composed of (i) the cost related to the power consumption of the compressor, (ii) the steam consumed by the stripper, (iii) the flowrate in the purge, and (iv) the flowrate in the production stream 11.

Figure 15 depicts the simulation results for 30 RTO iterations with MA, IMA-A and -B starting from conservative initial inputs $\boldsymbol{u}_0 = [10, 30, 60, 50, 120]^\mathsf{T}$. The filter gain used is $K = 0.9$[*10]. One can see that MA does not converge to the plant optimum and, instead, oscillates around an infeasible point. On the other hand, IMA-A and -B lead to similar results and reach the plant optimum in about 5 iterations. Notice that the optimal steady-state reached here over performs slightly the one from Ricker et al. (1995)[31]. This is due to the fact exact cost and constraint values are considered here (see footnote *8). Results are also different from Golshan et al. (2000)[32], whereby the two step approach, i.e., online identification of the models parameters followed by re-optimization of the updated model, has been applied leading to convergence to a sub-optimal point, which happens to be quite luckily close from the true plant optimum. IMA converges to the true plant optimum, i.e. $m_{11}^{sp} = 22.797$ kg/h, $\%G^{sp} = 53.83$, $c_A^{sp} = 63.22$, $c_{AC}^{sp} = 50.92$, and $T_r = 122.84$°C for an operating cost of 113.53\$/h, while satisfying all operating constraints.

[*10]With $K = 1$, results are very similar, although more oscillations are observed before convergence for IMA-A and -B.

Remark 4 *Better results are obtained with MA when the model parameters are closer to the corresponding plant values. This seems to show that MA is more sensitive to the model parameters than IMA-A and -B.*

Conclusions

In this article, MAy and DMA have been combined and improved to maximize the use of available plant measurements, and the resulting method, IMA, has been successfully applied to two large-scale simulated case studies, among which the Tennessee Eastman process. Instead of mimicking the structure of the plant for modeling, it is argued that clustering the model into a network of sub-models on one hand, and a network of sub-costs and sub-constraints on the other, all sub-elements being connected by modeled outputs that are also measured on the plant, allows to use all plant measurements in the optimization framework. Doing so brings back the model at the core of the definition of the optimization problem and choice of the solution method, but also exploits the whole set of available plant measurements. The list of plant measurements (and not the list of units, neither of states – which do not need to be measured) is used, and, clearly, different sets of measurements would lead to different IMA-clusterings, but all would enable the use of all the corresponding plant measurements.

The properties of IMA have been analyzed and it has been proven that the plant-optimality-upon-convergence property, common to all MA methods, is preserved. Also, the problem of correcting simultaneously all sub-models, which corresponds to the solving of a potentially large system of nonlinear equations, has been treated and a practical method proposed. Despite these results, which make of IMA a promising route to the optimization of large-scale processes, many research directions and questions remain open, among which the three following:

1. Firstly, the fact that the sub-models, -costs, and -constraints are separated into the networks $\mathcal{N}^{sm}$ and $\mathcal{N}^{sc}$ enables to investigate the application of IMA to cases where

different agents want to keep some knowledge private (e.g., with distributed plant, or when RTO is outsourced to a consulting company reluctant to share its process models).

2. Secondly, research is required to compare in more details MA, MAy, DMA, and IMA, and, if possible, identify which method is best suited to which application. Also, future work should consider mixing IMA-A and -B, i.e., modifying some measured variables with affine functions of $\boldsymbol{u}$ and others with affine functions of $\boldsymbol{z}_{m,k}^{(i)}$, which could be a way to enforce the satisfaction of the Model Consistency Assumption 5, in some cases.

3. Finally, another direction for future research would be to combine IMA framework with all the extensions of standard MA enumerated in the fourth paragraph of the introduction. The authors believe that such combinations could reduce, or even remove, some assumptions – e.g., the availability of accurate estimates of the plant gradients. Also, the use of nonlinear modifying structures, still satisfying the conditions of Theorem 2, should be investigated. Indeed, as discussed in Remark S1 (in the supporting information), some of the interconnection variables may have physical bounds (e.g., a molar fraction must be within the set $[0, 1]$) that are not compatible with the affine corrections of MA, as such, leading e.g. corrected molar fractions to be outside $[0, 1]$. Hence the need to investigate the introduction of nonlinear corrections that are locally linear and meet the conditions of Theorem 2, while enforcing the satisfaction of additional physical properties on the corrected variables.

Supporting Information Available

The Supporting Information is available free of charge on the ACS Publications website at DOI: xxx

- Theorem 1 proof;

- Detailed description of the Tennessee Eastman challenge process;
- Remark S1 (An alternative to affine corrections).

This material is available free of charge via the Internet at `http://pubs.acs.org/`.

References

(1) Conn, A. R.; Scheinberg, K.; Vicente, L. N. *Introduction to Derivative-Free Optimization*; Cambridge University Press, 2009.

(2) Box, G. E. P.; Draper, N. R. *Evolutionary Operation. A Statistical Method for Process Improvement*; John Wiley, New York, 1969.

(3) Perincherry, V.; Kikuchi, S.; Hamamatsu, Y. Uncertainties in the Analysis of Large-Scale Systems. *(2nd) International Symposium on Uncertainty Modeling and Analysis* **1993**, *17 (3)*, 216–222.

(4) Marchetti, A.; Chachuat, B.; Bonvin, D. Modifier-Adaptation Methodology for Real-Time Optimization. *Ind. Eng. Chem. Res.* **2009**, *48*, 6022–6033.

(5) Gao, W.; Wenzel, S.; Engell, S. A reliable modifier-adaptation strategy for real-time optimization. *Comp. Chem. Eng.* **2016**, *91*, 318–328.

(6) de Avila Ferreira, T.; Shukla, H.; Faulwasser, T.; Jones, C.; Bonvin, D. Real-Time optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes. *European Control Conference (ECC)* **2018**, 465–470.

(7) François, G.; Bonvin, D. Use of transient measurements for the optimization of steady-state performance via modifier adaptation. *Ind. Eng. Chem. Res.* **2014**, *53*, 5148–5159.

(8) Gao, W.; Hernandez, R.; Engell, S. Real-time optimization of a novel hydroformylation process by using transient measurements in modifier adaptation. *IFAC-PapersOnLine* **2017**, *50 (1)*, 5731–5736.

(9) de Avila Ferreira, T.; Francois, G.; Marchetti, A.; Bonvin, D. Use of transient measurements for static real-time optimization. *IFAC-PapersOnLine* **2017**, *50 (1)*, 5737–5742.

(10) Rodriguez-Blanco, T.; Sarabia, D.; Pitarch, J.; de Prada, C. Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty. *Comp. Chem. Eng.* **2017**, *106*, 480–500.

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

(12) Rodriguez-Blanco, T.; Sarabia, D.; de Prada, C. Efficient nested modifier-adaptation methodology for dealing with process constraints in real-time optimization. *(in press) Comp. Chem. Eng.* **2018**,

(13) Forbes, J.; Marlin, T.; McGregor, J. Model adequacy requirements for optimizing plant operations. *Comp. Chem. Eng.* **1994**, *18*, 497–510.

(14) François, G.; Bonvin, D. Use of convex model approximations for real-time optimization via modifier adaptation. *Ind. Eng. Chem. Res.* **2013**, *52*, 11614–11625.

(15) Papasavvas, A.; Marchetti, A.; de Avila Ferreira, T.; Bonvin, D. Analysis of Output Modifier Adaptation for Real-Time Optimization. *Comp. Chem. Eng.* **2019**, *121*, 285–293.

(16) Ahmad, A.; Singhal, M.; Gao, W.; Bonvin, D.; Engell, S. Enforcing Model Adequacy in Real-Time Optimization via Dedicated Parameter Adaptation. *IFAC-PapersOnLine* **2018**, *51*, 49–54.

(17) Papasavvas, A.; François, G. Filter-based Additional Constraints to Relax the Model Adequacy Conditions in Modifier Adaptation. *(in press) IFAC-PapersOnLine* **2019**,

(18) Marchetti, A. G.; Faulwasser, T.; Bonvin, D. A feasible-side globally convergent modifier-adaptation scheme. *J. Process Contr.* **2017**, *54*, 38–46.

(19) Bunin, G. A.; François, G.; Bonvin, D. Sufficient Conditions for Feasibility and Optimality of Real-Time Optimization Schemes - I. Theoretical Foundations. *ArXiv:1308.2620* **2013**,

(20) Papasavvas, A.; François, G. Filter-based Constraints to Easier Plant Feasibility in Modifier Adaptation Schemes. *(in press) ESCAPE-29* **2019**,

(21) Schneider, R.; Milosavljevic, P.; Bonvin, D. Distributed modifier-adaptation schemes for real-time optimisation of uncertain interconnected systems. *Int. J. Control* **2017**, 1–14.

(22) Milosavljevic, P.; Schneider, R.; Faulwasser, T.; Bonvin, D. Distributed Modifier Adaptation using a Coordinator and Measured Interconnection Variables. *IFAC-PapersOnLine* **2017**, *50*, 5743–5748.

(23) Bazaraa, M. S.; Sherali, H. D.; Shetty, C. M. *Nonlinear Programming: Theory and Algorithms*, 2nd ed.; John Wiley and Sons, New York, 1993.

(24) Milosavljevic, P.; Cortinovis, A.; Marchetti, A.; Faulwasser, T.; Bonvin, D. Optimal Load Sharing of Parallel Compressors via Modifier Adaptation. *IEEE Conference on Control Applications (CCA)* **2016**, 1488–1493.

(25) Fath, B. D.; Patten, B. C. Review of the Foundations of Network Environ Analysis. *Ecosystems* **1999**, *2*, 167–179.

(26) Ray, W. H.; Szekely, J.; Ajinkya, M. B. Optimization of the ironmaking-steelmaking sequence in an integrated steel plant having non-linear and distributed elements. *Metallurgical Transactions* **1973**, *4 (6)*, 1607–1614.

(27) Bathelt, A.; Ricker, N. L.; Jelali, M. Revision of the Tennessee Eastman Process Model. *IFAC–PapersOnLine,* **2015**, *48 (8)*, 309–314.

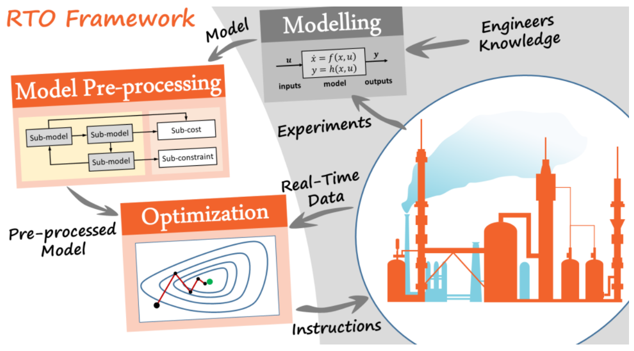
(28) Ricker, N. L. Decentralized control of the Tennessee Eastman Challenge Process. *J. Process Contr.* **1996**, *6 (4)*, 205–221.

(29) Ricker, N. L.; Lee, J. H. Nonlinear modeling and state estimation for the Tennessee Eastman challenge process. *Comp. Chem. Eng.* **1995**, *19 (9)*, 983–1005.

(30) Downs, J. J.; Vogel, E. F. A plant-wide industrial process control problem. *Comp. Chem. Eng.* **1993**, *17 (3)*, 245–255.

(31) Ricker, N. L. Optimal steady-state operation of the Tennessee Eastman challenge problem. *Comp. Chem. Eng.* **1995**, *19 (9)*, 949–959.

(32) Golshan, M.; Boozarjomehry, R.; Pisgvaie, M. A new approach to real-time optimization of the Tennessee eastman challenge problem. *Chem. Eng. J.* **2000**, *112*, 33–44.


RTO Framework
Model
Modelling
$\dot{x} = f(x,u)$
$y = h(x,u)$
u
y
inputs
model
outputs
Engineers Knowledge
Model Pre-processing
Sub-model
Sub-model
Sub-model
Sub-cost
Sub-constraint
Experiments
Real-Time Data
Optimization
Pre-processed Model
Instructions