DYNAMIC OPTIMIZATION INTEGRATING MODIFIER ADAPTATION USING TRANSIENT MEASUREMENTS

Erika Oliveira-Silva[1,2][ORCID], Cesar de Prada[1,2][ORCID], Daniel Navia[3][ORCID]

[1] Department of Systems Engineering and Automatic Control, School of Industrial Engineering, University of Valladolid, Dr. Mergelina s/n, 47011, Valladolid, Spain
[2] Institute of Sustainable Process, Dr. Mergelina s/n, 47011, Valladolid, Spain
[3] Dpto. Ingeniería Química y Ambiental, Universidad Técnica Federico Santa María, Avd. Vicuña Mackenna, Campus San Joaquín, Santiago, Chile
erika.oliveira@autom.uva.es, prada@autom.uva.es, daniel.navia@usm.cl

Abstract

In this work, a dynamic optimizer (DO) is enhanced with elements taken from the Modifier Adaptation (MA) methodology for Real-Time Optimization (RTO). The objective is to use the capability of MA methodologies to reach the real optimum of a process despite the existence of process-model mismatch. The modifiers are computed during a process transient, without waiting for several steady states to occur, speeding up the time required to reach the plant optimum. The architecture proposed includes a modified dynamic optimization, a module for estimating the model states and an additional module for direct estimation of the modifiers during transient. The proposed approach was applied in a well-known benchmark example: the Williams-Otto reactor, where the model and process have significant parametric and structural mismatch. The results show that the proposed integration is capable of bringing the process closer to the real optimal economic operation point.

Keywords: Real-time optimization; Modifier Adaptation; Uncertainty; Dynamic optimization; Process optimization; Transient measurements

1 INTRODUCTION

Generally speaking, the objective of process optimization is to take the right operation decisions that minimize production costs and maximize profits, fulfilling safety, environmental and quality constraints. Normally, decision-making inside the industrial sector is organized as in Figure 1. The top of the pyramid consists in the Enterprise Resource Planning (ERP) and it is responsible for production planning over a long time (months). The next level corresponds to the Manufacturing Execution Systems (MES) responsible for the operation planning and scheduling, which deal with assignments of products and tasks to corresponding equipment and timing over a period of days or hours. Inside the MES level, the Real Time Optimization (RTO) layer is located. RTO is a set of algorithms and techniques that automatically calculate, using real-time data, the optimum operating point of a process, considering economic criteria. Commonly, RTO uses a nonlinear stationary model with online process data to calculate the optimum value of key controller set points in steady state. These set points values are sent to a Model Predictive Control (MPC) layer. The final level corresponds to basic control, actuators, and sensors in the field.

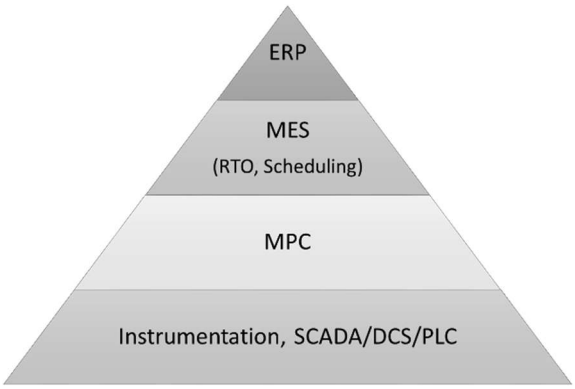


Figure 1: Decision layers for control and operation in industry (Galan et al., 2019).

In this work, special attention is given to the RTO and MPC layers. Ideally, RTO should provide the best operating point to solve the problem (1).

$$\begin{aligned} &\min_{\boldsymbol{u}} \phi_p(\boldsymbol{u}, \boldsymbol{y}_p) \\ s.t\ \ &\boldsymbol{g}_p(\boldsymbol{u}, \boldsymbol{y}_p) \leq 0 \\ &\boldsymbol{f}_{\boldsymbol{p}}(\boldsymbol{u}, \boldsymbol{y}_p) = 0 \\ &\boldsymbol{u}^{\boldsymbol{L}} \leq \boldsymbol{u} \leq \boldsymbol{u}^{\boldsymbol{U}} \end{aligned} \tag{1}$$

Where ϕ_p is the process cost function to be minimized, $\boldsymbol{u}$ are the decision variables between the lower and upper limits $[\boldsymbol{u}^L, \boldsymbol{u}^U]$, $\boldsymbol{y}_p$ are measured output variables, $\boldsymbol{g}_P$ a set of inequality constraints and $\boldsymbol{f}_P$ the "real" process model. In this paper, the subscript p, will indicate that the variable or function correspond to the correct one of the process.

As the steady-state input-output function $\boldsymbol{y}_p(\boldsymbol{u})$ of the real process is generally unknown, an approximate steady-state nonlinear model is used to predict the output $\boldsymbol{y}(\boldsymbol{u})$ and, using this relation to replace $\boldsymbol{y}$, the RTO problem is formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{u}} \phi(\boldsymbol{u}) \\ s.t\ \ &\boldsymbol{g}(\boldsymbol{u}) \leq \boldsymbol{0} \\ &\boldsymbol{u}^{\boldsymbol{L}} \leq \boldsymbol{u} \leq \boldsymbol{u}^{\boldsymbol{U}} \end{aligned} \tag{2}$$

The solution of problem (2) generally differs from the real process (1) due to parametric or structural plant-model mismatch, and disturbances. One intuitive strategy to remedy this problem is to use process data to update the parameters of the nonlinear steady-state model, using data reconciliation or estimation algorithms. This method is called a two-step approach and it is probably the only commercial RTO system used in the industry nowadays (Câmara et al., 2016).

However, the two-step approach does not solve the problem of finding the process optimum when there is structural process-model mismatch (Yip and Marlin, 2004). To improve this solution, further developments incorporated process gradient information in the calculation, in order to match the necessary conditions of optimality (NCO) of the plant and model. The Integrated System Optimization and Parameter Estimation (ISOPE) (Brdyś and Tatjewski, 1994) method uses a gradient term in the optimization problem to converge to a point that satisfies the NCO. Afterwards, it was observed that the parameter estimation step of ISOPE was unnecessary to satisfy the NCO conditions of the plant and modifications in the constraints must

also be considered. Later on, Gao and Engell (Gao and Engell, 2005) defined new modifiers for the constraints and finally, Marchetti (Marchetti et al., 2009) formalized the Modifier Adaptation methodology (MA). In the basic MA (Marchetti et al., 2009), at the current steady state k, additional terms involving past inputs $\boldsymbol{u}_{k-1}^{*}$ and modifiers $\boldsymbol{\lambda}$, $\boldsymbol{\gamma}$, $\boldsymbol{\varepsilon}$ are added to the original optimization problem in (2), formulating the modified problem as problem (3).

$$\begin{aligned} \min_{\boldsymbol{u}} \phi_M &:= \phi(\boldsymbol{u}) + \boldsymbol{\lambda}_k^T(\boldsymbol{u} - \boldsymbol{u}_{k-1}^{*}) \\ s.t \quad \boldsymbol{g}_M(\boldsymbol{u}) &:= \boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{\gamma}_{k,i}^T(\boldsymbol{u} - \boldsymbol{u}_{k-1}^{*}) + \boldsymbol{\varepsilon}_k \leq 0 \\ \boldsymbol{u}^{\boldsymbol{L}} &\leq \boldsymbol{u} \leq \boldsymbol{u}^{\boldsymbol{U}} \end{aligned} \tag{3}$$

The variables ϕ_M and $\boldsymbol{g}_M$ are the modified cost function and constraints respectively, $\boldsymbol{u}_{k-1}^{*}$ are the optimum values of the manipulated variables calculated and applied to the process in the previous steady state. The modifiers are given by equation (4).

$$\boldsymbol{\lambda}_k^T = \left.\frac{\partial \phi_p}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^{*}} - \left.\frac{\partial \phi}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^{*}} \tag{4.a}$$

$$\boldsymbol{\gamma}_k^T = \left.\frac{\partial \boldsymbol{g}_p}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^{*}} - \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{u}}\right|_{\boldsymbol{u}_{k-1}^{*}} \tag{4.b}$$

$$\boldsymbol{\varepsilon}_k = \boldsymbol{g}_p(\boldsymbol{u}_{k-1}^{*}) - \boldsymbol{g}(\boldsymbol{u}_{k-1}^{*}) \tag{4.c}$$

Problem (3) is solved iteratively calculating the values $\boldsymbol{u}_k^{*}$, applying to the process the decision variables suggested in problem (3), waiting until the next steady state, measuring the process outputs, and updating modifiers.

To find $\boldsymbol{u}_k^{*}$, the model used in the RTO layer must be adequate, that is, it must follow the adequacy criterion addressed by (Forbes and Marlin, 1996) that says that the model must have a local minimum in the same point of plant optimum, $\boldsymbol{u}_P{}^{*}$. Upon convergence, MA methodology already ensures the match between the model's first order Karush-Kuhn-Tucker (KKT) conditions and the process conditions. For the second order conditions to also be met, the reduced Hessian of the Lagrangian of the problem (3) must be positive definite in $\boldsymbol{u}_P{}^{*}$ (Marchetti et al., 2016).

As can be seen in equation (4.a) and (4.b), MA methodology needs the values of the gradients of the real plant and model to calculate the modifiers. There are different methods of accurately estimating model gradients; however, the plant gradients are more difficult to estimate. Variants of the MA methodology differ mainly in the methods for estimating the plant gradients. Others alternatives appeared as in the Nested Modifier Adaptation (NMA) approach (Navia et al., 2015) where modifiers are directly updated using an additional derivative-free optimization algorithm, such as the Nelder-Mead algorithm. In NMA the gradient estimation step is replaced by an optimization layer that aims to optimize a performance function measured from the real process, using the modifiers as decision variables. Notice that all the previous methods still need process information in steady state, which could be a problem when the process have a large settling time.

Even though MA is a powerful methodology that could guarantee the real optimum of the process despite the use of incorrect models, the industrial application of MA is almost nonexistent (Marchetti et al., 2016). One reason can be associate to the mismatch between the RTO+MA and MPC layers when the setpoint calculated at the upper layer is unreachable by the controller layer, leading to a poor economic performance. Some recent contributions have presented solutions for that. Vaccari and Pannocchia (Vaccari and Pannocchia, 2016) used the so-called economic MPC (eMPC) where the optimization problem has an economic target. Their

contribution includes the modifiers from the MA methodology to achieve the NCO of the process despite plant-model mismatch. So, the RTO layer inside MES in the decision-making hierarchy (Figure 1) is formulated within the MPC with additional modifier terms as in MA, and the results are used as targets in the control layer. The implementation has two layers: the target optimization layer and the regulator optimization layer. The eMPC is applied in an example of a continuous stirred tank reactor that uses a state space model in the controller with a parameter mismatch. But the gradient of the process in steady state for MA modifiers calculation is considered known. Afterwards, (Pannocchia, 2018; Vaccari and Pannocchia, 2018) expanded the previous work estimating the process gradients using online data and identification algorithms. Identification algorithms require a sufficient excitation for a correctly estimation of process gradients. So, during a period of time (identification horizon) a random signal is added to the input to excite the system, increasing the convergence time of the algorithm.

Hernández and Engell (2019) also have contributed with a new integration of RTO+MA with an eMPC with two layers. The formulation corrects the nominal dynamic plant model instead of the objective function and predict plant gradients from transient data. The gradient estimation uses a linear dynamic model and identification algorithms to approximate the measurements of the true plant.

Faulwasser and Pannocchia (Faulwasser and Pannocchia, 2019) have used the output modifier adaptation and the offset-free economic MPC with an augmented state estimator of previous works and present a formulation of the optimal control problem without the terminal state constraint. The authors confirm that the correct estimation of these gradients is very important to an eMPC+MA framework. In this work, they assume the plant gradients are available.

Recently, Vaccari and coworkers have presented a two-layer eMPC+MA framework using a technique for direct estimation of the modifiers using steady-state perturbations and a Broyden update algorithm (Vaccari et al., 2020). In the same work the authors use transient data to significantly reduce the convergence time, but the estimation is inaccurate.

In Table 1 an overview of the eMPC+MA contributions mentioned before is presented. The table shows how the plant gradient was calculated in each study, the ratio between the convergence time using correct gradients and estimated ones, and the number of layers inside the controller. The contributions using a process gradient estimation spend approximately 5-8 times more to converge in comparison to the results considering gradients known. These results show that the correct estimation of plant gradients still is a problem as the traditional/static MA requires a large number of time-consuming iterations to estimate correctly the steady-state process gradients of the cost function and constraints (Câmara et al., 2016). This may be a serious drawback from an application point of view as, not only real industrial processes are constantly under disturbances that makes difficult reaching steady state but also because many processes naturally have a large transient period covering several hours. This may make the correct plant gradient estimation impractical as, when a solution is computed, the process operation conditions may have changed.

Some authors have presented alternatives to improve the performance of MA in real processes, estimating the process gradients using transient data. François & Bonvin (François and Bonvin, 2014) presented an estimation of the gradients using Neighboring Extremals (NE). Rodriguez-Blanco et al. (Rodríguez-Blanco et al., 2017) also developed a method that uses a truncated Taylor expansion of the process cost combined with identification algorithms to estimate process gradients using transient data.

Table 1: Overview of eMPC+MA studies.

Problems	Reference	Process Gradient estimation	Ratio between convergence time with estimated gradients and perfect gradients [1]	Number of layers inside the controller
Isothermal Continuous- Stirred Tank Reactor (CSTR) A→ B → C	(Vaccari and Pannocchia, 2016)	Assuming known	-	2
Isothermal Continuous- Stirred Tank Reactor (CSTR) A→ B → C	(Pannocchia, 2018)	• Assuming known • Linear system identification	8	2
Isothermal Continuous- Stirred Tank Reactor (CSTR) A→ B → C	(Vaccari and Pannocchia, 2018)	• Assuming known • System identification algorithm (N4SID method)	5	2
Isothermal Continuous- Stirred Tank Reactor (CSTR) A→ B → C	Hernández and Engell (2019)	Linear system identification	-	2
Isothermal Continuous- Stirred Tank Reactor (CSTR) A→ B → C and Williams Otto Reactor	(Faulwasser and Pannocchia, 2019)	Assuming known	-	1
Williams Otto Reactor	(Vaccari et al., 2020)	• Assuming known • Modifiers directly estimated using Broyden's approximation	6	2

(1) The ratio was approximately calculated from the results presented by the authors in the studies. The reference is only indicative as the examples are different and depend on the algorithm parameters and the problem. The ratio calculated only aims to evaluate the orders of magnitude of the algorithms currently available in the literature.

The present work also integrates RTO, MPC and MA with a single control and optimization layer, presenting a new approach for directly estimating the MA modifiers with transient measurements such that they will coincide with the real ones when the steady state is reached. With this paper we wish to contribute to the advance of this field, which has important practical applications. The new modifier estimation technique instead of trying to estimate the process gradients, directly estimates the modifiers and it is called Dynamic Modifier Estimation (DME). DME considers the MA modifiers as terms that try to approximate the modified model to the real process using past process transient data. The goal behind the development is to find a formulation able to speed up the convergence to the plant optimum in order to apply this new strategy in slow dynamics processes that requires hours/days to achieve steady state, which makes the application of any traditional MA impractical. The proposed approach is tested in a well-known process, the Williams-Otto reactor with a process model that has significant parametric and structural mismatch.

The paper is organized as follows: after the introduction, Section 2 explains the architecture of the system focusing on the dynamic optimization with MA and the state estimation using Moving Horizon Estimator (MHE). Then, Section 3 describes how DME directly estimates the modifiers on-line, using transient measurements. Section 4 presents the proposed algorithm. Section 5 shows an application of the methodology in the Williams Otto Reactor problem. Finally, Section 6 presents the conclusions and future work.

2 Unification of RTO, MPC and MA architecture

The architecture of the proposed integrated framework is depicted in Figure 2 and it is composed of three modules: a module integrating RTO, MPC and MA that performs the core task, called Dynamic Optimizer with MA (DOMA), a Moving Horizon Estimator (MHE) for estimating states and disturbances, and a third one, denoted as Dynamic Modifier Estimation (DME), for computing values of the modifiers used in DOMA. This scheme is executed at regular time intervals denoted by the subindex k.

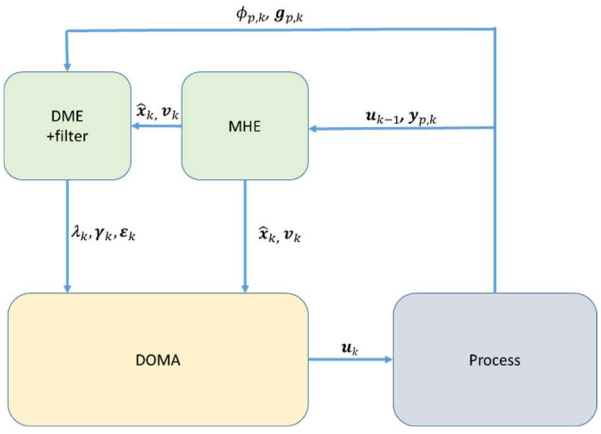


Figure 2: Architecture of DOMA with DME.

At each iteration k, that coincides with the sampling time of all modules, values of manipulated variables applied to the process in the past interval $\boldsymbol{u}_{k-1}$ and current process measurements $\boldsymbol{y}_k$ are collected and sent to the MHE module to estimate the current model states $\hat{\boldsymbol{x}}_k$ and disturbances $\boldsymbol{v}_k$. The solution given by the MHE module ($\hat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k^*$), is used by the DME module

with current values of the process economic cost function $\phi_{p,k}$ and constraints $\boldsymbol{g}_{p,k}$, to estimate on-line the values of the cost and constraints MA modifiers $\boldsymbol{\lambda}_k$, $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$, after filtering. In this work, the MA modifiers are not calculated using the gradients as in (4), but they are directly computed using the new algorithm proposed in this paper, the DME module. After that, the modifiers are used in the DOMA to compute new control actions over a certain horizon. Then, the first control action of the horizon, corresponding to current time $\boldsymbol{u}_k$, is applied to the process at instant k. Finally, in the next sample time, the whole procedure is repeated, following a moving horizon policy, as in MPC. All the three modules presented are described in the following sections.

2.1 Dynamic Optimizer with MA (DOMA)

This module incorporates an economic target and the MA methodology in a dynamic optimization problem, searching for the control moves $\Delta\boldsymbol{u}_{k+i}$, $i = 0,1,2,\dots,n_u-1$, that minimize the cost function (5) subjected to constraints (6). Note that the problem is formulated in the continuous domain of time (t).

The process is represented by a continuous dynamic model (6.a), (6.b) assumed continuously differentiable. In problem (5), (6), $\boldsymbol{x} \in \mathbb{R}^{n_x}$ represent the states, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the control actions, $\boldsymbol{y} \in \mathbb{R}^{n_y}$ the measured outputs, and $\boldsymbol{v}_k \in \mathbb{R}^{n_v}$ the disturbances estimated by the MHE module. As the model is formulated in continuous time, $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{y}$ are functions of t, but for simplicity this dependence has been omitted throughout the document unless necessary (e.g. 6.f). Using a control vector parametrization, the control actions are only allowed to change at regular time intervals $\Delta t = t_k - t_{k-1}$. Denoting k the current sampling time, the control actions $\boldsymbol{u}_k \in \mathbb{R}^{n_u}$, computed and applied at time t_k, are kept constant within each time interval $[t_k, t_{k+1}]$, as in (6.f). The current and future control moves, denoted as $\Delta\boldsymbol{u}_{k+i}$, $i = 0,1,2,\dots,n_u-1$ defined in (6.e), are the decision variables of problem (5), (6). The selection of the control horizon n_u and other tuning parameters follows the usual rules of MPC (Shah and Engell, 2011).

The model allows to compute predictions of the cost function and constraints over a future horizon from t_k to the final prediction horizon t_{pred}, long enough so that the model can reach steady state. n_{pred} refers to the number of time instants to reach t_{pred} from t_k.

The control moves are computed every sampling time from current time t_k until a control horizon t_{k+n_u} after which $\Delta\boldsymbol{u}_{k+i} = 0$, but only the first control move $\Delta\boldsymbol{u}_k$ is applied to the process.

$$\min_{\substack{\Delta\boldsymbol{u}_{k+i} \\ i=0,1,\dots n_u-1}} \phi(t_{pred}) + \boldsymbol{\lambda}_k^{\boldsymbol{T}}(\bar{\boldsymbol{u}} - \boldsymbol{u}_{k-1}) + \sum_{i=0}^{n_u-1} \Delta\boldsymbol{u}_{k+i}^T \boldsymbol{Q}_u \Delta\boldsymbol{u}_{k+i} \tag{5}$$

$$s.t. \quad \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \boldsymbol{0}, \ \forall t \in [t_k, t_{pred}] \tag{6.a}$$

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \boldsymbol{0}, \forall t \in [t_k, t_{pred}] \tag{6.b}$$

$$\boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{\gamma}_k^T(\bar{\boldsymbol{u}} - \boldsymbol{u}_{k-1}) + \boldsymbol{\varepsilon}_k \leq \boldsymbol{0}, \ \forall t \in [t_k, t_{pred}] \tag{6.c}$$

$$\boldsymbol{u}^{\boldsymbol{L}} \leq \boldsymbol{u}_{\boldsymbol{k+i}} \leq \boldsymbol{u}^{\boldsymbol{U}}, \qquad i = 0,1,\dots n_{pred}-1 \tag{6.d}$$

$$\boldsymbol{u}_{\boldsymbol{k+i}} = \boldsymbol{u}_{\boldsymbol{k+i-1}} + \Delta\boldsymbol{u}_{\boldsymbol{k+i}}, \qquad i = 0,1,\dots n_{pred}-1 \tag{6.e}$$

$$\boldsymbol{u}(t) = \boldsymbol{u}_{\boldsymbol{k+i}}, \qquad t \in [t_{k+i}, t_{k+i+1}], \qquad i = 0,1,\dots n_{pred}-1 \tag{6.f}$$

$$\Delta\boldsymbol{u}_{\boldsymbol{k+i}} = 0, \ \ i = n_u,\dots n_{pred}-1 \tag{6.g}$$

$$\boldsymbol{x}(t_k) = \hat{\boldsymbol{x}}_k \tag{6.h}$$

The cost function (5) of the problem is composed of three terms:

1. As one wishes to reach and maintain the process operating steadily at the real process optimum, the first term, $\phi(t_{pred})$, corresponds to an economic target computed with the value of model variables and control actions at the end of the prediction horizon t_{pred}, where the variables are supposed to reach steady state. Examples of economic targets can be maximizing benefits or production or minimizing costs or energy. We assume that the value of $\phi_{p,k}$ at any time instant k can be computed from process measurements and control actions. $\phi(t_{pred})$ depends on the expected dynamic of the system, control moves and disturbances.
2. The second one, $\boldsymbol{\lambda}_k^T(\overline{\boldsymbol{u}} - \boldsymbol{u}_{k-1})$, is a MA type additional term responsible of modifying the economic cost function ϕ to match the NCO of the real plant in steady state. Here $\overline{\boldsymbol{u}} = \boldsymbol{u}_{k-1} + \sum_{i=0}^{n_u-1} \Delta \boldsymbol{u}_{k+i}$ is the final value of the control actions. Note that this term is equivalent to the one added to the cost function in the traditional MA formulation (3). The modifier $\boldsymbol{\lambda}_k$ is computed every sampling time k by the external module DME and kept constant in problem (5), (6) as $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$.
3. Finally, the third term, $\sum_{i=0}^{n_u-1} \Delta \boldsymbol{u}_{k+i}^T \boldsymbol{Q}_u \Delta \boldsymbol{u}_{k+i}$ penalizes changes in the manipulated variables increasing stability and contributes to model adequacy and convexification (François and Bonvin, 2013). $\boldsymbol{Q}_u$ is a positive definite matrix, with weighting factors on the control moves $\Delta \boldsymbol{u}$ that can be considered tuning factors for normalization and stabilization as in current practice of MPC.

Finally, the problem is subjected to model (6.a), (6.b) and to inequality constraints (6.c), (6.d). In (6.c), $\boldsymbol{g}(\boldsymbol{u})$ are computed using (6.a) and (6.b), the constraint modifiers $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k$ are calculated in the DME module and they are kept constant at every iteration k.

To solve problem (5), (6), at every sampling time k the initial value of the model states $\boldsymbol{x}(t_k)$ and the disturbances are initialized at the values $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k$ respectively, which have been estimated by the MHE module.

The DOMA presented in (5), (6) has been made considering the formulation (3) of MA, but other alternative and equivalent formulations are possible if the MAy (Papasavvas et al., 2019) approach is selected. In this case, the modifiers are used in the predicted output directly, instead of adding an extra term to the cost function. In the same way, the dynamic optimization problem could had been formulated as a two-step one, as in (Vaccari et al., 2020), where a static optimization to compute optimal steady-state targets is performed first and then, the results are used as terminal constraints in a horizon optimal control problem (FHOCP). Nevertheless, in (Vaccari et al., 2020) formulation, it may happen that the target values computed by target optimization problem are not reachable by the FHOCP, because path constraints are not considered when computing the optimal targets. This problem can be mitigated using the proposed formulation (5), (6), where the search only takes place within the feasible region.

2.2 MHE estimation

To provide offset free behavior to DOMA problem (5), (6), as well as computing the initial values of states and disturbances, an augmented state estimator can be employed. Extended Kalman filter (EKF) or Moving Horizon Estimation (MHE) are two well-known methods for this task, and any of them could be used in our scheme. Nevertheless, MHE was chosen because it fits better in the optimization framework, allows estimations of specific disturbances affecting the non-

linear model and, if necessary, includes constraints in the formulation (Huang et al., 2010; Rawlings et al., 2019; Vaccari and Pannocchia, 2018). The MHE uses past information of measurements and control actions to estimate values of the states and disturbances $\boldsymbol{v}$, solving the dynamic optimization problem (7), (8).

$$\min_{\substack{x_{k-n} \; v_{k-i} \\ i=1,\dots,n_e}} \sum_{i=0}^{n_e-1} \Delta \boldsymbol{y}_{k-i}^T \boldsymbol{Q}_y \Delta \boldsymbol{y}_{k-i} + \Delta \boldsymbol{x}_{k-n_e}^T \boldsymbol{Q}_x \Delta \boldsymbol{x}_{k-n_e} + \sum_{i=1}^{n_e} \boldsymbol{v}_{k-i}^T \boldsymbol{Q}_v \boldsymbol{v}_{k-i} \tag{7}$$

$$s.t. \qquad \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{0}, \forall t \in \left[t_{k-n_e}, t_k\right], \; \boldsymbol{x}\left(t_{k-n_e}\right) = \boldsymbol{x}_{k-n_e} \tag{8.a}$$

$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}) = \boldsymbol{0}, \; \forall t \in \left[t_{k-n_e}, t_k\right] \tag{8.b}$$

$$\boldsymbol{g}(\boldsymbol{u}, \boldsymbol{y}) \leq \boldsymbol{0}, \; \forall t \in \left[t_{k-n_e}, t_k\right] \tag{8.c}$$

$$\boldsymbol{u}(\boldsymbol{t}) = \boldsymbol{u_{k-i}}, \qquad \boldsymbol{v}(\boldsymbol{t}) = \boldsymbol{v}_{k-i}, \qquad t \in [t_{k-i}, t_{k-i+1}], \qquad i = 1, \dots, n_e \tag{8.d}$$

$$\boldsymbol{v^L} \leq \boldsymbol{v}_{k-i} \leq \boldsymbol{v^U}, \qquad i = 1 \dots n_e \tag{8.e}$$

$$\Delta \boldsymbol{y}_{k-i} = \boldsymbol{y}_{k-i} - \boldsymbol{y}_{P,k-i}, \qquad i = 0, \dots, n_e - 1 \tag{8.f}$$

$$\Delta \boldsymbol{x}_{k-n_e} = \boldsymbol{x}_{k-n_e} - \widehat{\boldsymbol{x}}_{k-n_e} \tag{8.g}$$

Problem (7), (8), has also been formulated in continuous time, and it is executed at every sampling time k, using the same non-linear model (8.a), (8.b) of the DOMA problem, considering a past horizon $t \in \left[t_{k-n_e}, t_k\right]$. In this past horizon, the control variables $\boldsymbol{u}_{k-i}$ applied to the process in $[t_{k-i}, t_{k-i+1}]$, and the process measurements collected in t_{k-i}, i.e., $\boldsymbol{y}_{P,k-i}$, are known. The MHE problem is illustrated with reference to Figure 3.

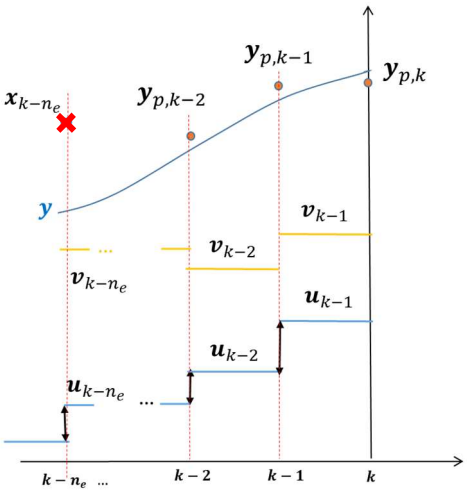


Figure 3: Past values for the MHE estimation.

Note that the decision variables of the MHE problem are the values of the states at time t_{k-n_e} ($\boldsymbol{x}_{k-n_e}$) represented with a red "x" in Figure 3, and the past disturbances $\boldsymbol{v}_{k-i}, i = 1, \dots, n_e$, denoted with yellow lines in Figure 3. The MHE assumes that, if $\boldsymbol{u}_{k-i}$ were applied to the process and $\boldsymbol{v}_{k-i}$ were applied to the model (8.a) and (8.b) starting from $\boldsymbol{x}_{k-n_e}$, then, the corresponding model output at sampling times $k-i, \; i = 0, \dots, n_e - 1$, i.e., $\boldsymbol{y}_{k-i}$, must be as close as possible to $\boldsymbol{y}_{P,k-i}$. This aim corresponds to the second term in the cost function (7), which also incorporates two additional terms: the last one minimizes the magnitude of the estimated disturbances $\boldsymbol{v}_{k-i}$, while the first one is a prior weighting penalizing the distance of the decision

variable $\boldsymbol{x}_{k-n_e}$, with respect to the one that was previously estimated at sampling time $k-n_e$ ($\widehat{\boldsymbol{x}}_{k-n_e}$), as in (8.g). $\boldsymbol{Q_x}$, $\boldsymbol{Q_v}$ and $\boldsymbol{Q_y}$ are positive definite matrices, with weighting and normalization factors. The problem also includes inequality constraints to bound the disturbances in the allowed range (8.e), as well as others (8.c) to avoid non-desirable values of the variables.

The solution of problem (7), (8) gives $\boldsymbol{x}^*_{k-n_e}$ and $\boldsymbol{v}^*_{k-i}, i = 1 \dots n_e$. This implies that once problem (7), (8) has been solved, its solution can be used to estimate the initial value of the model state at time t_k, i.e., $\widehat{\boldsymbol{x}}_k$, that is required by the DOMA problem in (6.h). To obtain $\widehat{\boldsymbol{x}}_k$ the model equations (9) must be integrated over $t \in [t_{k-n_e}, t_k]$ starting from $\boldsymbol{x}^*_{k-n_e}$, using the estimated disturbances $\boldsymbol{v}^*_{k-i}, i = 1, \dots, n_e$, and applying $\boldsymbol{u}_{k-i}$.

$$\boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}^*) = 0\ ,\ \forall t \in [t_{k-n_e}, t_k]\ ,\ \boldsymbol{x}(t_{k-n_e}) = \boldsymbol{x}^*_{k-n_e} \tag{9}$$

Once (9) has been solved, the states calculated at time t_k are defined as the initial value for problem (5), (6) as it is stated in (10).

$$\widehat{\boldsymbol{x}}_k = \boldsymbol{x}(t_k) \tag{10}$$

The value of the disturbances $\boldsymbol{v}_k$ required in problem (5), (6), is taken as the estimation given by the MHE at the most recent sampling period $\boldsymbol{v}^*_{k-1}$, this is $\boldsymbol{v}_k = \boldsymbol{v}^*_{k-1}$.

3 ONLINE MODIFIERS ESTIMATION (DME)

The third module of Figure *2*, denoted DME, has the task of computing the modifiers $\boldsymbol{\lambda}_k, \boldsymbol{\gamma}_k$ used in the DOMA module, while $\boldsymbol{\varepsilon}_k$ can be directly computed from (4.c). From their definitions in (4.a) and (4.b), several approaches have been proposed in the literature to compute modifiers based on the estimation of the process gradients from available measurements. As mentioned before, the use of transient measurements to estimate modifiers opens the door to the application of MA in slow dynamic processes. With that in mind, the MA modifiers could be seen as correction terms added to the economic optimization problem, so that the modified problem solution matches the process optimum. From this point of view, they can be estimated directly without computing process and model gradients explicitly. The module DME (Dynamic Modifier Estimation) of Figure 2 will try to estimate directly the modifiers using past data which, after filtering, will be incorporated in (5) and (6).

The idea behind DME approach is explained next, considering only the economic cost function in a continuous form for simplicity. Consider the value of a vector $\boldsymbol{\theta}$ such that the modified cost function that appears in equation (11.a) reproduces the measured process cost function ϕ_p over the past time interval $[t_{k-n_d}, t_k]$ as in Figure 4. n_d is a small integer number, the DME past horizon, that may coincide or not with the past horizon of the MHE.

$$\phi_p \approx \phi + \boldsymbol{\theta}^T \Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T \boldsymbol{Q} \Delta\boldsymbol{u} + \sum_{j=1}^{n_u-1} \Delta\boldsymbol{u}_j^T \boldsymbol{Q} \Delta\boldsymbol{u}_j \tag{11.a}$$

Where ϕ_p and ϕ represent respectively the measured process cost function, and the cost estimated by the model, obtained with the $\Delta\boldsymbol{u}$ applied to the process, in the interval of the DME past horizon. Both ϕ_p and ϕ are time functions obtained either by interpolation of measured

values or by model integration. In equation (11.a), the term $\Delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{Q} \Delta \boldsymbol{u} + \sum_{j=1}^{n_u-1} \Delta \boldsymbol{u}_j^{\boldsymbol{T}} \boldsymbol{Q} \Delta \boldsymbol{u}_j$ corresponds to the optimal value of the quadratic term in (5), computed in the resolution of the DOMA at past iterations $k - i$. Control moves $\Delta \boldsymbol{u}$ in $\Delta \boldsymbol{u}^{\boldsymbol{T}} \boldsymbol{Q} \Delta \boldsymbol{u}$ are the ones really applied to the process in the past. Finally, the summation term in (11.a) is the calculation of future moves that were not applied to the process because of the moving horizon strategy of the DOMA.

Taking into account that output variables are functions of the manipulated variables $\boldsymbol{u}$, one can compute the derivative of both sides of (11.a) w.r.t. $\boldsymbol{u}$ applied to the process and obtain (11.b). Notice that the derivative of summation term in (11.a) is zero because this term does not have the variable $\boldsymbol{u}$ applied to the process:

$$\frac{\partial \phi_p}{\partial \boldsymbol{u}} \approx \frac{\partial \phi}{\partial \boldsymbol{u}} + \boldsymbol{\theta}^T + 2\Delta \boldsymbol{u}^T \boldsymbol{Q}_u \tag{11.b}$$

$$\boldsymbol{\lambda}^T = \frac{\partial \phi_p}{\partial \boldsymbol{u}} - \frac{\partial \phi}{\partial \boldsymbol{u}} \approx \boldsymbol{\theta}^T + 2\Delta \boldsymbol{u}^T \boldsymbol{Q}_u \tag{11.c}$$

Now, if we rearrange equation (11.b) and use the definition of the modifiers, we could obtain an estimation of $\boldsymbol{\lambda}$ using (11.c) at any transient time, calculating the vector $\boldsymbol{\theta}$ that match the past process cost as in (11.a).

In this way, we could consider the value of $\boldsymbol{\theta}^T + 2\Delta \boldsymbol{u}^T \boldsymbol{Q}_u$ at t_k as an estimation of the cost modifiers at k. So, when the system reaches steady state, $\boldsymbol{\theta}^T + 2\Delta \boldsymbol{u}^T \boldsymbol{Q}_u$ will correspond to the steady-state modifiers.

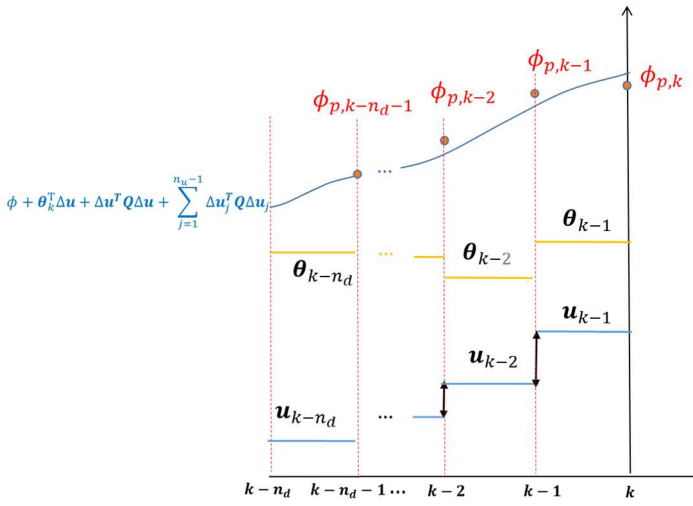


Figure 4. Past process cost function and evolution of the extended model cost function corresponding to past values of control actions and values of $\boldsymbol{\theta}_{k-i}$ and $\boldsymbol{u}_{k-i}$.

Finding such values of $\boldsymbol{\theta}$ can be formulated as an optimization problem in the interval $[t_{k-n_d}, t_k]$ in which the cost process evolution ϕ_p and control moves applied are known. The cost function of the optimization problem would incorporate the squared difference between the process cost and the RHS of equation (11.a) in the previous sampling times. Then, the value of the modifiers (11.c) corresponding to the last interval $[t_{k-1}, t_k]$ is used in the DOMA problem to compute the next optimal control actions. The same idea can be extended to compute the constraints' modifiers $\boldsymbol{\gamma}$, using in the cost function the squared difference between the process and the modified constraint from (6.c).

Formally, the DME problem can be formulated as in (12) and (13) for $\boldsymbol{\lambda}_k$, and (14) for $\boldsymbol{\gamma}_k$. At iteration k we know the past actions applied $\boldsymbol{u}_{k-i}$ and vectors $\boldsymbol{\lambda}_{k-i}$ and $\boldsymbol{\gamma}_{k-i}$ used in previous iterations, as well as the values of the process cost function $\phi_{p,k-i}$ and constraints $\boldsymbol{g}_{p,k-i}$ obtained. The only unknowns are $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}^{DME}$, which are parameterized conveniently. In (12) – (14), n_d is the DME past horizon.

$$\min_{\substack{\boldsymbol{\theta}_{k-i-1},\\ i=0\dots n_d-1}} \sigma_\phi \int_{t_{k-n_d}}^{t_k} \Delta\phi^2 dt + \sum_{i=0}^{n_d-1} \left(\Delta\boldsymbol{\theta}_{k-i-1}^T \boldsymbol{Q}_\theta \Delta\boldsymbol{\theta}_{k-i-1}\right) \tag{12.a}$$

$$s.t.\ \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \boldsymbol{0} \tag{12.b}$$
$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \boldsymbol{0}$$

$$\Delta\phi(t) = \phi_p(t) - [\phi(t) + \boldsymbol{\theta}(t)^T \Delta\boldsymbol{u}(t) + \boldsymbol{S}(t)] \tag{12.c}$$
$$\boldsymbol{S}(t) = \Delta\boldsymbol{u}(t)^T \boldsymbol{Q}_{\boldsymbol{u}} \Delta\boldsymbol{u}(t)$$

$$\boldsymbol{\theta}(t) = \boldsymbol{\theta}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \tag{12.d}$$

$$\boldsymbol{\theta}_{k-i-1} = \overline{\overline{\boldsymbol{\theta}}}_{k-i-1} - \Delta\boldsymbol{\theta}_{k-i-1}\ , \qquad i = 0, \dots, n_d - 1$$

$$\boldsymbol{u}(t) = \boldsymbol{u}_{k-i-1}\,, t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \tag{12.e}$$
$$\Delta\boldsymbol{u}_{k-i-1} = \boldsymbol{u}_{k-i} - \boldsymbol{u}_{k-i-1}\ , \quad i = 0,1, \dots n_d - 1$$

$$\boldsymbol{x}(t_{k-n_d}) = \hat{\boldsymbol{x}}_{k-n_d} \tag{12.f}$$

In the first term of the cost function (12.a), expanded in (12.c), $\boldsymbol{S}$ is the value of the quadratic term that penalizes changes in the manipulated variable from (5) solved at the past iteration. σ_ϕ is a scalar weight for this difference. Notice that (12.a), (12.c) also include another quadratic term that try to smooth the changes over the values computed in the previous iterations (12.d) and contribute to convexify the problem, being $\boldsymbol{Q}_\theta \in \mathbb{R}^{\boldsymbol{n_u} \times \boldsymbol{n_u}}$ a diagonal matrix with weighting factors for each manipulated variable. The optimization is subjected to the dynamic model (12.b). The problem uses the value $\hat{\boldsymbol{x}}_{k-n_d}$ that was estimated by the MHE as an initial state (12.f) and includes the disturbances $\boldsymbol{v}_k$ to maintain consistency with the other modules. Also notice that ϕ and ϕ_p are computed as continuous functions interpolating the past iterations $k - i$. $\overline{\overline{\boldsymbol{\theta}}}$ is the vector of optimum solutions of problem (12) obtained in the last iteration.

The solution of (12), $\boldsymbol{\theta}_{k-1}^*$, is considered equal to $\boldsymbol{\theta}_k$ and allows estimating the cost modifiers in iteration k using (13):

$$\boldsymbol{\lambda}_k^{DME} = \boldsymbol{\theta}_k{}^{\boldsymbol{T}} + 2\Delta\boldsymbol{u}_{k-1}^T \boldsymbol{Q}_{\boldsymbol{u}}, \qquad \boldsymbol{\theta}_k = \boldsymbol{\theta}_{\boldsymbol{k-1}}^* \tag{13}$$

In the same way, we can formulate a similar problem (14) to compute each constraint's modifiers $\boldsymbol{\gamma}$. $\boldsymbol{\sigma}_g \coloneqq \left[\sigma_{g,1}, \dots, \sigma_{g,n_g}\right]^T$ is a column vector with the weighting values for each constraint and $\boldsymbol{\sigma}_\gamma^j \in \mathbb{R}^{n_u \times n_u}$ a diagonal matrix with the weighting values for each constraint modifier, $j = 1 \dots n_g$.

$$\min_{\substack{\boldsymbol{\gamma}_{k-i-1}^{DME} \\ i=0\dots n_d-1}} \sum_{j=0}^{n_g} \sigma_{g,j} \int_{t_{k-n_d}}^{t_k} {\Delta g_j}^2 dt + \sum_{j=1}^{n_g} \sum_{i=0}^{n_d-1} \left(\Delta \boldsymbol{\gamma}_{k-i-1}^{j,DME\,T} \sigma_\gamma^j I \Delta \boldsymbol{\gamma}_{k-i-1}^{j,DME} \right) \tag{14.a}$$

$$s.t.\ \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}_k) = \boldsymbol{0} \tag{14.b}$$
$$\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{y}, \boldsymbol{v}_k) = \boldsymbol{0}$$

$$\Delta \boldsymbol{g}(t) = \boldsymbol{g}_P(t) - \boldsymbol{g}(t) - \boldsymbol{\gamma}^{DME}(t) \Delta \boldsymbol{u}(t), \qquad i = 0 \dots n_d - 1 \tag{14.c}$$

$$\boldsymbol{\gamma}^{DME}(t) = \boldsymbol{\gamma}_{k-i-1}^{DME}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \tag{14.d}$$
$$\Delta \boldsymbol{\gamma}_{k-i}^{DME} = \boldsymbol{\gamma}_{k-i}^{DME} - \bar{\bar{\boldsymbol{\gamma}}}_{k-i}, \qquad i = 1, \dots, n_d$$

$$\boldsymbol{u}(t) = \boldsymbol{u}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \tag{14.e}$$

$$\boldsymbol{x}\left(t_{k-n_d}\right) = \hat{\boldsymbol{x}}_{k-n_d} \tag{14.f}$$

The solution of problem (14) is $\boldsymbol{\gamma}_{k-1}^{DME^*}$. Similarly, to the estimation of $\boldsymbol{\theta}_k$, the value of $\boldsymbol{\gamma}_k^{DME}$ is taken equal to $\boldsymbol{\gamma}_{k-1}^{DME^*}$. So, when the system reaches steady state $\boldsymbol{\gamma}_k^{DME}$ will correspond to the steady-state modifiers.

Finally, (4.c), (12), (13), and (14) give the new values of the cost and constraint modifiers $\boldsymbol{\Lambda}_k^{DME} = (\boldsymbol{\lambda}_k^{DME}, \boldsymbol{\gamma}_k^{DME}, \boldsymbol{\varepsilon}_k^{DME})$ which, after filtering (15), can be used as estimations of the modifiers to be used at iteration k in the DOMA problem. The value of $\boldsymbol{\varepsilon}_k^{DME}$ is obtained using (4.c) directly, while the superscript "DME" has been added to differentiate this value with the one applied to problem (5), (6), obtained after filtering. The term $\boldsymbol{\Lambda}_{k-1}$ is the vector of filtered modifiers applied to the process in the previous iteration $k-1$ and $\boldsymbol{\alpha}$ is a gain matrix.

$$\boldsymbol{\Lambda}_k = \boldsymbol{\alpha} \boldsymbol{\Lambda}_{k-1} + (\boldsymbol{I} - \boldsymbol{\alpha}) \boldsymbol{\Lambda}_k^{DME} \tag{15}$$
$$\boldsymbol{\Lambda}_k = (\boldsymbol{\lambda}_k, \boldsymbol{\gamma}_k, \boldsymbol{\varepsilon}_k)$$

Where $\boldsymbol{\alpha}$ can be represented by a block-diagonal matrix given by (16).

$$\boldsymbol{\alpha} := diag(b_1, \dots, b_{n_u}, q_1 \boldsymbol{I}_{n_u}, \dots, q_{n_g} \boldsymbol{I}_{n_u}, d \boldsymbol{I}_{n_g}) \tag{16}$$

The gain entries $b_1, \dots, b_{n_u}, q_1, \dots, q_{n_g}, d$ are taken in $(0,1]$. n_g is the number of constraints $\boldsymbol{g}$ in the problem and n_u the number of manipulated variables.

4 ALGORITHM

The full algorithm of the DOMA architecture proposed in this paper (Figure 2) is presented below:

Algorithm DOMA using DME to estimate MA modifiers.

i. Initialization

1. Collect $N := \max\{n_d, n_e\}$ previous data (sample time of the controller) of variables $\boldsymbol{u}_{k-i}$, $\boldsymbol{y}_{p,k-i}$, $\phi_{p,k-i}$, $\boldsymbol{g}_{p,k-i}$ where $i = 1, \dots, N$.

   MHE module: given $\boldsymbol{Q}_x$, $\boldsymbol{Q}_y$ and $\boldsymbol{Q}_v$

2. Initialize $\hat{\boldsymbol{x}}_{k-n_e}$

a. For measured states, consider $\hat{\boldsymbol{x}}^M_{k-n_e} = \boldsymbol{y}^S_{p,k-n_e}$, being $\hat{\boldsymbol{x}}^M_{k-n_e}$ the subset of states that are measured, and $\boldsymbol{y}^S_{p,k-n_e}$ the subset of $\boldsymbol{y}_{p,k-n_e}$ containing the measured states in $k - n_e$
b. For unmeasured states, use values predicted from the model with $\boldsymbol{v}_{k-i} = \boldsymbol{0}, i = 1 \dots n_e$

3. Solve problem (7), (8) to find the past values of the states $\boldsymbol{x}^*_{k-n_e}$ and disturbances $\boldsymbol{v}^*_{k-i}, i = 1 \dots n_e$.
4. Evaluate the estimated state $\hat{\boldsymbol{x}}_k$ integrating (9) from t_{k-n_e} to t_k. Estimate the value of the disturbance using $\boldsymbol{v}_k = \boldsymbol{v}^*_{k-i}$
5. Update $\hat{\boldsymbol{x}}_{k-n_e} = \boldsymbol{x}^*_{k-n_e}$.

DME module: given $\boldsymbol{\sigma}_\phi$, $\boldsymbol{Q}_\theta$, $\boldsymbol{\sigma}_g$, $\boldsymbol{\sigma}^j_\gamma$ where $j = 1 \dots n_g$, and $\boldsymbol{\alpha}$

6. Initialize previous values of $\boldsymbol{\lambda}_{k-i}$ and $\boldsymbol{\gamma}_{k-i}, i = 1 \dots n_d$, using an identification method such as recursive least squares. Here the idea is starting with good estimators of the modifiers to improve the convergence of the DME module.
7. Evaluate $\boldsymbol{\varepsilon}_{k-1}$ using (4.a)
8. Solve problems (12) and (14) to obtain $\boldsymbol{\theta}^*_{k-1}, \boldsymbol{\gamma}^{DME^*}_{k-1}$.
9. Use (13) to obtain $\boldsymbol{\lambda}^{DME}_k$ and (4.c) to obtain $\boldsymbol{\varepsilon}^{DME}_k$**.** Make $\boldsymbol{\gamma}^{DME}_k = \boldsymbol{\gamma}^{DME^*}_{k-1}$
10. Use (15) and obtain the filtered modifiers $\boldsymbol{\Lambda}_k$.

DOMA module: given $\boldsymbol{Q}_u$

11. Go to step 19

ii. **For next iterations**

MHE module:

12. Collect n_e previous data of variables $\boldsymbol{u}_{k-i}$, $\boldsymbol{y}_{p,k-i}$, $\phi_{p,k-i}$, $\boldsymbol{g}_{p,k-i}$ where $i = 1, \dots, n_e$.
13. Solve problem (7), (8) to find the past values of the states $\boldsymbol{x}^*_{k-n_e}$ and disturbances $\boldsymbol{v}^*_{k-i}$.
14. Evaluate the estimated state $\hat{\boldsymbol{x}}_k$ integrating (9) from t_{k-n_e} to t_k. Estimate the value of the disturbance using $\boldsymbol{v}_k = \boldsymbol{v}^*_{k-i}$
15. Update $\hat{\boldsymbol{x}}_{k-n_e} = \boldsymbol{x}^*_{k-n_e}$

DME module:

16. Solve problems (12) and (14) to obtain $\boldsymbol{\theta}^*_{k-1}, \boldsymbol{\gamma}^{DME^*}_{k-1}$.
17. Use (13) to obtain $\boldsymbol{\lambda}^{DME}_k$ and (4.c) to obtain $\boldsymbol{\varepsilon}^{DME}_k$. Make $\boldsymbol{\gamma}^{DME}_k = \boldsymbol{\gamma}^{DME^*}_{k-1}$
18. Use (15) and obtain the filtered modifiers $\boldsymbol{\Lambda}_k$.

DOMA module:

19. Solve problem (5), (6) using $\hat{\boldsymbol{x}}_k$ and $\boldsymbol{v}_k$ from MHE, and $\boldsymbol{\Lambda}_k$ calculated in DME.
20. Apply $\boldsymbol{\Delta u}_k$ to the process.
21. Wait until next sample time and update $k = k + 1$
22. Go to step 12.

5 WILLIAMS OTTO REACTOR CASE STUDY

5.1 The Williams Otto reactor

Results for the proposed method for DOMA+DME are shown using a variation of the Williams-Otto benchmark example (Williams and Otto, 1960) using molar concentrations. This example has been widely used to study the performance of others eMPC+MA frameworks as mentioned in Table 1. The process consists of a continuous stirred tank reactor CSTR where the reactants A and B combine to generate four species C, E, G, P in three different reactions.

$$A + B \xrightarrow{k_1} C \qquad B + C \xrightarrow{k_2} P + E \qquad C + P \xrightarrow{k_3} G$$

The system is fed with a flow of component A (F_A), and component B (F_B), with an inlet concentration X_{A0} and X_{B0}, respectively. The products leave the reactor by overflow, so that its volume V_R can be considered constant. The concentration of the outflow, F_R, for each component is denoted by X_A, X_B, X_C, X_E, X_G and X_P.

The velocity of each reaction is given by equations (17)-(19). The rate constant of each reaction, k_i, is described by the Arrhenius equation (20) that depends on the pre-exponential factor $k_{i,0}$, the activation energy E_{ai} for each reaction, and the temperature inside the reactor, T.

$$r_1 = k_1 X_A X_B \tag{17}$$

$$r_2 = k_2 X_B X_C \tag{18}$$

$$r_3 = k_3 X_C X_P \tag{19}$$

$$k_i = k_{i,0} e^{-(E_{ai}/T)}, \qquad i = 1,2,3 \tag{20}$$

A cooling system maintains the temperature of the reactor around the desired value *T*. It is assumed that no significant change in the density or similar intrinsic parameters takes place.

The inflow F_B, and the temperature set point *T* are the manipulated process variables, while inflow F_A can be considered as a disturbance. Products P and E are the valuable ones, while C and G are sub-products of the reactions.

The nonlinear dynamics of the process is given by equations (21)-(26), which combined with the total mass balance (27) will be used in a simulation that will be considered as the real plant.

Table 2 presents the parameters used in the simulation.

$$V_R \frac{dX_A}{dt} = F_A X_{A0} - F_R X_A - V_R r_1 \tag{21}$$

$$V_R \frac{dX_B}{dt} = F_B X_{B0} - F_R X_B - V_R r_1 - V_R r_2 \tag{22}$$

$$V_R \frac{dX_C}{dt} = -F_R X_C + V_R r_1 - V_R r_2 - V_R r_3 \tag{23}$$

$$V_R \frac{dX_E}{dt} = -F_R X_E + V_R r_2 \tag{24}$$

$$V_R \frac{dX_G}{dt} = -F_R X_G + V_R r_3 \tag{25}$$

$$V_R \frac{dX_P}{dt} = -F_R X_P + V_R r_2 - V_R r_3 \tag{26}$$

$$F_R = F_A + F_B \tag{27}$$

Table 2: Parameters of Williams-Otto reactor real process.

$k_{1,0}$	9.9594×10^6	$l/(mol\ min)$
$k_{2,0}$	8.66124×10^9	$l/(mol\ min)$
$k_{3,0}$	1.6047×10^{13}	$l/(mol\ min)$
E_{a1}	6666.7	K
E_{a2}	8333.3	K
E_{a3}	11111	K
X_{A0}	10	mol/l
X_{B0}	10	mol/l
V_R	2105	l
F_A	112.35	l/min

Table 3: Cost of the reactants and price of products.

p_A	$7.623\ €/mol$
p_B	$11.434\ €/mol$
p_p	$114.338\ €/mol$
p_E	$5.184\ €/mol$

The purpose of the optimal operation is to maximize the operation benefits computed as the difference between the value of the useful products E and P and the cost of the raw materials A and B. Table 3 provides prices of these products. If a perfect steady model of the process were available and no disturbances would take place, the optimal operating point could be obtained by solving the optimization problem (28). Here, the inflow F_B and reactor temperature T are constrained to operate within a certain range.

$$\begin{gathered} \max_{\boldsymbol{u}=[F_B,T]} \phi = F_R(X_P p_P + X_E p_E) - F_A X_{A0} p_A - F_B X_{B0} p_B \\ s.t.\ equations\ (21)-(26)\ in\ steady\ state \\ 180\ l/min \le F_B \le 360\ l/min \\ 75ºC \le T \le 100ºC \\ 0 \le X_A \le 1.2\ \mathrm{mol}/l \\ 0 \le X_G \le 0.5\ \mathrm{mol}/l \end{gathered} \tag{28}$$

The real optimum of the process in steady state is $F_B = 293.55\ l/min$ and $T = 89.98\ ºC$ and the value of objective function is $\phi = 11594.40€$ (black dot in Figure 5). As can be seen in Figure 5, which displays the value of the benefit ϕ for a range of values of the manipulated

variables, the cost is in a relatively flat zone, which can create convergence problems for the optimizers.

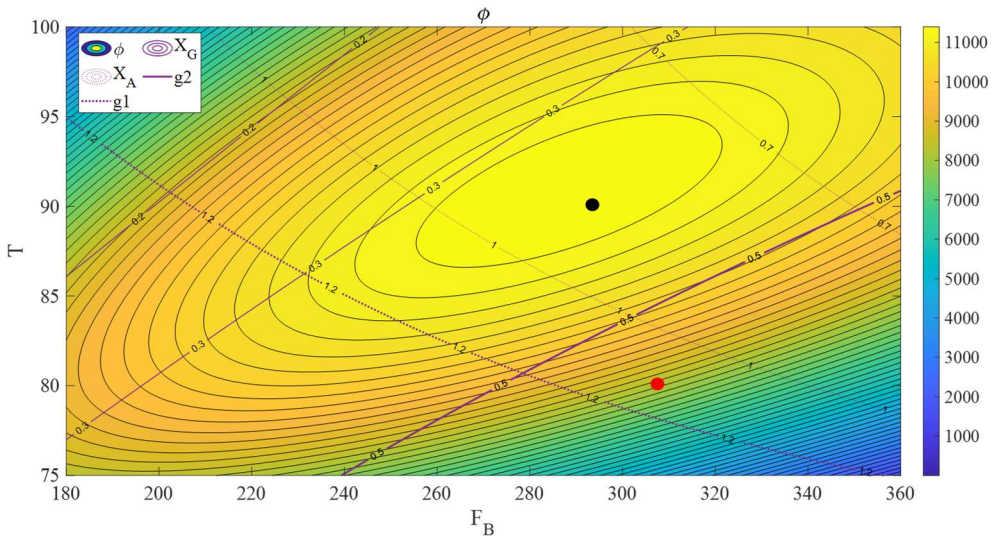


Figure 5. Process cost function and some contours values for X_A and X_G. The contour 1.2 (dotted line) and 0.5 (solid line) are the problem constraints in (28) for X_A and X_G respectively. The black and red dots correspond to plant and model optimum respectively.

5.2 Application of DOMA+DME

The DOMA+DME uses a simplified dynamic model that differs in structure and parameters from the process mentioned above. As the component C is one order of magnitude below the other components, a common choice is to consider only 5 components and two reactions:

$$A + 2B \xrightarrow{k_{1\_model}} P + E \quad A + B + P \xrightarrow{k_{2\_model}} G$$

The velocity of each reaction is given by equations (29)-(30). The rate constant of each reaction, k_{i_appr}, is described by the Arrhenius equation (31) that depends on the pre-exponential factor, $k_{i,0\_model}$, the activation energy E_{ai_model} for each reaction and the temperature inside the reactor, T.

$$r_{1\_model} = k_{1\_appr} X_A X_B^2 \tag{29}$$

$$r_{2\_model} = k_{2\_appr} X_A X_B X_P \tag{30}$$

$$k_{i\_appr} = k_{i,0\_model} e^{-(E_{ai\_model}/T)}, \qquad i = 1,2 \tag{31}$$

The dynamic model includes the following equations that represents mass balances of the different components considered:

$$V_R \frac{dX_A}{dt} = F_A X_{A0} - F_R X_A - V_R r_{1_{model}} - V_R r_{2_{model}} \tag{32}$$

$$V_R \frac{dX_B}{dt} = F_B X_{B0} - F_R X_B - 2V_R r_{1\_model} - V_R r_{2\_model} \tag{33}$$

$$V_R \frac{dX_E}{dt} = -F_R X_E + V_R r_{1\_model} \tag{34}$$

$$V_R \frac{dX_G}{dt} = -F_R X_G + V_R r_{2\_model} \tag{35}$$

$$V_R \frac{dX_P}{dt} = -F_R X_P + V_R r_{1_{model}} - V_R r_{2_{model}} \tag{36}$$

$$F_R = F_A + F_B \tag{37}$$

Therefore, the model presents a structural mismatch, since only five states are considered, instead of the six states of the process. In addition, we will assume that only four components A, E, G, P are measured at the reactor output. Model parameters are summarized in Table 4.

Table 4: Parameters of Williams-Otto dynamic model for the DOMA.

$k_{1,0\_model}$	1.3134×10^8	$l/(mol\ min)$
$k_{2,0\_model}$	2.586×10^{13}	$l/(mol\ min)$
E_{a1_model}	8077.6	K
E_{a2_model}	12438.5	K
X_{A0}	10	mol/l
X_{B0}	10	mol/l
V_R	2105	l
F_A	112.35	l/min

Solving a problem similar to (28), but using the model (29)-(36) in steady state without disturbances, one can compute the optimum that corresponds to that model in steady state. The solution is $F_B = 307.23\ l/min$ and $T = 79.82\ ºC$ with the objective function equal to $\phi = 11259.52€$ which is not far away from the optimal benefit of the process optimum. Nevertheless, when this solution is applied to the process, the value of the process cost is completely different, dropping the benefit down to $8655.4€$ (red dot in Figure 5). This remarks about the importance of integrating MA methods into dynamic optimizers in order to recover optimality.

DOMA uses the modified problem given by (38). The economic objective function considers the variables as computed at the end of the prediction horizon (line above variables). The states $\bar{X}_P$ and $\bar{X}_E$ are predicted using equations (29)-(36) adding the disturbances $\boldsymbol{v}$ from MHE. The two terms with λ_1 and λ_2 corresponds to the MA cost modifiers for each control variable. The last two terms are the penalties on the control efforts to smooth the control actions and contribute to model adequacy and convexity.

$$\max_{\boldsymbol{u}=[F_B,T]} \phi_M = \bar{F}_R\big((\bar{X}_P)p_P + (\bar{X}_E)p_E\big) - F_A X_{A0} p_A - \bar{F}_B X_{B0} p_B + \lambda_{1,k}\big(\bar{F}_B - F_{B,k-1}\big) + \lambda_{2,k}(\bar{T} - T_{k-1}) + \sigma_{u_1} \sum_{i=1}^{n_u-1} \big(\Delta F_{B,k+i}\big)^2 + \sigma_{u_2} \sum_{i=1}^{n_u-1} (\Delta T_{k+i})^2$$

$$s.t. \, equations \; (29) - (36) \text{ with disturbances } v$$

$$\begin{aligned} \Delta F_{B,k+i} &= F_{B,k+i} - F_{B,k+i-1}, \qquad i = 0 \dots n_{pred} - 1 \\ \Delta T_{k+i} &= T_{k+i} - T_{k+i-1}, \qquad i = 0 \dots n_{pred} - 1 \\ F_B(t) &= F_{B_{k+n_u}} \;, \; t \in \left[t_{k+n_u}, t_{pred}\right] \\ T(t) &= T_{k+n_u} \;, t \in \left[t_{k+n_u}, t_{pred}\right] \\ 180\frac{l}{min} \le F_{B,k+i} &\le 360\frac{l}{min}, \qquad i = 0 \dots n_{pred} - 1 \\ 75ºC \le T_{k+i} &\le 100ºC, \qquad i = 0 \dots n_{pred} - 1 \end{aligned} \tag{38}$$

Note that the cost function includes two conflicting objectives: maximize the benefit at the end of the prediction horizon and achieve a smooth and stable operation. This compromise is given by the weighting factors β_i . The MHE module provides the initial value for the state at current iteration k and disturbances.

Similarly, the DME problem at every sampling time k is given by (39). Here ϕ_p refers to the value measured in the process, as well as the concentrations $X_{p,j}$ (for each component j) , while ϕ is a value estimated from the model (29)-(36) with disturbances. ϕ_p is calculated as a continuous function of time, interpolating the measurements in the interval $[t_{k-3}, t_k]$. $\Delta\boldsymbol{u}$ and $\boldsymbol{\theta}$ are also calculated as continuous function of time t, using the equation in (39).

$$\min_{\substack{\boldsymbol{\theta}_{k-i-1} \\ i=0\dots n_d-1}} \sigma_\phi \int_{t_{k-n_d}}^{t_k} \Delta\phi^2 dt + \left(\Delta\boldsymbol{\theta}_{k-i-1}^T \boldsymbol{Q}_\theta \Delta\boldsymbol{\theta}_{k-i-1}\right)$$

$$s.t. \, equations \; (29) - (36) \text{ with disturbances } v$$

$$\begin{aligned} \Delta\phi(t) &= \left(\phi_p(t) - (\phi(t) + \boldsymbol{\theta}^T \Delta\boldsymbol{u}(t) + \Delta\boldsymbol{u}(t)^T \boldsymbol{Q}_u \Delta\boldsymbol{u}(t))\right) \\ \phi_p(t) &= F_R\big(X_{p,P}(t)p_P + X_{p,E}(t)p_E\big) - F_A X_{A0} p_A - F_B X_{B0} p_B \\ \phi(t) &= F_R(X_P(t)p_P + X_E(t)p_E) - F_A X_{A0} p_A - F_{B,k-i} X_{B0} p_B \\ \boldsymbol{u}_{k-i-1} &= \left[F_{B,k-i-1} \; T_{k-i-1}\right]^T, \qquad i = 0 \dots n_d - 1 \\ \boldsymbol{u}(t) &= \boldsymbol{u}_{k-i-1} \,, t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \\ \boldsymbol{\theta}(t) &= \boldsymbol{\theta}_{k-i-1}, \qquad t \in [t_{k-i-1}, t_{k-i}], \qquad i = 0, \dots, n_d - 1 \\ \Delta\boldsymbol{\theta}_{k-i-1} &= \boldsymbol{\theta}_{k-i-1} - \bar{\bar{\boldsymbol{\theta}}}_{k-i-1-1} \;, \qquad i = 0 \dots n_d - 1 \end{aligned} \tag{39}$$

Equation (13) uses the solution $\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1}^*$ of (39) and filtering (15) to compute $\boldsymbol{\lambda}_k$ that is used in (38).

5.3 Results

Results of the proposed methods are given next. The starting point is given in Table 5 and corresponds to a value of the process state that is far from the optimum.

Table 5: Starting point of the simulation.

X_A	0.8264	mol/l
X_B	4.8075	mol/l
X_C	0.0843	mol/l
X_E	1.2618	mol/l
X_G	0.2059	mol/l
X_P	1.0558	mol/l
F_B	350	l/min
T	82	^{o}C

The DOMA problem (5), (6) could be solved numerically either with simultaneous or sequential approaches. The first one implies full discretization of the model, e.g. using orthogonal collocation on finite elements, which transforms the dynamic optimization into a NLP problem that can be solved with algorithms like IPOPT (Wächter and Biegler, 2006). This approach increases considerably the size of the problem to solve but allows a good treatment of path constraints and facilitate the use of automatic differentiation to calculate exact derivatives. Depending on the specific case considered, the sequential approach may offer also a good alternative by combining a dynamic simulation of the model with a NLP solver like SNOPT (Gill et al., 2005). One advantage of this method is the smaller size of the optimization problem and the fact that it does not require state discretization but requires integrating the extended system for computing exact derivatives and it is more difficult to deal with path constraints and unstable systems. In the MHE problem, similar numerical methods can be used. In the case study presented in this section a sequential approach has been used.

Table 6: Parameters used in the simulation for DOMA+DME.

Module	Parameter	Value
MHE	$\sigma_{\boldsymbol{y}}$ $(\boldsymbol{Q_y} = \sigma_{\boldsymbol{y}}\boldsymbol{I})$	250
MHE	$\sigma_{\boldsymbol{x}}$ $(\boldsymbol{Q_x} = \sigma_{\boldsymbol{x}}\boldsymbol{I})$	1×10^{-2}
MHE	$\sigma_{\boldsymbol{v}}$ $(\boldsymbol{Q_v} = \sigma_{\boldsymbol{v}}\boldsymbol{I})$	1×10^{-5}
MHE	n_e	3
DME	σ_ϕ	10
DME	$\boldsymbol{Q}_\theta$	$\begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}$
DME	n_d	3
DOMA	σ_{u_1}	0.05
DOMA	σ_{u_2}	0.44
DOMA	α	0.95
DOMA	n_u	3
DOMA	n_{pred}	30

The reactor and the three dynamic optimization problems DOMA, MHE and DME, are formulated in the continuous domain in the simulation environment EcosimPro/Proosis (EA Int., 2020), a modern object-oriented software. They are solved with a sampling time of 2 minutes. The optimization problems have been solved using a sequential approach in which a dynamic simulation of the model is connected to SNOPT, a reduced-space SQP optimization algorithm. Exact derivatives of cost functions and constraints are generated by the integration of the extended system as provided by the integration software IDAS. The entire problem is solved in an average time of 3 sec. per iteration in a PC under Windows10 with a four cores i7 processor running at 3GHz and with 16 Gb of memory. The DOMA module starts at time $t = 8\,min$ to allow collecting past measurements. The MHE, DME and DOMA parameters used are in Table 6.

Figure 6 shows the states estimated by the MHE (blue line), together with the measured values from the process (red line). Estimated disturbances appear at the bottom for the five states of the model.

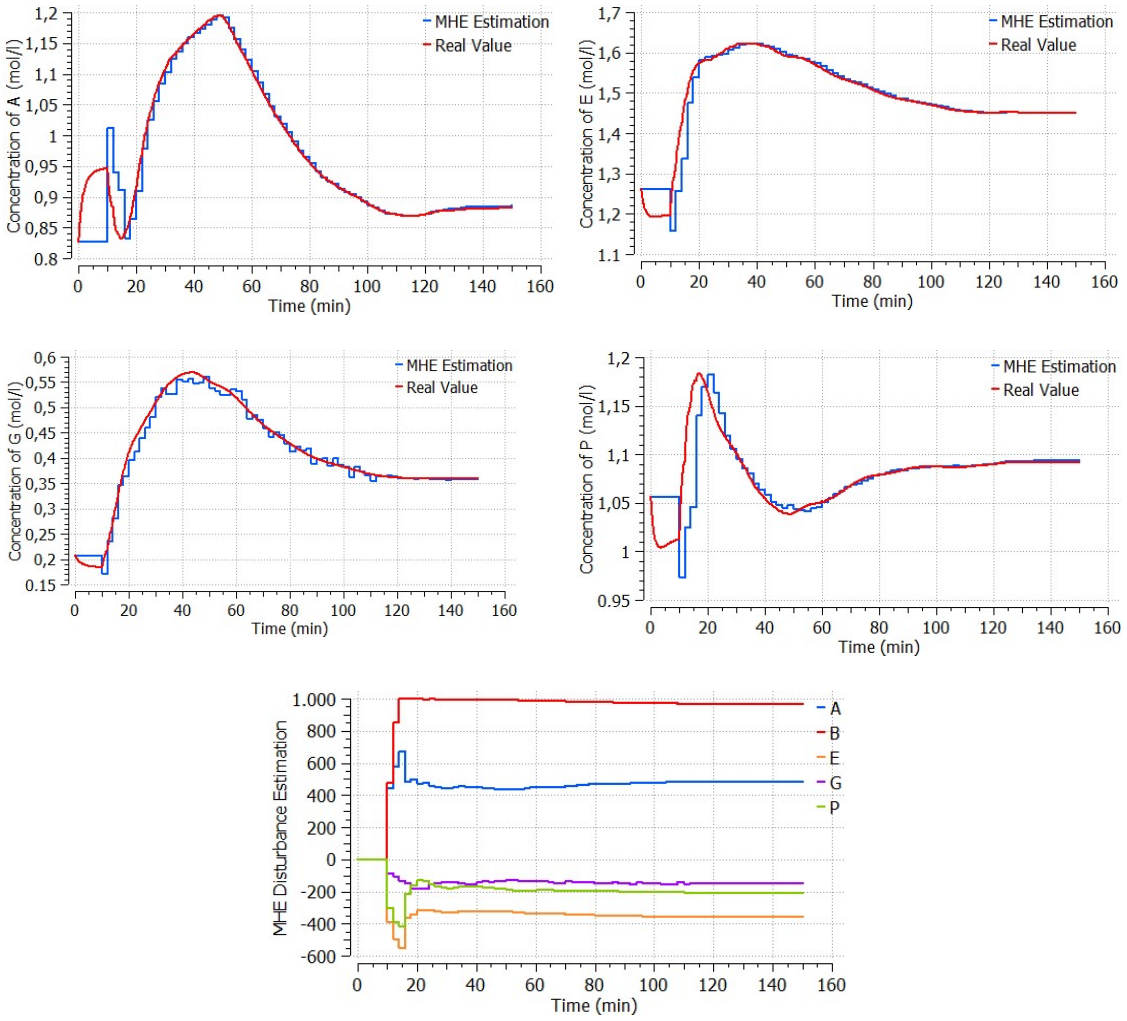


Figure 6. Results of the MHE module

The lambda modifiers computed by DME after filtering are presented in Figure 7. Figure 8 shows the time evolution of the process cost function and Figure 9 shows the manipulated variables.

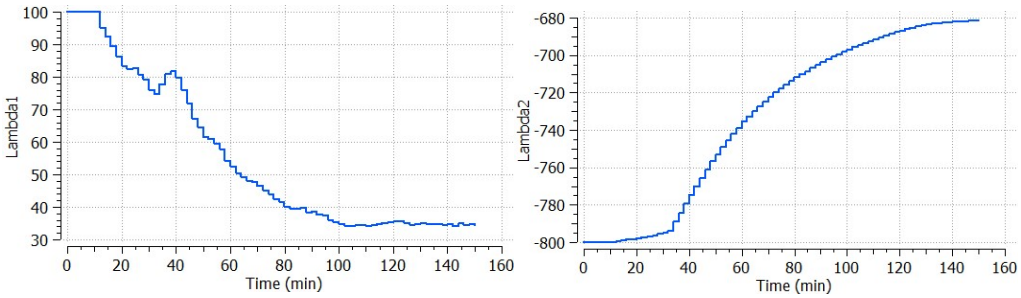


Figure 7. Values of lambda modifiers calculated from the solution of the online DME.

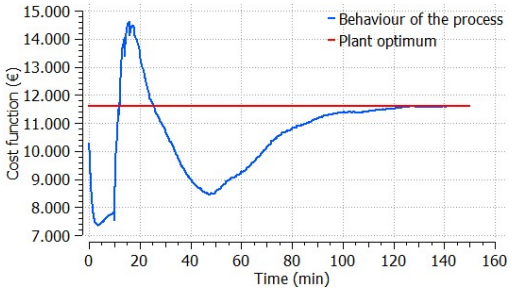


Figure 8. Cost function value during simulation DOMA+DME.

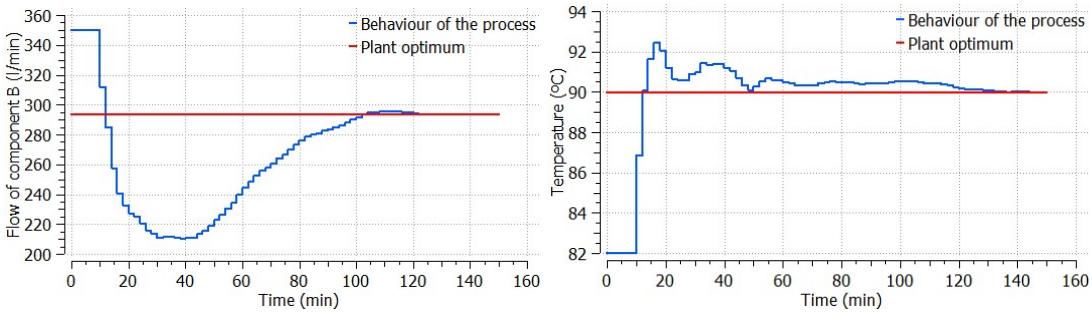


Figure 9. Manipulated variables over time DOMA+DME.

As can be seen, after an initial transient, the closed loop system stabilizes around 140 minutes, which is about four times the open loop response of the reactor (40 min), reaching an average benefit pretty close to the real process optimum.

The next figures (Figure 10 and Figure 11) compare the results of the use of dynamic optimizer without modifiers, DOMA+DME and DOMA with correct modifiers. The application of DOMA (dark blue line) clearly improves the cost function of the process in comparison to the application of the dynamic optimizer without the correction in the economic function (green line). For comparison, the closed loop evolution of the DOMA with real optimum modifiers (assuming they were known) is also added (orange line). It can be noticed that using the real values of the modifiers in steady state, the system reaches the optimal operation in 50 minutes, which is almost one third of the convergence time of the DOMA using DME to estimate the modifiers. This reduction in the performance of the proposed algorithm can be explained because of the use of transient measurements (and not steady-state measurements) to estimate the modifiers. Nevertheless, the algorithm could reach a stationary point in the neighborhood of the process optimum in a short time. The small errors obtained can be attributed to the low sensibility that this variable can have in ϕ_p near to the optimum (flat zone commented from Figure 5).

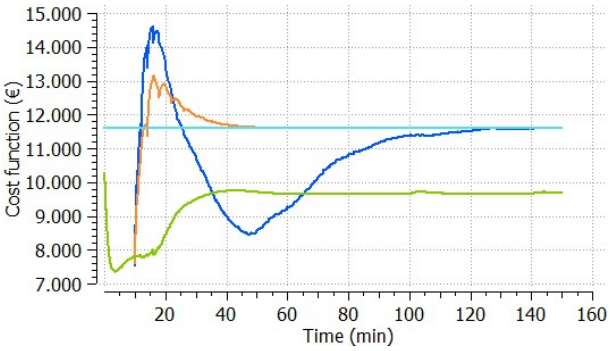


Figure 10: Comparison of the cost function with DOMA+DME (dark blue), DOMA with perfect modifiers (orange) and without MA (green). The light blue line corresponds to the plant optimum.

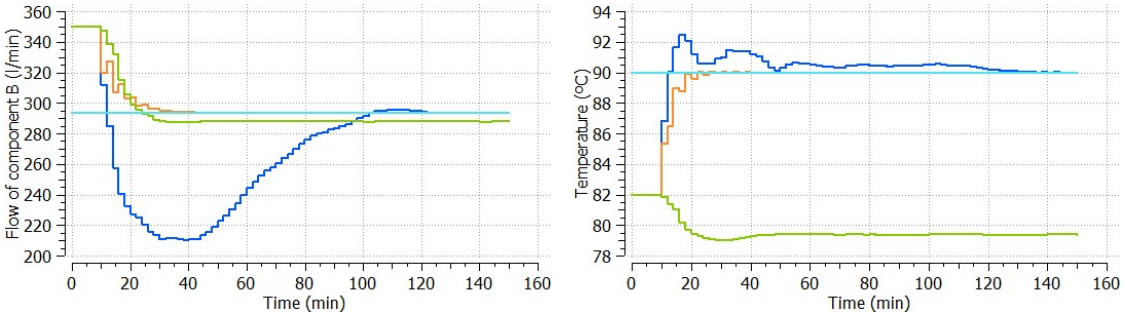
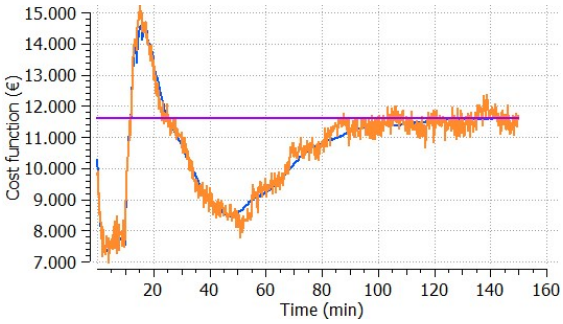


Figure 11: Comparison of the control actions with DOMA with DME (dark blue), MA with perfect modifiers (orange) and without MA (green). The light blue line corresponds to the plant optimum.

We made comparisons between the dynamic optimizer without modifiers, DOMA using perfect modifiers and DOMA with the modifiers obtained with the DME method. Comparisons with other papers that use the Williams Otto case study are not easy, as, surprisingly, many of them do not use the same models and parameters (Table 1). Besides, the comparison is subject to the value of the algorithm's parameters that must be estimated from previous simulations and then implementation of the algorithm. Nevertheless, just to illustrate the order of magnitude of the ratio between the time to convergence using gradients estimated online with process measurements and considering the gradients known, the DOMA using DME achieve a value about 3 (140 minutes /50 minutes).

Finally, noise has been added to the process reflecting in 4% oscillations in the cost function, as shown in Figure 12. The noise decreases the performance of the DME algorithm, creating a small offset in the decision variables.



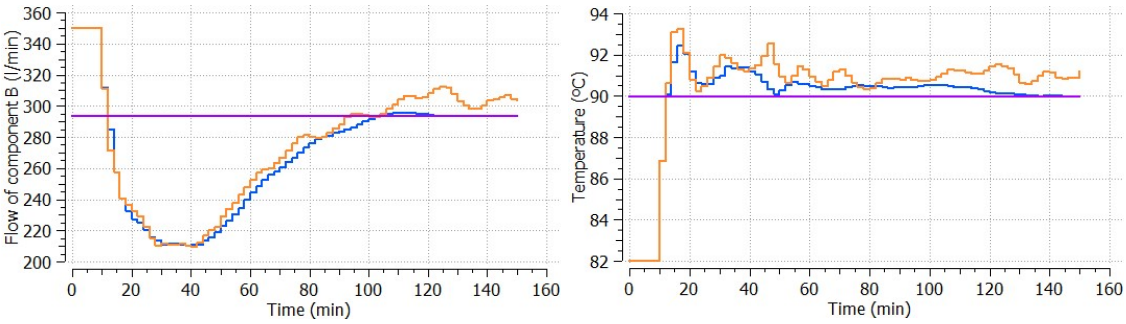


Figure 12: Up: Cost function without (dark blue) and with (orange) noise added to the process. Bottom: control efforts without (dark blue) and with (orange) noise added to the process. The light purple line corresponds to the plant optimum.

6 CONCLUSIONS

This paper aims to study new ways of improving the practical application of MPC and RTO in industrial processes and their integration under the MPC framework. Good models are a key element of these technologies, however, as they are only approximations of reality, and usually not well maintained, a successful application in the industry depends on the integration of correction tools such as Modifier Adaptation. In addition, because of the time scales of many processes in the industry, these tools need to be applied on-line at rates much higher than the large number of steady states required by traditional MA. This paper wishes to contribute to the research in the field proposing one approach that combines standard RTO, MPC with Modifier Adaptation methodology and an additional module for direct estimation of the modifiers. The case study presented showed that this new approach can achieve values quite close to the real economic optimum operational point despite the parametric and structural mismatch between model and process. As the performance of the proposed algorithm depends on the value of the weighting factors from the different optimization problems, this can be a drawback to implementation in a real process. More research is required to analyze the joint operation of the three modules and a methodology to select tuning values, but this is out of the scope of this work. To improve the convergence time, further research might investigate other approaches for the balance between performance and stability, besides the application of different process gradients estimation methods from transient data. Future work also aims to formalize DOMA regarding the properties of the model that can ensure convergence of the method and conditions for convergence. Beside it will be studied the use of linear models in DOMA, as happens in industrial MPC controllers such as DMC (Dynamic Matrix Control).

ACKNOWLEDGMENTS

This work was supported by JCyL under programs CLU 2017-09 and UIC 233, as well as with project InCO4In, from Spanish AEI under project PGC2018-099312-B-C31, both with FEDER funds. The first author thanks the European Social Fund and the “Conserjería de Educación de la Junta de Castilla y León”.

NOMENCLATURE

Nomenclature	
E_a	Activation energy
F	Flow of component
$\boldsymbol{f}$ $\boldsymbol{h}$	Model equations
$\boldsymbol{g}$	Constraints of the optimization problem
i j	Index integer or component
k	Current iteration
n_u	Control horizon
n_{pred}	Prediction horizon
n_e	MHE horizon
n_d	DME horizon
n_g	Number of constraints
p	Cost or price of reactants or products
$\boldsymbol{Q}$	Positive definite matrix with weighting and/or normalization factors
r	Reaction rate
T	Temperature
t	Time
$\boldsymbol{u}$	Manipulated variable
V_R	Reactor volume
$\boldsymbol{v}$	Disturbances in state variables vector
X	Molar concentration
$\boldsymbol{x}$	State variables vector
$\hat{\boldsymbol{x}}$	Estimated state variables vector
$\boldsymbol{y}$	Output variable
Greek letters	
$\boldsymbol{\alpha}$	MA filters
ϕ	Cost function of the optimization problem
σ	Weighting factors
$\boldsymbol{\lambda}$	Cost function first order MA modifiers
$\boldsymbol{\gamma}$	Constraint first order MA modifiers
$\boldsymbol{\varepsilon}$	Constraint zero order MA modifiers
$\boldsymbol{v}$	Disturbances
$\boldsymbol{\theta}$	Decision variable of DME module
$\boldsymbol{\Lambda}$	Filtered modifiers
β	Weighting factor
Superscript	
*	Optimum
^	Value estimated
—	Value at the end of the prediction horizon
M	Measured
S	Subset
Subscript	
p	Plant
M	Cost function or constraint with MA modifiers
k	Current iteration
$pred$	Value at the end of the prediction horizon
DME	Values calculated in the DME module

REFERENCES

Brdyś, M.A., Tatjewski, P., 1994. An Algorithm for Steady-State Optimizing Dual Control of Uncertain Plants. IFAC Proc. Vol. 27, 215–220. https://doi.org/10.1016/S1474-6670(17)47650-6

Câmara, M., Quelhas, A., Pinto, J., 2016. Performance Evaluation of Real Industrial RTO Systems. Processes 4, 44. https://doi.org/10.3390/pr4040044

EA Int., 2020. EcosimPro 6.2 [WWW Document]. User Man. URL https://www.ecosimpro.com/

Faulwasser, T., Pannocchia, G., 2019. Toward a Unifying Framework Blending Real-Time Optimization and Economic Model Predictive Control. Ind. Eng. Chem. Res. 58, 13583–13598. https://doi.org/10.1021/acs.iecr.9b00782

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

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

François, G., Bonvin, D., 2013. Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation. Ind. Eng. Chem. Res. 52, 11614–11625. https://doi.org/10.1021/ie3032372

Galan, A., de Prada, C., Gutierrez, G., Sarabia, D., Grossmann, I.E., Gonzalez, R., 2019. Implementation of RTO in a large hydrogen network considering uncertainty. Optim. Eng. 20, 1161–1190. https://doi.org/10.1007/s11081-019-09444-3

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

Gill, P.E., Murray, W., Saunders, M.A., 2005. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. SIAM Rev. 47, 99–131. https://doi.org/10.1137/S0036144504446096

Hernández, R., Engell, S., 2019. Economics optimizing control with model mismatch based on modifier adaptation. IFAC-PapersOnLine 52, 46–51. https://doi.org/10.1016/j.ifacol.2019.06.035

Huang, R., Patwardhan, S.C., Biegler, L.T., 2010. Offset-free Nonlinear Model Predictive Control Based on Moving Horizon Estimation for an Air Separation Unit, IFAC Proceedings Volumes. IFAC. https://doi.org/10.3182/20100705-3-be-2011.00105

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

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

Navia, D., Briceño, L., Gutiérrez, G., de Prada, C., 2015. Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem. Ind. Eng. Chem. Res. 54, 12054–12071. https://doi.org/10.1021/acs.iecr.5b01946

Pannocchia, G., 2018. An economic MPC formulation with offset-free asymptotic performance.

IFAC-PapersOnLine 51, 393–398. https://doi.org/10.1016/j.ifacol.2018.09.332

Papasavvas, A., de Avila Ferreira, T., Marchetti, A.G., Bonvin, D., 2019. Analysis of output modifier adaptation for real-time optimization. Comput. Chem. Eng. 121, 285–293. https://doi.org/10.1016/j.compchemeng.2018.09.028

Rawlings, J.B., Mayne, D.Q., Diehl, M.M., 2019. Model predictive control:Theory, Computation, and Design, 2nd ed, Studies in Systems, Decision and Control. Nob Hill Publishing, Santa Barbara.

Rodríguez-Blanco, T., Sarabia, D., Pitarch, J.L., de Prada, C., 2017. Modifier Adaptation methodology based on transient and static measurements for RTO to cope with structural uncertainty. Comput. Chem. Eng. 106, 480–500. https://doi.org/10.1016/j.compchemeng.2017.07.001

Shah, G., Engell, S., 2011. Tuning MPC for desired closed-loop performance for MIMO systems, in: Proceedings of the 2011 American Control Conference. IEEE, pp. 4404–4409. https://doi.org/10.1109/ACC.2011.5991581

Vaccari, M., Pannocchia, G., 2018. Implementation of an economic MPC with robustly optimal steady-state behavior. IFAC-PapersOnLine 51, 92–97. https://doi.org/10.1016/j.ifacol.2018.10.180

Vaccari, M., Pannocchia, G., 2016. A Modifier-Adaptation Strategy towards Offset-Free Economic MPC. Processes 5, 2. https://doi.org/10.3390/pr5010002

Vaccari, M., Pelagagge, F., Bonvin, D., Pannocchia, G., 2020. Estimation technique for offset-free economic MPC based on modifier adaptation, in: IFAC 2020 - 21st IFAC World Congress.

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

Williams, T.J., Otto, R.E., 1960. A generalized chemical processing model for the investigation of computer control. Trans. Am. Inst. Electr. Eng. Part I Commun. Electron. 79, 458–473. https://doi.org/10.1109/tce.1960.6367296

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