Modifier-Adaptation Methodology for Real-Time Optimization Reformulated as a Nested Optimization Problem

D. Navia[a], L. Briceño[b] G. Gutiérrez[c], C. de Prada[c]

[a] *Departamento de Ingeniería Química y Ambiental, Universidad Técnica Federico Santa María, Chile. daniel.navia@usm.cl*

[b] *Departamento de Matemática, Universidad Técnica Federico Santa María, Chile. luis.briceno@usm.cl*

[c] *Systems Engineering and Automatic Control Department, University of Valladolid, Spain. prada@autom.uva.es*

Abstract

The following work presents a reformulation of the modifier adaptation methodology for real-time optimization as a nested optimization problem. Using the idea of iteration over the modifiers, this method makes it possible to find a point that satisfies the necessary conditions of optimality (NCO) of the process, despite modeling mismatch, using an outer optimization layer that updates the gradient modifiers with the objective of minimizing the Lagrangian function estimation of the process. Moreover, if a direct search algorithm is implemented in this layer, we can find the optimum without explicitly computing the gradients of the process. The presented scheme was tested in three optimization examples, assuming the absence and presence of process noise, with parametric and structural uncertainty. The results show that in all the cases studied, the method converges to a close neighborhood of a point that satisfies the NCO of the real plant, being robust under noisy scenarios and without the need to estimate process derivatives.

1. Introduction

Finding the optimal operation point of an industrial process is not a trivial task because of the inherent difficulties of the process itself and the uncertainties and disturbances that continuously modify the operating conditions. In highly automated plants, optimal operation is typically addressed by a decision hierarchy involving several levels that include scheduling, real-time operation (RTO) and process control. At the RTO level, medium-term decisions are made on a time scale of hours to a few days, explicitly considering economic objectives. This step typically consists of an optimizer that determines the optimal settings under slowly changing conditions [1], and hence, steady-state assumptions are considered in the optimization.

RTO emerged in the late 1970s as a two-stage algorithm combining parameter estimation and economic optimization. When the process reaches the steady state, the uncertain parameters of a nonlinear model are updated to solve a parameter estimation problem using the actual process measurements. After the model update, economic optimization is performed, and a new stationary point is found [2]. The new operating point is then applied to the process in an iterative scheme, seeking an optimal operating point.

Because of the inherent uncertainty, which is translated in the difference between the model and reality (modeling mismatch), and the fact that the two steps are separately solved, the two-step

algorithm by itself will not necessarily converge to the optimum of the process[3]. Roberts explicitly considered the interaction of these steps, and as a result of his work, an additional parameter was added to the economic optimization, defined as the difference between the gradient of the cost function of the process and the model: the ISOPE algorithm [3]. Almost two decades later, Piotr Tatjewsky demonstrated that the convergence properties of ISOPE did not depend on the parameter estimation problem but only on the fact that the process measurements must be equal to the predictions of the model at the actual point, replacing the complete estimation problem with a bias term [4]. Later, Gao and Engell presented a method to cope with optimizations with inequality constraints on the dependent variables, considering the difference in the gradient of the constraints between the process and the model [5]. Recently, the use of terms that modify the original model was generalized in the work of Chachuat and co-workers [6]. In this work, the authors define the "Modifier-Adaptation Methodology" as a general method to describe and employ the modifiers of the gradients for the cost and the inequality constraints, as well as the difference between the constraint values, to reach a point that satisfies the NCO of the process and, under the constraint qualification assumption, corresponds to a local optimum of the real plant. Later, Marchetti and co-workers [7-9] extended the methodology to include filters to smooth its evolution and describe the use of a dual approach to estimate the gradient modifiers. Notice that the use of the modifier adaptation approach requires that enough process measurements be available to compute the value of the economic cost function and the constraints.

Although previous methods can find a point that satisfies the KKT conditions of the plant, it is mandatory to estimate the real gradients of the system, which is not a trivial task. In this work, we have reformulated the modifier-adaptation methodology as a nested optimization procedure, changing the gradient estimation step with an upper optimization layer where the modifiers are defined as the decision variables, which are updated by a function measured from the process. With this configuration, we can avoid estimating the process derivatives, provided a gradient-free algorithm is implemented in the upper optimization.

The paper is organized as follows. Section two presents a summary of the modifier-adaptation methodology and a review of different methods to estimate the process derivatives, to contextualize this work. Section three describes the proposed methodology and its implementation. Section four describes the formalization of the method proposed. Section five discusses three examples of applications of the method proposed, as well as providing a comparison with the original methodology and the sensitivity of both under noise in process measurements. Finally, section six presents some concluding remarks.

2. Modified Adaptation Methodology

The idea of exploring the use of modifiers in process optimization is founded in the work of Biegler and co-workers, who proposed a methodology to optimize a complex model using a simplified one, matching the necessary conditions of optimality of both models [10]. Later, Forbes and Marlin applied this idea to process optimization, using the real process instead of a complex model [11, 12].

The problem of finding the optimal point of operation for a real system can be stated as solving problem (1). Below, the subscript "*p*" represents a magnitude computed from the process, while

values without the subscript refer to the model or to the manipulated variables that are common to the process and the model.

$$\begin{aligned} &\min_{\mathbf{u}} \Phi_p(\mathbf{u}) \\ &s.t.: \\ &\mathbf{G}_p(\mathbf{u}) \le 0 \end{aligned} \tag{1}$$

Here, $\mathbf{u} \in R^{n_u}$ are the manipulated variables of the process (decision variables for the economic optimization), $\Phi_P \in R$ is the economic performance of the process, and $\mathbf{G}_P \in R^{n_g}$ are the process constraints, both computed from the available measurements. Unfortunately, the exact description of the process is unknown, and only a model that represents the reality is available for use in the process optimization. The model-based economic optimization is represented by problem (2), where Φ and $\mathbf{G}$ represent the economic cost and constraints of the model, which, in general, depend on some uncertain parameters $\boldsymbol{\theta} \in R^{n_\theta}$.

$$\begin{aligned} &\min_{\mathbf{u}} \Phi(\mathbf{u}, \boldsymbol{\theta}) \\ &s.t.: \\ &\mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) \le 0 \end{aligned} \tag{2}$$

An open loop implementation of the solution of problem (2) will lead to suboptimal and even infeasible operation in the process due to the model-process mismatch. The modifier adaptation methodology changes problem (2) so that in a closed-loop execution, the necessary conditions of optimality of the modified problem correspond to the necessary conditions of the process, upon convergence of the algorithm. Problem (3) shows the model-based optimization with additional modifiers defined in equation (4).

$$\begin{aligned} &\min_{\mathbf{u}} \Phi_M := \Phi(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\lambda}_k^T \mathbf{u} \\ &s.t.: \\ &\mathbf{G}_M := \mathbf{G}(\mathbf{u}, \boldsymbol{\theta}) + \boldsymbol{\gamma}_k^T (\mathbf{u} - \mathbf{u}_k) + \boldsymbol{\varepsilon}_k \le 0 \end{aligned} \tag{3}$$

$$\begin{aligned} \boldsymbol{\lambda}_k^T &:= \nabla_{\mathbf{u}} \boldsymbol{\Phi}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}} \boldsymbol{\Phi}(\mathbf{u}_k, \boldsymbol{\theta}) \\ \boldsymbol{\gamma}_k^T &:= \nabla_{\mathbf{u}} \mathbf{G}_p(\mathbf{u}_k) - \nabla_{\mathbf{u}} \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}) \\ \boldsymbol{\varepsilon}_k &:= \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \boldsymbol{\theta}) \end{aligned} \tag{4}$$

Here, $\mathbf{u}_k \in R^{n_u}$ represents the actual operation point, calculated at the previous RTO iteration, and the modifiers $\boldsymbol{\lambda}_k \in R^{n_u}$, $\boldsymbol{\gamma}_k \in R^{n_u \times n_g}$ $\boldsymbol{\gamma}_k$ and $\boldsymbol{\varepsilon}_k \in R^{n_g}$ are evaluated using the information available at this point. Figure 1 summarizes the implementation of the modifier-adaptation methodology. In the figure, $\mathbf{u}_{k+1}$ is the solution of problem (3).

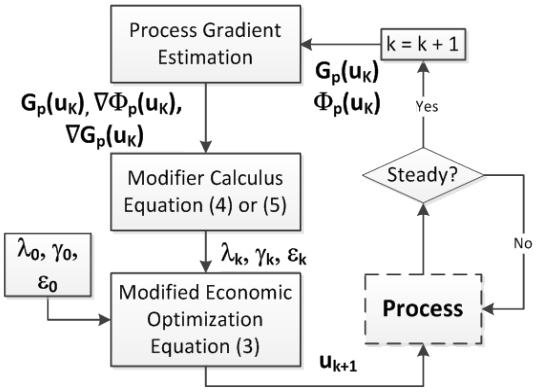

Figure 1. Implementation of modifier-adaptation methodology.

Marchetti and co-workers demonstrated that under the assumption of convergence of the iterative scheme, the KKT conditions of the modified model match the ones of the real process [7, 8], which implies that if second-order conditions hold at this point, then the modified problem makes it possible to find a local optimum of the real plant. The authors also recommend the use of a filtering procedure of the modifiers to improve the convergence. Equation (5) summarizes the filtering step, where $\mathbf{K}_\lambda$, $\mathbf{K}_\gamma$ and $\mathbf{K}_\varepsilon$ represent the respective first-order filter constants for each modifier, and sub index *"k"* indicates that the variables are evaluated in the k^{th} RTO iteration using $\mathbf{u}_k$. An alternative to this step can be found in the work of Bunin and co-workers [13], where the authors filter the inputs to be applied to the process to look for feasibility.

$$\begin{aligned}\boldsymbol{\lambda}_k^T &= (\mathbf{I}-\mathbf{K}_\lambda)\boldsymbol{\lambda}_{k-1}^T + \mathbf{K}_\lambda\left(\nabla_u\boldsymbol{\Phi}_p(\mathbf{u}_k) - \nabla_u\boldsymbol{\Phi}(\mathbf{u}_k,\boldsymbol{\theta})\right)\\ \boldsymbol{\gamma}_k^T &= (\mathbf{I}-\mathbf{K}_\gamma)\boldsymbol{\gamma}_{k-1}^T + \mathbf{K}_\gamma\left(\nabla_u\mathbf{G}_p(\mathbf{u}_k) - \nabla_u\mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta})\right)\\ \boldsymbol{\varepsilon}_k &= (\mathbf{I}-\mathbf{K}_\varepsilon)\boldsymbol{\varepsilon}_{k-1} + \mathbf{K}_\varepsilon\left(\mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k,\boldsymbol{\theta})\right)\end{aligned} \qquad (5)$$

The process gradient estimation stage from Figure 1 is the key issue of the modifier-adaptation method. In chapter 4 of the book of Brdys and Tatjewski [14], there is an interesting discussion regarding the estimation of the process mapping derivatives. The authors justify the importance of the method to estimate the experimental gradients based on the fact that the time spent in their evaluation may be significantly higher than the whole remaining measurement and calculation during each iteration. The authors, as well as Mansour and Ellis [15], group the available alternatives according to the type of information used to estimate the gradients in steady-state or dynamic approaches.

Regarding the use of steady-state methods, two alternatives are listed:

Finite Differences [3]: this approach consists of perturbing the plant around the operation point as many times as it has decision variables, and then applying the finite difference definition of a derivative with the steady-state measurements from the process.

Dual control optimization [16]: this procedure uses the past measurements of the process to estimate the plant derivatives using a rank-1 update. To ensure an accurate estimation of the experimental derivatives, an additional constraint (the dual constraint) must be added to ensure sufficient energy

to the system. In the literature, directional derivatives [8] and the Broyden formula [17] have been used to update the process gradients with past measurements.

There have also been attempts to estimate the steady-state gradients of the process using dynamic measurements. The Dynamic Model Identification (DMI) approach identifies a local dynamic model with the measurements obtained during the transient between two RTO iterations.

The idea of using dynamic data to estimate the gradients seems to be very promising, as it might not require additionally perturbing the system to estimate the gradients, as steady-state methods do. However, severe difficulties can be found in the identification step that can seriously compromise the correct estimation of process derivatives. To obtain sufficiently accurate identification results, the plant must be sufficiently excited, as it can hardly be assumed that a passive identification experiment based on the measures recorded in the transient between two RTO iterations can always deliver sufficiently rich data. Hence, it is necessary to plan an active identification experiment around the actual operating point, adding additional dynamic perturbations into the process, thus falling into the same paradigm as the perturbation methods based on steady-state measurements [14]. Recently, Francois and Bonvin have presented a new methodology to estimate plant gradients using the transient between RTO executions, combining the techniques to identify dynamic gradients from gradient control with MA methods [18].

Alternatively, the methods based on static measurements could be a reliable alternative to obtain an adequate mapping of the process gradients, as they use steady-state measurements from the system. However, the finite differences approach can be impractical because it requires as many perturbations as there are decision variables. Otherwise, the dual approach uses past measurements to estimate the process curvature, decreasing the number of RTO iterations to converge into a stationary point, proposing an operating point that looks for both optimality and gradient estimation. Along the same line, an approach has been presented that exploits the assumption of local quasiconvexity (or quasiconcavity) of the process to accurately estimate the process gradient by bounding the experimental derivatives and solving a parameter estimation problem [19]. An extension of this work has been presented by Bunin and co-workers, proposing a methodology to bound the process gradients using regularizing structures and a statistical procedure based on the maximum likelihood criterion [20]. Recently, Gao and co-workers presented a new methodology that combines the advantages of derivative-free optimization with the convergence properties of the modifier-adaptation method [21, 22]. In this approach, the mapping of the process is obtained through a methodology of screening previous points, followed by solving a regression problem. This mapping is then used to calculate the process derivatives. In the same line, but with the objective of estimating an expected loss in the Lagrangian function of the process due to an inaccurate estimation of the experimental gradients, Marchetti has presented a reformulation of the modifier adaptation algorithm directly using a first-order estimation for the model and the process with the actual operating point and previous n_u measurements [9].

Furthermore, there are other methods to estimate experimental gradients that have been implemented in extremum-seeking methodology, based on the use of dithering signals or models (under the assumption of parametric uncertainty) [23, 24], but they have not been implemented in the context of the modifier-adaptation method.

As can be noted, currently, there is no clear answer about the best way to estimate the experimental derivatives because in almost all of the approaches, the accuracy of the estimation is confronted with the number of iterations, more precisely, in the case of the dual optimization and the quasiconvexity approach, with the size of the feasible region. For this reason, there have been several attempts to find new alternatives to improve this step [25], and it is also an important motivation to present the proposed reformulation of the modifier-adaptation method with a gradient-free alternative.

In this work, we have used the dual approach to compare the actual modifier-adaptation with our proposed reformulation, as it uses steady-state past measurements to estimate the process derivatives, implying a decrease in the number of iterations to converge to a KKT point of the process.

The dual methodology based on directional derivatives calculates the gradients of the process, assuming that the last n_u operational points are close enough to approximate a differential change as a finite difference [14] (equation (6)):

$$\begin{aligned}
&\nabla\boldsymbol{\Phi}_p(\mathbf{u}_k) \approx [\mathbf{S}_k]^{-1}\boldsymbol{\Delta\Phi}_{p_k}\\
&\nabla\mathbf{G}_p{}^j(\mathbf{u}_k) \approx [\mathbf{S}_k]^{-1}\boldsymbol{\Delta}\mathbf{G}_{p_k}{}^j,\ j=1\ldots n_g,\\
&\mathbf{S}_k := \left[\mathbf{u}_{k-1}-\mathbf{u}_{k-1-1},\ldots,\mathbf{u}_{k-1}-\mathbf{u}_{k-1-n_u}\right]^T\\
&\boldsymbol{\Delta\Phi}_{p_k} := \left[\Phi_p(\mathbf{u}_k)-\Phi_p(\mathbf{u}_{k-1}),,\ldots,\Phi_p(\mathbf{u}_k)-\Phi_p(\mathbf{u}_{k-n_u})\right]^T\\
&\boldsymbol{\Delta}\mathbf{G}_{p_k}{}^j = \left[G_p{}^j(\mathbf{u}_k)-G_p{}^j(\mathbf{u}_{k-1}),\ldots,G_p{}^j(\mathbf{u}_k)-G_p{}^j(\mathbf{u}_{k-n_u})\right]^T,\ j\in 1\ldots n_g
\end{aligned} \tag{6}$$

where the rows of $\mathbf{S}_k \in R^{n_u\times n_u}$, $\boldsymbol{\Delta\Phi}_{pk} \in R^{n_u}$ and $\boldsymbol{\Delta}\mathbf{G}^j_{pk} \in R^{n_u}$, $j = 1\ldots n_g$, are computed as the differences of the current values of the corresponding magnitude and the ones taken in the n_u previous steps. The complete details of the relation between directional derivatives and the process gradients in the context of RTO can be found in [14, 16]. The inversion of the $\mathbf{S}_k$ matrix implies that to accurately estimate the gradient and without the amplification of the process noise, the inverse of its condition number must be always greater than the lower bound. If the inverse of the condition number is defined as δ, the dual modifier-adaptation methodology is summarized in problem (7), where δ^L is the lower bound of δ.

$$\begin{aligned}
&\min_u \Phi(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\lambda}_k^T u\\
&s.t.:\\
&\mathbf{G}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}_k^T(\mathbf{u}-\mathbf{u}_k) + \boldsymbol{\varepsilon}_k \le 0\\
&\delta(\mathbf{S}_k) \ge \delta^L
\end{aligned} \tag{7}$$

The addition of the dual constraint in problem (7) may require a different computation policy than in problem (3) to guarantee a good estimation of the process gradients, changing the direction of the iterations as well as increasing their number. With respect to the feasible region of problem (7), it is clear that the addition of the dual constraint produces an additional non-convexity in the optimization, as the next operating point must not belong to the hyperplane formed with $\mathbf{u}_k \ldots \mathbf{u}_{k-nu-1}$. This drawback can be relaxed by splitting the problem in two, finding the value of $\mathbf{u}_{k+1}$ in both sides of the hyper plane solving two modified optimizations, and applying one solution to the process.

For convergence of the algorithm, necessary conditions have been reported in the work of Marchetti and co-workers using a linearization around a fixed point [7]. A generalization of the convergence properties of the algorithm has been proposed by Faulwasser and Bonvin, based on the results from fixed point theorem [26]. In their work, a sufficient condition for the convergence of modifier-adaptation methods with different orders is presented, provided that problem (3) is globally feasible and has a unique solution and that a first-order filter is applied between each RTO iteration [27].

3. Modified Adaptation Methodology as a Nested Optimization Problem

Notice that the iterative implementation of modifier adaptation methodology can be summarized as follows: starting with a given state of the plant, problem (3) is solved to calculate and apply the next operation point and re-compute the modifiers. This process is repeated until no further changes in the decision variables are observed. That is, the modifiers are continuously updated with the final goal of finding a stationary point that satisfies the KKT conditions of the process. Then, if sufficient second-order conditions hold, a local optimum of the process has been found upon convergence of the algorithm. With this idea in mind, one can iterate using the modifiers until the optimum is found but replace each gradient estimation and modifier calculus with one step of an optimization layer (called upper optimization layer) that uses the modifiers as decision variables. These modifiers are updated according to the value of a given performance function measured from the process with the goal of improving the cost function and satisfying the KKT conditions of the real system. This concept is behind the nested modifier-adaptation that will be explained below.

To be more precise on our methodology, we recall that the process optimization problem is given by

$$P_p: \quad \begin{array}{l} \min_{\mathbf{u}} \Phi_p(\mathbf{u}) \\ s.t.: \\ \mathbf{G}_p(\mathbf{u}) \le 0 \end{array} \tag{8}$$

and the associated Lagrangian is

$$L_p(\mathbf{u},\boldsymbol{\mu}) := \Phi_p(\mathbf{u}) + \boldsymbol{\mu}^T \mathbf{G}_p(\mathbf{u}). \tag{9}$$

Given a decision variable $\mathbf{u}_k$ obtained in the iteration $k > 0$ (eventually $k = \infty$) and for every $\boldsymbol{\Lambda} = (\boldsymbol{\gamma}, \boldsymbol{\lambda})$, we denote by $\mathbf{u}_k(\boldsymbol{\Lambda})$ and $\boldsymbol{\mu}_k(\boldsymbol{\Lambda})$ the solution and the corresponding multiplier associated with the modified optimization problem

$$P_{Mk}(\boldsymbol{\Lambda}): \quad \begin{array}{l} \min_{\mathbf{u}} \Phi_M(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) := \Phi(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\lambda}^T \mathbf{u} \\ s.t.: \\ G^i_{Mk}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) := G^i(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}^{iT}(\mathbf{u}-\mathbf{u}_k) + G^i_p(\mathbf{u}_k) - G^i(\mathbf{u}_k,\boldsymbol{\theta}) \le 0, \qquad \forall i = 1,...,n_g \end{array} \tag{10}$$

and we define the Lagrangian associated with problem (10) by

$$L_{Mk}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) := \Phi_M(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) + \boldsymbol{\mu}^T \mathbf{G}_{Mk}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \Phi_M(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) + \sum_{i=1}^{n_g} \mu^i G^i_{Mk}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}). \tag{11}$$

Note that we have included the upper and lower bounds of the decision variable $\mathbf{u}$ in the inequality constraint for simplicity. The main idea of the NMA is to solve the *upper optimization problem (12), as its stationary point corresponds to the KKT conditions of the process.*

$$\min_{\mathbf{\Lambda}} L_p(\mathbf{u}_\infty(\mathbf{\Lambda}), \boldsymbol{\mu}_\infty(\mathbf{\Lambda})) = \Phi_p\left(\mathbf{u}_\infty(\mathbf{\Lambda})\right) + \boldsymbol{\mu}_\infty(\mathbf{\Lambda})^T \mathbf{G}_p\left(\mathbf{u}_\infty(\mathbf{\Lambda})\right), \tag{12}$$

Where $\left(\mathbf{u}_\infty(\mathbf{\Lambda}), \boldsymbol{\mu}_\infty(\mathbf{\Lambda})\right)$ are obtained by solving $P_{M\infty}(\mathbf{\Lambda})$. However, as problem (12) is difficult to solve, as it is assumed that the functions describing the process are unknown, we approximate a solution iteratively by performing one minimization step of problem (12) by iteration, solving in each iteration $P_{Mk}(\mathbf{\Lambda})$, as shown in Algorithm 1.

ALGORITHM 1

STEP 0: Set $k = 0$, and initialize $\mathbf{u}_0$ and $\boldsymbol{\mu}_0$. These values could be obtained by solving problem (2) or by solving problem (10) estimating the modifiers using equation (4).

For every $k \geq 0$:

STEP 1: Given $\mathbf{U}_k := (\mathbf{u}_k, \boldsymbol{\mu}_k)$, apply $\mathbf{u}_k$ in the process, and wait until steady state to compute $\Phi_p(\mathbf{u}_k)$, $\mathbf{G}_p(\mathbf{u}_k)$, and an estimation of the Lagrangian of the process via $L_p(\mathbf{u}_k, \boldsymbol{\mu}_k) := \Phi_p(\mathbf{u}_k) + \boldsymbol{\mu}_k{}^T \mathbf{G}_p(\mathbf{u}_k)$.

STEP 2: Using $L_p(\mathbf{u}_k, \boldsymbol{\mu}_k)$ as a target to minimize, perform one step of the upper optimization to obtain the modifiers $\mathbf{\Lambda}_k = (\boldsymbol{\gamma}_k, \boldsymbol{\lambda}_k)$. The update policy of $\mathbf{\Lambda}_k$ depends on the algorithm implemented in the upper optimization layer. For example, if the Nelder-Mead algorithm is applied, then this step is equivalent to one update of the simplex.

STEP 3: Solve the modified optimization problem $P_{Mk}(\mathbf{\Lambda}_k)$ defined in problem (10) to obtain $\mathbf{u}_{k+1} = \mathbf{u}_k(\mathbf{\Lambda}_k)$ and $\boldsymbol{\mu}_{k+1} = \boldsymbol{\mu}_k(\mathbf{\Lambda}_k)$. Compare $\mathbf{u}_{k+1}$ with $\mathbf{u}_k$, and if a convergence criterion is satisfied, i.e., $\|\mathbf{u}_{k+1} - \mathbf{u}_k\|$ is less than a given tolerance, then stop. Otherwise, set $k = k + 1$, and go to Step 1.

The proposed methodology uses the upper optimization layer to generate values of the modifiers $\boldsymbol{\lambda}_k$ and $\boldsymbol{\gamma}_k$ for the MA optimization layer. These modifiers are used as decision variables to minimize an estimate of the Lagrangian function of the process in the upper optimization layer.

Figure 2 summarizes the nested modifier-adaptation methodology described, where the analogy can be seen in the implementation with respect to the original modifier-adaptation, noting that the process gradient estimation has been replaced by the upper optimization block. The name nested modifier-adaptation (NMA) arises from the fact that, at every iteration of the upper layer, a full optimization of the modified problem (10) is performed.

In addition to the filtering that the tuning of the upper optimization layer could provide, one could keep the filtering step from equation (4) as an option. Its use may allow convergence to the KKT point of the plant in a smoother way, avoiding possible infeasibilities that can appear in the evolution of the algorithm, but is not required by the method [7]. To implement the filter, the value of

$\boldsymbol{\Lambda}_k$ in $P_{Mk}(\boldsymbol{\Lambda}_k)$ should be an affine combination of the decision variable suggested by the upper layer and the value $\boldsymbol{\Lambda}_{k+1}$, as shown by equation (13).

$$\boldsymbol{\Lambda}_k^T := (\mathbf{I} - \mathbf{K}_\Lambda)\boldsymbol{\Lambda}_{k-1}^T + \mathbf{K}_\Lambda\left(\boldsymbol{\Lambda}_k^{UL^T}\right) \tag{13}$$

where $\boldsymbol{\Lambda}_k^{UL}$ is the value of the decision variable suggested by the upper layer at iteration *k*, and $\mathbf{K}_\Lambda$ is a constant matrix similar to $\mathbf{K}_\lambda$ and $\mathbf{K}_\gamma$ in (5). Alternatively, a filtering step in the decision variables can be applied in the same way as the original gradient-based algorithm [13].

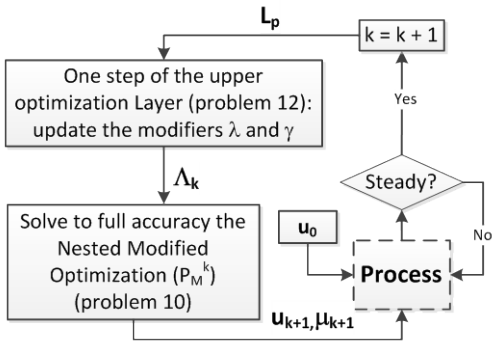

Figure 2: Proposed nested optimization

The purpose of the nested optimization is to obtain a stationary point that fulfills the KKT conditions of the process in iteration with the modifiers. When selecting the optimization method for the upper layer, we can consider the fact that the decision variables, the gradient modifiers, are not constrained, and the feasibility of the process constraints is considered in the modified optimization problem so that we can implement an unconstrained method. Additionally, if a direct search algorithm is used, then there is no need to estimate the gradient of the process to update the modifiers in step 2 of the algorithm. Accordingly, the Nelder-Mead algorithm has been chosen for this task. Notice that if one wishes to use a gradient-based method in the upper layer, then it is necessary to apply additional perturbations to the real system to estimate the gradient of the process in a way that is similar to the original modifier-adaptation methodology, and then, the main advantage of skipping the gradient estimation step disappears.

The Nelder-Mead algorithm works with the idea of finding the optimum of the process by exploring a cost surface through a geometric figure with $n_\lambda + n_\gamma + 1$ vertices: the simplex. Each vertex corresponds to a value of the decision variables and is associated with its corresponding value of the cost function. Then, with 4 basic operations, reflection, expansion, contraction and shrinking, the algorithm iterates using the set of decision variables to seek the optimum. The main reason to choose this algorithm is because it is particularly parsimonious in function evaluations per iteration, as in practice it typically requires only one or two function evaluations to construct a new iteration, while several popular direct search methods use n_u or more function evaluations to obtain a new simplex [28]. This property is very important considering that each function evaluation implies changing the operation point of the real process. The other reason this method is chosen to update

the modifiers is its popularity in the chemical engineering field for nonlinear optimizations when gradients are not available, e.g., in experimental design or the optimization of analytics tests [29].

In addition to the suppression of the gradient estimation step, it can be expected that the proposed methodology, applied with a direct search method in the upper layer, would be less sensitive to the process noise [29, 30]. This expectation is based on the good results reported for scenarios with inaccurate function evaluations using the gradient-free algorithms and on the fact that if the gradient has been estimated with insufficient excitation, the process noise would be amplified, producing a wrong update policy. It can also be noted that we could implement other strategies, such as evolutionary algorithms, that might behave better than the direct search methods under stochastic noise influence; however, the smaller number of points is critical in the nested approach because we are iterating on the real plant. Additionally, it can be noted that one of the most sensitive parameters in dual modifier-adaptation methodology is neglected: the size of the perturbation necessary to estimate the experimental gradient (e.g., δ^L), which increases the interest of the proposed method because it implies a simpler way to apply the proposed framework.

4. Formalization of the Methodology

The idea of the nested procedure is quite intuitive, as it seeks the KKT conditions of the process by iterating directly with the gradient modifiers. Note that at a stationary point, the gradient of the Lagrangian must vanish. For this reason, the Lagrangian computed with process measurements has been chosen as a target of the upper optimization layer. With this point in mind, in this section, the equivalence of the NMA with the MA is presented.

First, recall that for any $\boldsymbol{\Lambda}$, the *strict complementarity slackness* assumption in problem $P_{Mk}(\boldsymbol{\Lambda})$ at some point $(\mathbf{u},\boldsymbol{\mu})$ is given by $\mu^i > 0$ for every i in the set of active constraints $A_k(\mathbf{u}) = \{i = 1,\ldots,n_g : G^i_{Mk}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) = 0\}$ of problem $P_{Mk}(\boldsymbol{\Lambda})$, and we say that $\mathbf{u}$ *is regular* for problem $P_{Mk}(\boldsymbol{\Lambda})$ if the set of vectors $\{\nabla_{\mathbf{u}}\mathbf{G}^i_{Mk}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda})\}_{i\in A_k(\mathbf{u})}$ is linearly independent.

Theorem 4.1 (Optimality equivalence of the nested modifier adaptation method)

Suppose that Algorithm 1 converges to a stationary point $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty) = (\mathbf{u}_\infty(\boldsymbol{\Lambda}_\infty),\boldsymbol{\mu}_\infty(\boldsymbol{\Lambda}_\infty))$, *which is a solution to* $P_{M\infty}(\boldsymbol{\Lambda}_\infty)$ *with modifiers* $\boldsymbol{\Lambda}_\infty = (\boldsymbol{\gamma}_\infty,\boldsymbol{\lambda}_\infty)$. *Moreover, suppose that functions* $\Phi(\cdot,\boldsymbol{\theta})$ *and* $\mathbf{G}_p(\cdot,\boldsymbol{\theta})$ *are twice continuously differentiable and that the Lagrangian function* $L_p(\mathbf{u},\boldsymbol{\mu}) = \Phi_p(\mathbf{u}) + \boldsymbol{\mu}^T\mathbf{G}_p(\mathbf{u})$ *is continuously differentiable in a neighborhood of* $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)$. *In addition, assume the following:*

a) $\mathbf{u}_\infty$ *is regular for problem* $P_{M\infty}(\boldsymbol{\Lambda}_\infty)$.
b) Strict complementary slackness for problem $P_{M\infty}(\boldsymbol{\Lambda}_\infty)$ *holds at* $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)$.
c) $\nabla^2_{\mathbf{uu}}\mathbf{L}_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda})$ *is positive definite.*

Then, $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)$ *is a KKT point associated with the process problem (*P_p*).*

Proof. *Note that, for every* $\boldsymbol{\Lambda}$ *and any solution* $(\mathbf{u},\boldsymbol{\mu})$ *to problem* $P_{M\infty}(\boldsymbol{\Lambda}_\infty)$*, the first-order necessary optimality conditions yield*

$$\begin{aligned}
&\nabla_{\mathbf{u}}\mathbf{L}_{M\infty}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) = \nabla_{\mathbf{u}}\boldsymbol{\Phi}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\lambda} + \boldsymbol{\mu}^T\left(\nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}\right) = \mathbf{0}\\
&\mu^i G^i_{M\infty}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) = \mu^i\left(G^i(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}_i^T(\mathbf{u}-\mathbf{u}_\infty) + G_p^i(\mathbf{u}_\infty) - G^i(\mathbf{u}_\infty,\boldsymbol{\theta})\right) = 0, \qquad \forall i = 1,\ldots,n_g\\
&G^i(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}_i^T(\mathbf{u}-\mathbf{u}_\infty) + G_p^i(\mathbf{u}_\infty) - G^i(\mathbf{u}_\infty,\theta) \le 0, \mu^i \ge 0, \qquad \forall i = 1,\ldots,n_g.
\end{aligned} \tag{14}$$

In particular, as $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty) = (\mathbf{u}_\infty(\boldsymbol{\Lambda}_\infty),\boldsymbol{\mu}_\infty(\boldsymbol{\Lambda}_\infty))$ *is a solution to* $P_{M\infty}(\boldsymbol{\Lambda}_\infty)$*, we have*

$$\begin{aligned}
&\nabla_{\mathbf{u}}\mathbf{L}_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda}_\infty) = \nabla_{\mathbf{u}}\boldsymbol{\Phi}(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\lambda}_\infty + \boldsymbol{\mu}_\infty^T\left(\nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\gamma}_\infty\right) = \mathbf{0}\\
&\mu^i_\infty G^i_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\theta},\boldsymbol{\Lambda}_\infty) = \mu^i_\infty G^i_p(\mathbf{u}_\infty) = 0, \qquad \forall i = 1,\ldots,n_g\\
&G^i_p(\mathbf{u}_\infty) \le 0, \mu^i_\infty \ge 0, \qquad \forall i = 1,\ldots,n_g.
\end{aligned} \tag{15}$$

Note that we deduce from equation (15) that primal and dual feasibility and complementarity slackness hold for the process problem (P_p*). Now, we focus our attention on completing the equivalence of first-order optimality conditions by proving the stationarity of the Lagrangian of the process through the stationarity of the Lagrangian of the upper optimization problem achieved at* $(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty)$*. To this end, we split the proof into three parts. First, we prove the differentiability of functions* $\mathbf{u}_\infty(\cdot)$ *and* $\boldsymbol{\mu}_\infty(\cdot)$*. Second, we obtain the result that the null space of matrix* $\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_\infty(\boldsymbol{\Lambda}_\infty)$ *is zero, and finally, we provide the desired equivalence.*

(i) Differentiability of $\mathbf{u}_\infty(\cdot)$ ***and*** $\boldsymbol{\mu}_\infty(\cdot)$ ***in a neighborhood of*** $\boldsymbol{\Lambda}_\infty$ **.**

Denoting by $\mathbf{U} = (\mathbf{u},\boldsymbol{\mu})$ *and*

$$\mathbf{F}_\infty(\mathbf{U},\boldsymbol{\Lambda}) = \begin{bmatrix} \nabla_{\mathbf{u}}\mathbf{L}_{M\infty}(\mathbf{u},\boldsymbol{\mu},\boldsymbol{\Lambda}) \\ \mu^1 G^1_{M\infty}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) \\ \vdots \\ \mu^{n_g} G^{n_g}_{M\infty}(\mathbf{u},\boldsymbol{\theta},\boldsymbol{\Lambda}) \end{bmatrix} = \begin{bmatrix} \nabla_{\mathbf{u}}\boldsymbol{\Phi}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\lambda} + \boldsymbol{\mu}^T\left(\nabla_{\mathbf{u}}\mathbf{G}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}\right) \\ \mu^1\left(G^1(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}_1^T(\mathbf{u}-\mathbf{u}_\infty) + G_p^1(\mathbf{u}_\infty) - G^1(\mathbf{u}_\infty,\boldsymbol{\theta})\right) \\ \vdots \\ \mu^{n_g}\left(G^{n_g}(\mathbf{u},\boldsymbol{\theta}) + \boldsymbol{\gamma}_{n_g}^T(\mathbf{u}-\mathbf{u}_\infty) + G_p^{n_g}(\mathbf{u}_\infty) - G^{n_g}(\mathbf{u}_\infty,\boldsymbol{\theta})\right) \end{bmatrix},$$

we have from equation (15) that $\mathbf{F}_\infty(\mathbf{U}_\infty,\boldsymbol{\Lambda}_\infty) = 0$*, and, as* $G^i_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda}_\infty) = G^i_p(\mathbf{u}_\infty)$*, the Jacobian matrix of* $\mathbf{F}_\infty$ *with respect to* $\mathbf{U}$*, evaluated at* $(\mathbf{U}_\infty,\boldsymbol{\Lambda}_\infty) = 0$*, is given by*

$$\mathbf{M}_\infty(\mathbf{U}_\infty,\boldsymbol{\Lambda}_\infty) = \begin{bmatrix} \nabla^2_{\mathbf{uu}}\boldsymbol{\Phi}(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\mu}_\infty^T\nabla^2_{\mathbf{uu}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta}) & \nabla_{\mathbf{u}}\mathbf{G}^1(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\gamma}^1_\infty & \cdots & \nabla_{\mathbf{u}}\mathbf{G}^{n_g}(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\gamma}^{n_g}_\infty \\ \mu^1_\infty\left(\nabla_{\mathbf{u}}\mathbf{G}^1(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\gamma}^1_\infty\right)^T & G^1_p(\mathbf{u}_\infty) & 0 & 0 \\ \vdots & 0 & \ddots & 0 \\ \mu^{n_g}_\infty\left(\nabla_{\mathbf{u}}\mathbf{G}^{n_g}(\mathbf{u}_\infty,\boldsymbol{\theta}) + \boldsymbol{\gamma}^{n_g}_\infty\right)^T & 0 & 0 & G^{n_g}_p(\mathbf{u}_\infty) \end{bmatrix}$$

Let us prove that $\mathbf{M}_\infty(\mathbf{U}_\infty,\boldsymbol{\Lambda}_\infty)$ *is invertible. Indeed, let* $\mathbf{V} = (\mathbf{v},\boldsymbol{\eta})$ *be any vector in* $R^{n_u \times n_g}$*, and suppose that*

$$\mathbf{M}_\infty(\mathbf{U}_\infty,\boldsymbol{\Lambda}_\infty)\mathbf{V}=\begin{bmatrix}\left[\nabla^2_{\mathbf{uu}}\boldsymbol{\Phi}(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\mu}_\infty^T\nabla^2_{\mathbf{uu}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta})\right]\mathbf{v}+\sum_{i=1}^{n_g}\eta_i\left[\nabla_{\mathbf{u}}\mathbf{G}^i(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\gamma}_\infty^i\right]\\ \mu_\infty^1\left(\nabla_{\mathbf{u}}\mathbf{G}^1(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\gamma}_\infty^1\right)^T\mathbf{v}+G_p^1(\mathbf{u}_\infty)\eta_1\\ \vdots\\ \mu_\infty^{n_g}\left(\nabla_{\mathbf{u}}\mathbf{G}^{n_g}(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\gamma}_\infty^{n_g}\right)^T\mathbf{v}+G_p^{n_g}(\mathbf{u}_\infty)\eta_{n_g}\end{bmatrix}=\mathbf{0}. \tag{16}$$

Notice that as $G^i_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda}_\infty)=G^i_p(\mathbf{u}_\infty)$*, the set of active constraints of the problem (*P_p*) at* $\mathbf{u}_\infty$ *and* $A_\infty(\mathbf{u}_\infty)$ *coincide. Then, we deduce from equation (16) and assumption b) that for every* $i\in A_\infty(\mathbf{u}_\infty)$*,* $\left(\nabla_{\mathbf{u}}\mathbf{G}^i(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\gamma}_\infty^i\right)^T\mathbf{v}=\mathbf{0}$ *and for every* $i\notin A_\infty(\mathbf{u}_\infty)$*,* $\eta^i=0$ *, which yields*

$$\left[\nabla^2_{\mathbf{uu}}\boldsymbol{\Phi}(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\mu}_\infty^T\nabla^2_{\mathbf{uu}}\mathbf{G}(\mathbf{u},\boldsymbol{\theta})\right]\mathbf{v}+\sum_{i\in A_\infty(\mathbf{u}_\infty)}\eta^i\left[\nabla_{\mathbf{u}}\mathbf{G}^i(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\gamma}_\infty^i\right]=\mathbf{0} \tag{17}$$

Premultiplying equation (17) by $\mathbf{v}^T$*, we obtain* $\mathbf{v}^T\left[\nabla^2_{\mathbf{uu}}\boldsymbol{\Phi}(\mathbf{u}_\infty,\boldsymbol{\theta})+\boldsymbol{\mu}_\infty{}^T\nabla^2_{\mathbf{uu}}\mathbf{G}(\mathbf{u}_\infty,\boldsymbol{\theta})\right]\mathbf{v}$*, and assumption c) yields* $\mathbf{v}=0$*. Hence, from equation (16) and assumption a), we obtain* $\boldsymbol{\eta}=0$*, and, therefore,* $\mathbf{V}=\mathbf{0}$*, implying that the Jacobian matrix is invertible. The implicit function theorem provides a neighborhood of* $\boldsymbol{\Lambda}_\infty$ *and the existence of a differentiable function* $\mathbf{U}_\infty(\cdot)=(\mathbf{u}_\infty(\cdot),\boldsymbol{\mu}_\infty(\cdot))$ *such that* $\mathbf{F}_\infty(\mathbf{U}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda}_\infty)=0$ *for every* $\boldsymbol{\Lambda}$ *in that neighborhood. Assumption c) ensures that, for every* $\boldsymbol{\Lambda}$ *near to* $\boldsymbol{\Lambda}_\infty$*,* $(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\mu}_\infty(\boldsymbol{\Lambda}))$ *is a solution to* $P_{M\infty}(\boldsymbol{\Lambda})$ *because sufficient optimality conditions hold.*

(ii) Null space of matrix $\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_\infty(\boldsymbol{\Lambda}_\infty)$.

Now, let us prove that the null space of matrix $\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_\infty(\boldsymbol{\Lambda}_\infty)=\left[\nabla_{\boldsymbol{\Lambda}}\mathbf{u}^1_\infty(\boldsymbol{\Lambda}_\infty)\ldots\nabla_{\boldsymbol{\Lambda}}\mathbf{u}^{n_u}_\infty(\boldsymbol{\Lambda}_\infty)\right]$ *is zero. Denote by* n_A *the number of active constraints* $(0\le n_A\le n_g)$ *for problem* $P_{M\infty}(\boldsymbol{\Lambda})$ *and, without loss of generality, assume that* $G^i_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda}_\infty)=0$ *for every* $1\le i\le n_A$ *and* $G^i_{M\infty}(\mathbf{u}_\infty,\boldsymbol{\mu}_\infty,\boldsymbol{\Lambda}_\infty)<0$ *for any* $i>n_A$*. Then,* $\mu^i_\infty>0$ *for every* $1\le i\le n_A$*,* $\mu^i_\infty=0$ *for every* $i>n_A$*, and from assumption b) and the smoothness of functions* $\mathbf{G}_{\mathbf{M}}(\cdot)$*,* $\mathbf{u}_\infty(\cdot)$*, and* $\boldsymbol{\mu}_\infty(\cdot)$*, we have* $\nabla_{\boldsymbol{\Lambda}}\boldsymbol{\mu}^i_\infty(\boldsymbol{\Lambda})=0$*, for any* $i>n_A$*, and* $\nabla_{\boldsymbol{\Lambda}}\mathbf{G}^i_{M\infty}(\mathbf{u}(\boldsymbol{\Lambda}),\boldsymbol{\mu}(\boldsymbol{\Lambda}),\boldsymbol{\Lambda})=0$ *for every* $1\le i\le n_A$*, for any* $\boldsymbol{\Lambda}$ *close enough to* $\boldsymbol{\Lambda}_\infty$*. Therefore, by defining the strictly positive vector* $\boldsymbol{\mu}^A_\infty(\boldsymbol{\Lambda})=\left(\boldsymbol{\mu}^1_\infty(\boldsymbol{\Lambda}),\ldots,\boldsymbol{\mu}^{n_A}_\infty(\boldsymbol{\Lambda})\right)$ *and* $\mathbf{U}^A_\infty(\boldsymbol{\Lambda})=\left(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\mu}^A_\infty(\boldsymbol{\Lambda})\right)$ *we obtain from equation (14) that*

$$\nabla_{\mathbf{u}}\mathbf{L}_{M\infty}(\mathbf{U}^A_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda})=\nabla_{\mathbf{u}}\boldsymbol{\Phi}(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\theta})+\boldsymbol{\lambda}+\sum_{i\in A^\infty(\mathbf{u}_\infty)}\mu^i_\infty(\boldsymbol{\Lambda})\left(\nabla_{\mathbf{u}}\mathbf{G}^i(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\theta})+\boldsymbol{\gamma}^i\right)=\mathbf{0}$$

$$\nabla_{\boldsymbol{\mu}^i}\mathbf{L}_{M\infty}(\mathbf{U}^A_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda})=G^i_{M\infty}(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\theta},\boldsymbol{\Lambda})=G^i(\mathbf{u}_\infty(\boldsymbol{\Lambda}),\boldsymbol{\theta})+\boldsymbol{\gamma}^{iT}(\mathbf{u}_\infty(\boldsymbol{\Lambda})-\mathbf{u}_\infty)+G^i_p(\mathbf{u}_\infty)-G^i(\mathbf{u}_\infty,\boldsymbol{\theta})=0,\quad\forall i=1,\ldots,n_A$$

for every $\boldsymbol{\Lambda}$ *close to* $\boldsymbol{\Lambda}_\infty$*, or, equivalently,* $\nabla_{\mathbf{U}^A_\infty}\mathbf{L}_{M\infty}\left(\mathbf{U}^A_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda}\right)=0$*. By deriving with respect to* $\boldsymbol{\Lambda}$ *and applying the chain rule in the last equation, we obtain*

$$\nabla_{\boldsymbol{\Lambda}}\mathbf{U}^A_\infty(\boldsymbol{\Lambda})\mathbf{M}^A_\infty(\mathbf{U}^A_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda})+\mathbf{N}^A_\infty(\mathbf{U}^A_\infty(\boldsymbol{\Lambda}),\boldsymbol{\Lambda})=\mathbf{0}, \tag{17}$$

where $\nabla_{\boldsymbol{\Lambda}}\mathbf{U}^A_\infty(\boldsymbol{\Lambda})=[\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_\infty(\boldsymbol{\Lambda}),\nabla_{\boldsymbol{\Lambda}}\boldsymbol{\mu}^A_\infty(\boldsymbol{\Lambda})]\in R^{(n_u+n_gn_u)\times(n_u+n_A)}$,

$$\mathbf{M}_{\infty}^{A}(\mathbf{U}_{\infty}^{A}(\boldsymbol{\Lambda}),\boldsymbol{\Lambda}) = \begin{bmatrix} \nabla_{\mathbf{uu}}^{2}\boldsymbol{\Phi}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) + \sum_{i\in A^{\infty}(\mathbf{u}_{\infty})} \mu_{\infty}^{i}(\boldsymbol{\Lambda})\nabla_{\mathbf{uu}}^{2}\mathbf{G}^{i}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) & \nabla_{\mathbf{u}}\mathbf{G}^{1}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) + \boldsymbol{\gamma}_{\infty}^{1} & \cdots & \nabla_{\mathbf{u}}\mathbf{G}^{n_A}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) + \boldsymbol{\gamma}_{\infty}^{n_A} \\ \left(\nabla_{\mathbf{u}}\mathbf{G}^{1}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) + \boldsymbol{\gamma}_{\infty}^{1}\right)^{T} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \left(\nabla_{\mathbf{u}}\mathbf{G}^{n_A}(\mathbf{u}_{\infty}(\boldsymbol{\Lambda}),\boldsymbol{\theta}) + \boldsymbol{\gamma}_{\infty}^{n_A}\right)^{T} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} \text{ and}$$

$$\mathbf{N}_{\infty}^{A}(\mathbf{U}_{\infty}^{A}(\boldsymbol{\Lambda}),\boldsymbol{\Lambda}) = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mu_{\infty}^{1}(\boldsymbol{\Lambda})\mathbf{I} & \mathbf{u}_{\infty}(\boldsymbol{\Lambda}) - \mathbf{u}_{\infty} & \mathbf{0} & \cdots & \mathbf{0} \\ \mu_{\infty}^{2}(\boldsymbol{\Lambda})\mathbf{I} & \mathbf{0} & \mathbf{u}_{\infty}(\boldsymbol{\Lambda}) - \mathbf{u}_{\infty} & & \vdots \\ \vdots & \vdots & & \ddots & \mathbf{0} \\ \mu_{\infty}^{n_A}(\boldsymbol{\Lambda})\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{u}_{\infty}(\boldsymbol{\Lambda}) - \mathbf{u}_{\infty} \end{bmatrix}.$$

As before, under assumptions b) and c), the matrix $\mathbf{M}_{\infty}^{A}\left(\mathbf{U}_{\infty}^{\mathbf{A}}(\boldsymbol{\Lambda}_{\infty}),\boldsymbol{\Lambda}_{\infty}\right)$ *is invertible, and moreover, the elements* $\mathbf{u}_{\infty}(\boldsymbol{\Lambda}_{\infty}) - \mathbf{u}_{\infty}$ *of* $\mathbf{N}_{\infty}^{A}\left(\mathbf{U}_{\infty}^{\mathbf{A}}(\boldsymbol{\Lambda}_{\infty}),\boldsymbol{\Lambda}_{\infty}\right)$ *are zero. Hence,*

$$\nabla_{\boldsymbol{\Lambda}}\mathbf{U}_{\infty}^{A}(\boldsymbol{\Lambda}_{\infty}) = -\mathbf{N}_{\infty}^{A}(\mathbf{U}_{\infty}^{A}(\boldsymbol{\Lambda}_{\infty}),\boldsymbol{\Lambda}_{\infty})[\mathbf{M}_{\infty}^{A}(\mathbf{U}_{\infty}^{A}(\boldsymbol{\Lambda}_{\infty}),\boldsymbol{\Lambda}_{\infty})]^{-1} = -\mathbf{N}_{\infty}^{A}(\mathbf{u}_{\infty},\boldsymbol{\mu}_{\infty},\boldsymbol{\Lambda}_{\infty})\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

$$= -\left[\mathbf{A}_{11}\begin{bmatrix} \mathbf{I} \\ \mu_{\infty}^{1}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \\ \vdots \\ \mu_{\infty}^{n_A}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \end{bmatrix}, \mathbf{A}_{12}\begin{bmatrix} \mathbf{I} \\ \mu_{\infty}^{1}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \\ \vdots \\ \mu_{\infty}^{n_A}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \end{bmatrix}\right].$$

where $\mathbf{A_{11}}$, $\mathbf{A_{12}}$, $\mathbf{A_{21}}$, *and* $\mathbf{A_{22}}$ *are appropriate matrices in* $R^{n_u \times n_u}$, $R^{n_u \times n_A}$, $R^{n_A \times n_u}$, *and* $R^{n_A \times n_A}$, *respectively. In particular, we have*

$$\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_{\infty}(\boldsymbol{\Lambda}_{\infty}) = -\mathbf{A}_{11}\begin{bmatrix} \mathbf{I} \\ \mu_{\infty}^{1}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \\ \vdots \\ \mu_{\infty}^{n_A}(\boldsymbol{\Lambda}_{\infty})\mathbf{I} \end{bmatrix} \tag{19}$$

and as assumption c) and Chapter 4.2 in [31] *yield* $\mathbf{A_{11}} \in \mathbf{S}_{++}$, *we have that the null space of* $\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_{\infty}(\boldsymbol{\Lambda}_{\infty})$ *reduces to the zero vector.*

(iii) Equivalence.

Now, from the stationary condition of the NMA, because L_p *is* $\mathcal{C}^1$ *with respect to* $(\mathbf{u},\boldsymbol{\mu})$ *and because* $\mathbf{u}(\cdot)$ *and* $\boldsymbol{\mu}(\cdot)$ *are differentiable, by applying the chain rule to the first-order necessary condition of problem (12), we obtain*

$$\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_{\infty}(\boldsymbol{\Lambda}_{\infty})\left(\nabla_{\mathbf{u}}\boldsymbol{\Phi}_{p}(\mathbf{u}_{\infty}) + \boldsymbol{\mu}_{\infty}^{T}\nabla_{\mathbf{u}}\mathbf{G}_{p}(\mathbf{u}_{\infty})\right) = -\nabla_{\boldsymbol{\Lambda}}\boldsymbol{\mu}_{\infty}(\boldsymbol{\Lambda}_{\infty})\mathbf{G}_{p}(\mathbf{u}_{\infty}) = -\nabla_{\boldsymbol{\Lambda}}\boldsymbol{\mu}_{\infty}(\boldsymbol{\Lambda}_{\infty})\mathbf{G}_{M\infty}(\mathbf{u}_{\infty},\boldsymbol{\theta},\boldsymbol{\Lambda}_{\infty}) \tag{20}$$

As condition b) implies that the right hand side of equation (20) is zero, as $\nabla_{\boldsymbol{\Lambda}}\boldsymbol{\mu}_{\infty}^{i}(\boldsymbol{\Lambda}) = 0$ *for any* $i > n_A$ *and* $G_{M\infty}^{i}(\mathbf{u}_{\infty},\boldsymbol{\mu}_{\infty},\boldsymbol{\Lambda}_{\infty}) = 0$ *for every* $1 \le i \le n_A$, *for any* $\boldsymbol{\Lambda}$ *close to* $\boldsymbol{\Lambda}_{\infty}$, *we deduce from the null space of* $\nabla_{\boldsymbol{\Lambda}}\mathbf{u}_{\infty}(\boldsymbol{\Lambda}_{\infty})$ *that*

$$\nabla_{\mathbf{u}}\mathbf{L}_{p}(\mathbf{u}_{\infty},\boldsymbol{\mu}_{\infty}) = \nabla_{\mathbf{u}}\boldsymbol{\Phi}_{p}(\mathbf{u}_{\infty}) + \boldsymbol{\mu}_{\infty}^{T}\nabla_{\mathbf{u}}\mathbf{G}_{p}(\mathbf{u}_{\infty}) = \mathbf{0}, \tag{21}$$

which is the stationary condition of the Lagrangian of the process. □

The result of Theorem 4.1 was already implemented in [32] for the unconstrained case in the Otto-Williams reactor, showing the equivalence of the optimal point found with the proposed methodology and with the original method. It was also applied following a hybrid implementation in [33] for a continuous fermentation example [34]. As expected, in both cases, the objective function of the upper optimization layer must be Φ_p.

4.1 Model Adequacy

From Theorem 4.1, the following conditions of model adequacy can be listed:

a) Regularity conditions on constraints must hold at $\mathbf{u}_\infty$.
b) Strict complementary slackness holds at $\mathbf{u}_\infty$.
c) The Hessian of the Lagrangian with respect to **u** for the model-based optimization must positive definite at $\mathbf{u}_\infty$.

Condition a) is a standard qualification condition in optimization problems, which, moreover, ensures the uniqueness of Lagrange multipliers. Condition b) has been assumed previously in the convergence analysis of the modifier-adaptation method [7]. Condition c) could be our strongest assumption, but it is satisfied by a large class of problems. In fact, Francois and Bonvin have shown in the same line that sufficient conditions of optimality can always be ensured provided that a convex approximation of the model is implemented [35]. Thus, if the model is strictly convex, then assumption c) is achieved. Therefore, further research can be focused on weakening assumption c), studying the effect of changing the model implemented in problem (10). Nevertheless, our main goal in this paper is to provide a new technique for solving process optimization problems, which, for simplicity of the exposition, has been developed in a simple setting.

4.2 Comments about the convergence

Regarding the convergence properties proposed by Faulwasser and Bonvin [26], it can be noted that as problems (10) and (3) are equivalent, the assumptions of global feasibility and uniqueness of the solution can be applied in the algorithm proposed. Furthermore, it was mentioned that an additional filtering step can be applied to the method (see equation (13)); thus, convergence depends on the value of matrix **K** and the algorithm implemented in the upper layer.

For a convex (or locally convex) problem, it is expected that a quasi-Newton method should behave well, provided a good numerical estimation of the gradient of L_p is available, which implies perturbing the process. However, the numeric estimation of the process gradient amplifies the measurement noise, and therefore convergence may not be achieved. However, if a gradient-free algorithm is selected, a better performance is expected under noisy environments [36] and when the objective function is "nonnumerical" (i.e., with an unknown structure according to [37]), such as an uncertain process. Conversely, there is a price that must be paid using these algorithms: the rate of convergence is asymptotically slower than the steepest descent method, and their performance worsens as the number of decision variables increases (this topic is commented in section 6).

Concerning the convergence rate of a direct search method, in the review of Kolda and co-workers, it is noted that the asymptotic behavior is important when the exact value of the optimum is required (see point 1.1.3 in [37]). However, in most real applications, this situation can be avoided, as the goal is "improvement" rather than full-blown optimality. This idea holds in most of the chemical engineering industrial applications, where the users are looking for "one or two correct digits", either because it is considered good for the operation or because of the inherent inaccuracy of the process measurements that makes seeking more accuracy pointless [37], e.g., the precision of the sensors, an incorrect sensor location or human error, among other factors.

In the particular case of the Nelder-Mead algorithm, there is evidence of stagnation at a non-stationary point. However, in most cases, this problem can be avoided by starting with good guesses of Λ, which can be obtained using numerical approximations of the process gradients in the first iteration. This policy has been applied in the following application examples.

5. Application Examples

To illustrate the method, the proposed nested methodology was applied in three examples, using the *fminsearch* and *fmincon* routines from the optimization toolbox of MATLAB [38] as optimization algorithms for the upper and the nested layer, respectively. The examples presented in this section were solved with and without noise in the process measurements. The implementation was compared with the dual modifier-adaptation methodology for different values of δ^L, with the aim of justifying the elimination of this tuning parameter due to its sensitivity.

5.1 Convex optimization example

This example is a simulated process optimization extracted from the work of Marchetti and co-workers [7], where the difference between the process and the model is the value of four uncertain parameters, which are not updated during the iterations of the algorithms.

Consider the convex optimization problem (22), where $\mathbf{u}$ are the decision variables, $\boldsymbol{\theta}$ represent the parameters and G is a single constraint:

$$\begin{aligned} &\min_{\mathbf{u}} \Phi(\mathbf{u},\boldsymbol{\theta}) := (u_1-\theta_1)^2 + 4(u_2-2.5)^2 \\ &s.t.: \\ &G(u,\theta) := (u_1-\theta_2)^2 - 4(u_2-\theta_4)\theta_3 - 2 \le 0 \\ &\{u_1,u_2\} \in [0,\infty) \end{aligned} \tag{22}$$

The target is to find the optimum of the process, represented as a simulated reality with a given set of correct parameters, using the modifier-adaptation methodology applied to a model with a set of erroneous parameters (corresponding to a simulated modeling mismatch). Table 1 summarizes the values of the parameters considered.

Table 1. Value of the parameters for the process and the model of example 1

	θ_1	θ_2	θ_3	θ_4
Simulated Process	3.5	2.5	-0.4	1.0
Model	2.0	1.5	-0.5	0.5

As the cost and the constraints are affected by the modeling mismatch, the three modifiers from equation (4) can be applied in this example, as shown by problem (23). The evolution of the system with the dual and the nested modifier-adaptation method will be compared for different implementation strategies for these modifiers.

$$\begin{aligned} &\min_{\mathbf{u}} \Phi(\mathbf{u},\boldsymbol{\theta}) := (u_1-\theta_1)^2 + 4(u_2-2.5)^2 + \lambda_{1k}u_1 + \lambda_{2k}u_2 \\ &s.t.: \\ &G(\mathbf{u},\boldsymbol{\theta}) := (u_1-\theta_2)^2 - 4(u_2-\theta_4)\theta_3 - 2 \le 0 + \gamma_{1k}(u_1-u_{1k}) + \gamma_{2k}(u_2-u_{2k}) + \varepsilon_k \\ &\{u_1,u_2\} \in [0,\infty) \end{aligned} \tag{23}$$

Regarding the implementation of both methods, the starting point was the optimum of the model. At this initial point, the first two iterations were used to estimate the gradients using the finite difference approach, as the dual methodology of modifier-adaptation requires at least two previous operation points to calculate the gradient, as shown by equation (6). In the case of the nested methodology, this gradient served as a good initial guess of the decision variables for the NM algorithm.

Figure 3 presents the evolution of the decision variables applied to the system with the dual modifier-adaptation method for two different values of parameter δ^L, the contours of the cost function and the constraints. In both plots, three different sets of values of $\mathbf{K}_\varepsilon$, $\mathbf{K}_\lambda$ and $\mathbf{K}_\gamma$ from equation (5) have been tested, describing different strategies of adaptation similarly to [7], while the points M and R represent the optimum of the model and the process, respectively.

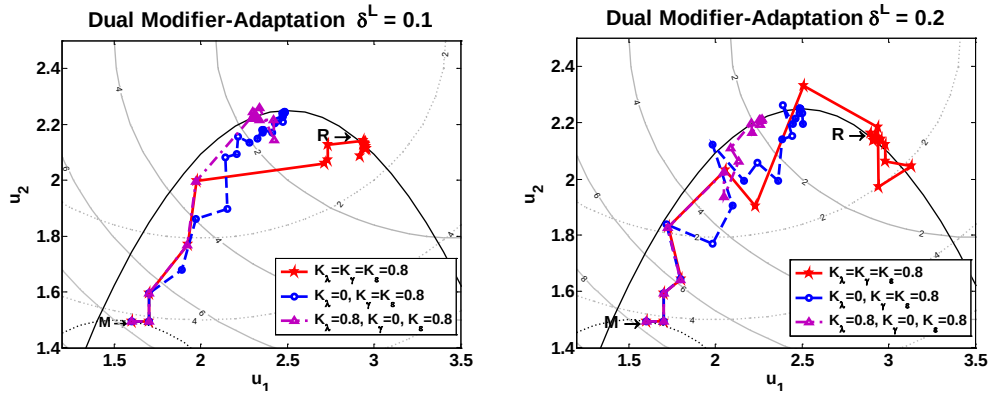

Figure 3. Evolution of dual modifier-adaptation. The thin lines are the contour of the model objective function (---), the contour of the process objective function (—), the model constraint (---), and the process constraint (—).

It can be seen from Figure 3 that the use of different adaptation strategies makes the iterative algorithm converge to different values, such that the full adaptation policy (the use of the three modifiers) is the only one that converges to the real optimum of the process for both values of δ^L tested. These results completely agree with the results previously reported in the literature.

The sensitivity of the dual methodology with respect to the value of δ^L can be observed by comparing the two graphs in Figure 3. It can be seen that the paths obtained with $\delta^L = 0.1$ converge in 6 iterations into a neighborhood of the real optimum, detecting the real active constraint, unlike

the evolution with $\delta^L = 0.2$, which requires 4 more iterations and violates the constraint *G*. This behavior indicates that the lower the value of the dual constraint, the more direct is the way to reach the optimum, but this approach implies a potential worse estimation of the process gradient. In fact, the choice of a larger value of this bound can generate infeasible points in the RTO path, as can be observed. The effect on the feasible region of the dual constraint has been studied in [14]. For the two-dimensional case, it consists of two discs with the same radius centered at the same distance from the centroid of the previous two decision variables, in an orthogonal direction to the line defined by these two last values (Figure 4). The size of the discs is inversely proportional to δ^L, which implies that the next operation point must be inside a smaller disc if the bound of the inverse of the condition number increases, which is translated into an evolution of the dual modifier adaptation that is continuously changing its direction, leading to a less direct route to the optimum.

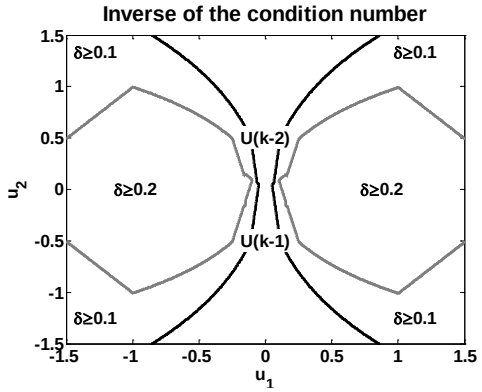

Figure 4. Feasible region of dual constraint for a two-dimensional problem; the contour lines represent δ^L.

Figure 5 shows the same experiment with the adaptation strategies in Figure 3, using the nested method proposed. Similarly to the original dual methodology, the only policy that reaches the real optimum of the process is the full modifier-adaptation, when all the modifiers are considered, while the other adaptation approaches become trapped at a suboptimal but feasible operation point. It can be noted that using this method, the evolution of the solutions does not present the zigzag behavior of Figure 3, and the algorithm converges in a direct path in to a neighborhood of the process optimum, using 5 iterations to detect the active constraint, whereas the remaining iterations attempt to refine the search of the real optimum. This behavior can be explained by the asymptotic rate of convergence of the NM algorithm, as noted in section 4.2. The evolution in the nested optimization problem is similar to the dual modifier-adaptation methodology with smaller values in δ^L, but the tuning of this parameter is avoided. In Appendix A, Theorem 4.1 is applied to this example using the convergence point achieved.

In the dual procedure, an important trade-off exists in the choice of δ^L. As previously described, a smaller value of this parameter implies a more direct evolution in the solutions. However, a lower parameter means that the gradient can be poorly estimated and that the noise in the measurements can be amplified. To test the sensitivity of δ^L with respect to noise, an additive error was simulated in the cost function and the constraint of the process. The stochastic simulated noise presents a uniform probability distribution centered at zero, with a range of ± 0.01.

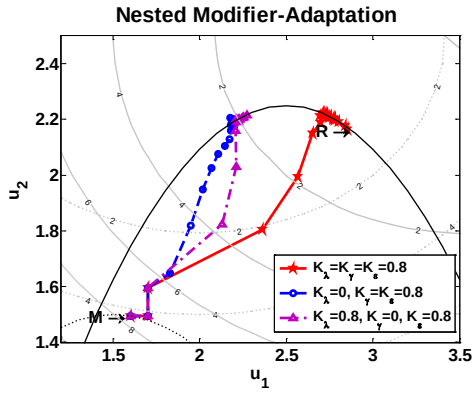

Figure 5. Evolution of the decision variables of the system with the nested modifier-adaptation policy. The thin lines are the contour of the model objective function (---), the contour of the process objective function (—), the model constraint (---), and the process constraint (—).

The same values of δ^L used in Figure 3 were employed to iterate with the dual modifier adaptation, while the terminal tolerance was increased by one order of magnitude with respect to the no-noise scenario, as recommended by Brdys and co-workers for the dual ISOPE methodology [14]. The results of the dual and nested methodologies, applying 30 trials for each method tested under stochastic noise conditions, are presented in Figure 6.

The evolution of the dual modifier-adaptation under noisy conditions shows how the wrong election of δ^L can make this method more sensitive to the noise stopping convergence because of a gradient estimation problem. The path shown for $\delta^L = 0.1$ presents similar performance to the no-noise conditions for the first 3 iterations of the dual modifier-adaptation. Afterward, when the solution approaches the optimum, the algorithm diverges. This behavior can be explained because starting from a value that is distant from the optimum of the process implies significant changes in the operation points, which can be translated as an inactive dual constraint, resulting in a well estimated gradient at the next iteration. As the system approaches the process optimum, the next operating point might be close enough to the previous values to activate the dual constraint, generating an *S* matrix more sensitive to noise. On the other hand, the path formed with $\delta^L = 0.2$ does not exhibit different behavior closer to the optimum, meaning that for a large enough value of the lower bound of the dual constraint, the algorithm converges to a neighborhood close to the real optimum. Nevertheless, the choice of a large value of δ^L is not free because it causes infeasibilities in the evolution of the algorithm in the same way as the noise-free scenario. These problems are avoided by the proposed methodology, as the δ^L parameter disappears and the feasible region remains unchanged with respect to the original modifier-adaptation method.

Upon comparing the evolution under noise conditions of the nested methodology and the dual methodology, it is clear that the convergence of the proposed methodology is less sensitive to the measurement error. This reduced sensitivity occurs because the way the algorithm reaches the real optimum is quite similar to the case when there is no noise, i.e., 5 iterations to detect the active constraint and the rest to locate the optimum of the process.

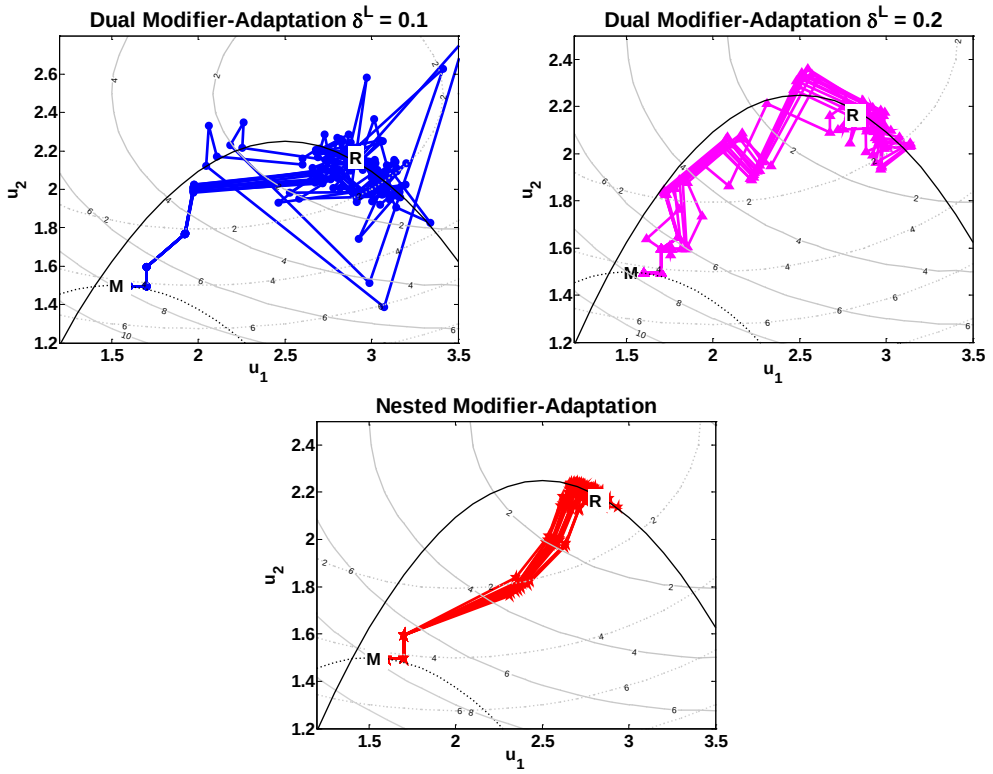

Figure 6. Comparison of the effect on the evolution of the decision variables for dual and nested modifier-adaptation methodologies under noisy conditions. The thin lines are the contour of the model objective function (—), the contour of the process objective function (—), the model constraint (---), and the process constraint (—). The thick lines are the evolutions of the modifier-adaptation algorithms.

5.2 Three interconnected tanks

The second example presented to test the proposed methodology is a system with three interconnected tanks (T_1, T_2 and T_3) that can store and exchange water with each other, as shown in Figure 7. The complete system receives flows of water q_{p1} and q_{p2} from two pumps (P_1 and P_2), while the outflows (q_1, q_2 and q_3) are the result of the fluid-dynamic potential between the surface and the bottom and the opening of the manual valves V_1, V_2 and V_3, respectively. The difference between the liquid heights (h_1, h_2 and h_3) will give the direction of the water that passes through the interconnections (q_{12} and q_{32}). These flows also depend on the head loss produced by the valves V_{12} and V_{32}.

This example was employed in the work of Marchetti and co-workers to test the modifier adaptation methodology in a real system [7]. Nevertheless, as the experimental setup is not available, the process and the model will be simulated in a similar way to example 1. A steady-state model of the system can be derived using a first principles approach, applying mass balances in each tank and Torricelli's rule to describe the flow that passes through the valves, as summarized by equation (24).

$$
\begin{aligned}
&q_{p1} - q_1 - q_{12} = 0 \\
&q_{12} + q_{32} - q_2 = 0 \\
&q_{p2} - q_3 - q_{32} = 0 \\
&q_i = Aa_i\sqrt{h_i}, i = 1,2,3 \\
&q_{j2} = sign\left(d_{j2}\right)Aa_{j2}\sqrt{\left|d_{j2}\right|}, j = 1,3 \\
&d_{j2} = h_j - h_2, j = 1,3
\end{aligned}
\tag{24}
$$

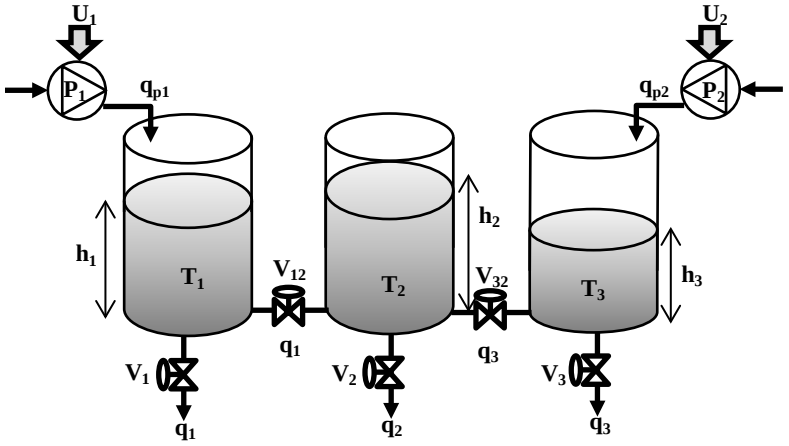

Figure 7. Diagram of the second example

Parameters a_i and a_{j2} represent the proportional constants of the valves V_i and V_{j2}, respectively, while $A = 154\ cm^2$ is the transversal area of each tank. The operational goal of the system is to determine the lowest energy to be applied to each pump (U_1 and U_2) such that the liquid inside the tanks remains within a certain range. If the water flow pumped is proportional to the energy consumed, the optimization problem is represented in problem (25):

$$
\begin{aligned}
&\min_{U_1,U_2} \Phi := U_1^{\,2} + U_2^{\,2} \\
&s.t.: \\
&\text{model from equation } 24 \\
&q_{pj} = w_j U_j \\
&U^L \le U_j \le U^U, j = 1,2 \\
&h^L \le h_i \le h^U, i = 1,2,3
\end{aligned}
\tag{25}
$$

where w_i is the proportional constant of the influent flow from the pumps. As previously described, to simulate the modeling mismatch, different values of the parameters for the model and the process for equation (24) must be used. The values that have been implemented in the simulated process are summarized in Table 2. They correspond to the original calibration coefficients obtained by Marchetti and co-workers after an identification stage with the original experimental setup [7]. On the other hand, the bounds of the constraints of problem (24) are h^L = 5 cm, h^U = 30 cm, U^L = 0 V and U^U = 8 V.

Table 2. Parameter values for the simulated process of example 2

Subscript (i or j)	a_i ($cm^{0.5}min^{-1}$)	a_{j2} ($cm^{0.5}$ min^{-1})	w_j (L V^{-1} min^{-1})
1	0.1203	0.0381	13.22
2	0.0613	–	14.96
3	0.1141	0.0285	–

The simulated modeling mismatch corresponds to a blockage in valves V_{12} and V_{32}, which can be translated into a reduction of the flow coefficient, as shown by equation (26).

$$a_{j2}^{M} = \frac{a_{j2}^{R}}{2}, j = 1,3 \tag{26}$$

Here, the superscripts M and R represent whether the parameter corresponds to the model or to the real process, respectively. The rest of the parameters of the model remain unchanged.

As in example 1, dual and nested modifier-adaptation methodologies were applied starting from the optimum of the model. In this example, only the modifiers of the constraints for h_i, i.e., γ_h and ε_h were updated because the constraints that restrict the value of U_i and the cost function depend only on the decision variables, and there is no uncertainty in them.

Figure 8 left shows the comparison between the evolutions of the decision variables with the dual and nested modifier-adaptation approaches in an ideal situation with no noise in the process. As before, in this figure, R and M represent the optimum of the real process and the model, respectively. Clearly, the mismatch changes the location of the optimum of the model; however, in both cases, the optimum occurs when the constraint $h_2 \geq 5$ is active.

It can be noted that the evolution of the decision variables of the dual methodology is affected by the value of the lower bound of the dual constraint, following different paths in the convergence to the real optimum, in a similar way to example 1: a lower value of this parameter implies a more rapid convergence to the real optimum because larger values of δ^L reduce the feasible region of the dual constraint (Figure 5). The performance of the dual methodology can be observed more clearly in the evolution of the objective function in Figure 8, on the right, where the tighter constraint requires only 5 iterations to find a region close to the optimum (coinciding with the detection of the active constraint), while the other two tested values of δ^L require 11 and 15 iterations to detect this region. On the other hand, the evolution of the nested modifier-adaptation methodology shows a convergence similar to the dual case with a lower value of δ^L, converging in 5 iterations to a region closer to the real optimum.

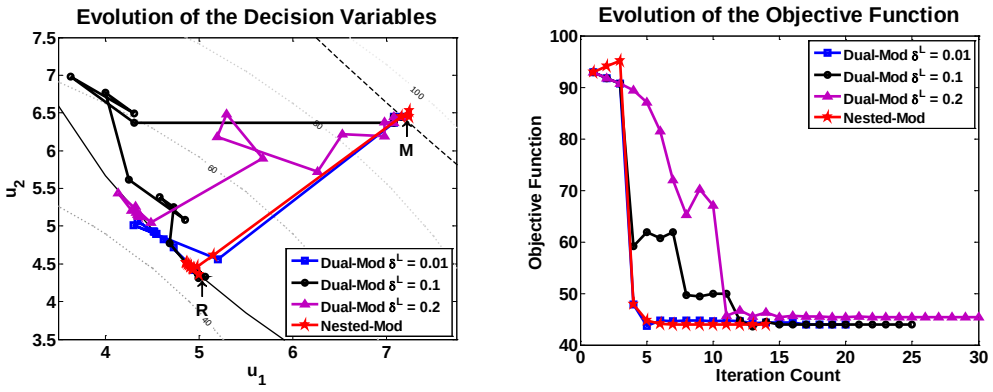

Figure 8. Evolution of the dual modifier-adaptation. The thin lines are the contour of the objective function (—), the constraint $h_2 \geq 5$ of the process (—) and the constraint $h_2 \geq 5$ of the model (---).

As observed, in the unlikely situation of a process without noise, the convergence of the nested methodology is comparable to the dual algorithm with lower values of δ^L in terms of optimality. Nevertheless, as subsequently shown, the trade-off between a larger feasible region and a matrix *S* with less energy to estimate the gradient of the process (see equation (6)) justifies the use of the proposed methodology for its robustness. To demonstrate this point, an additive noise was applied to the measurements of the liquid heights of the process, using a uniform probability distribution function centered at zero with a range of 0.1 cm. Figures 9 and 10 show the evolution of the dual and nested modifier-adaptation methodologies for 15 trials for each method under stochastic noise conditions.

As in example 1, the presence of noise affects the convergence to the real optimum of the dual modifier methodology, and the effect on measurement disturbances is more critical when a lower value of δ^L is used in the optimization. In fact, Figure 10 shows that the evolution of the cost function given by the dual methodology with $\delta^L = 0.01$ presents the largest dispersion and even the stagnation at different points for some trials tested. Increasing the value of this bound implies a better estimation of the process gradient, which can be observed in the fact that the other two dual methodologies converge to the real optimum ($\delta^L = 0.1$ and 0.2). However, the decrease in the size of the feasible region implies that if δ^L is overestimated, infeasible points can be produced, as shown by the evolution with $\delta^L = 0.2$. Unlike the dual modifier-adaptation, Figure 9 shows how the evolution of the nested methodology proposed in this work seems to be less sensitive to the measurement noise, converging similarly to the noise-free scenario. For all the cases tested, it can be seen in Figure 10 how the algorithm uses the same first 5 iterations to converge to a region closer to the optimum, corresponding to the detection of the active constraint in Figure 9, while the remaining iterations are used to refine the search of the real optimum, following the same pattern as the dual case with larger values of δ^L but with the advantage of removing a non-convex constraint (the dual one) while maintaining the original feasible region of the modified optimization problem.

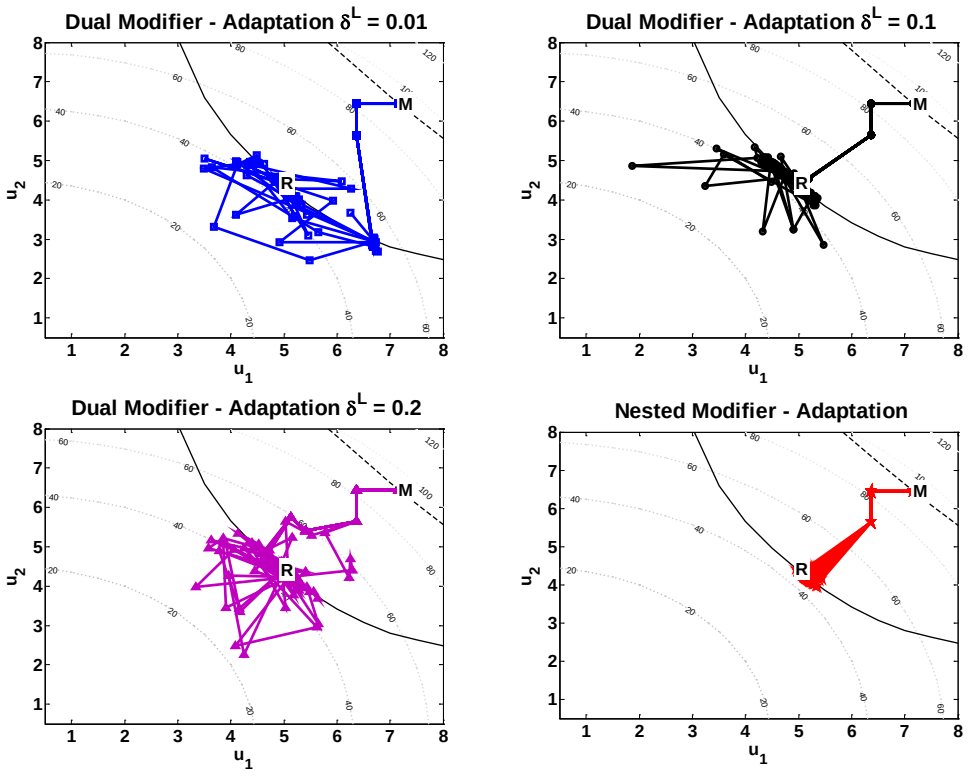

Figure 9. Evolution of the decision variables with the dual modifier-adaptation and the nested approach under noisy conditions. Thin lines are the contour of the objective function (---), the constraint $h_2 \geq 5$ of the process (—) and the constraint $h_2 \geq 5$ of the model (---). The thick lines are the evolutions of modifier-adaptation algorithms.

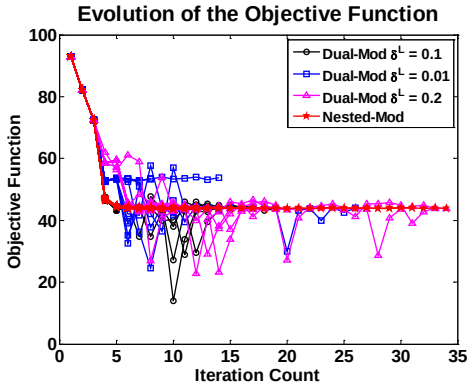

Figure 10. Evolution of the objective function under measurement noise conditions

5.3 Semi-Batch Reactor

The previous two examples have shown the expected evolution of the proposed nested approach when a parametric mismatch is observed. In the example presented in this section, the uncertainty in the model is given by a structural mismatch. In addition, it considers the process dynamics and the influence of process noise.

The system was presented as a test example in the work of Chachuat and co-workers in the context of a comparison among different ways of looking for the real optimum of a process with RTO [6]. It consists of a semi-batch reactor (Figure 11) used to produce 2-acetoacetyl pyrrole (C) from pyrrole (A) and diketene (B). Initially, the vessel contains a solution rich in A plus some amount of the other compounds and sub-products from parallel reactions that can occur: dehydroacetic acid (D), oligomers (E) and other undesired by-products (P). From the beginning to the end of the batch (t_f = 250 min), an inlet solution (F) with a constant concentration of diketene (C_B^{in}) can be fed into the reactor, and it is possible to regulate the flow rate throughout the period to change the concentrations of the products at the end of the batch.

The reactions that occur in the reactor can be summarized as follows:

$$\begin{aligned} &A + B \xrightarrow{k_1} C \\ &2B \xrightarrow{k_2} D \\ &B \xrightarrow{k_3} E \\ &C + B \xrightarrow{k_4} P \end{aligned} \tag{27}$$

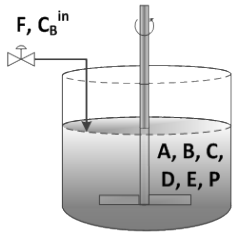

Figure 11. Diagram of the third example

A first-principles model can be used to describe the process based on mass balances:

$$\begin{aligned} \frac{dC_A}{dt} &= -r_1 - \frac{F}{V}C_A \\ \frac{dC_B}{dt} &= -r_1 - 2r_2 - r_3 - r_4 + \frac{F}{V}\left(C_B^{in} - C_B\right) \\ \frac{dC_C}{dt} &= -r_1 - r_4 - \frac{F}{V}C_C \\ \frac{dC_D}{dt} &= -r_2 - \frac{F}{V}C_D \\ \frac{dV}{dt} &= F \end{aligned} \tag{28}$$

where C_i is the molar concentration of the i compound, V is the reaction volume and r_j is the molar concentration rate from reaction j, which can be calculated as the product of the concentrations of the reactants:

$$
\begin{aligned}
r_1 &= k_1 C_A C_B \\
r_2 &= k_2 C_B^2 \\
r_3 &= k_3 C_B \\
r_4 &= k_4 C_B C_C
\end{aligned}
\tag{29}
$$

The operational goal of the system is to find the optimal profile of the flow rate of feed F such that the production of C at the final time is maximized, keeping C_B and C_D lower than specified upper bounds, solving the following optimization problem:

$$
\begin{aligned}
&\min_{F(t)} J_{Batch} := -C_C(t_f)V(t_f) \\
&s.t.: \\
&\text{equations 28 and 29} \\
&C_B(t_f) \le C_B^{up} \\
&C_D(t_f) \le C_D^{up} \\
&F^{lo} \le F(t) \le F^{up}, t \in [t_0, t_f]
\end{aligned}
\tag{30}
$$

Regarding the model of the process, as a source of mismatch, the existence of the last two reactions from equation (27) has been neglected, as they are parallel reactions that in general present conversion rates with orders of magnitude lower than the first two.

In the optimization problem (30), we are interested in obtaining a dynamic trajectory to be applied throughout the entire batch that minimizes the cost function and fulfills the inequality constraint at the end of the run. Even when this is not a steady-state problem, we can implement the modifier-adaptation methodology in a similar way to the steady problem, considering that the process measurements used to estimate the modifiers will be calculated after the batch is finished, and the decision variables calculated in the model-based optimization will be applied in the next batch.

The parameters used in the simulation of the process are summarized in Table 3, and the bounds of the optimization problem are F^L = 0 L min^{-1}, F^U = 2×10^{-3} L min^{-1}, $C_B{}^U$ = 0.025 mol L^{-1} and $C_D{}^U$ = 0.15 mol L^{-1}.

Regarding the model used in the optimization, the parameters k_3 and k_4 are set to zero, which is equivalent to neglecting the last two chemical reactions from equation (27). Far from being a parametric uncertainty, this omission reflects a structural mismatch, as physical dependences among the variables have been omitted because of partial knowledge of the real process.

Table 3. Parameter values and initial states for the simulated process of example 3

Parameters	Value	Initial States	Value
k_1	0.053 L mol^{-1} min^{-1}	$C_A(t_0)$	0.72 mol L^{-1}
k_2	0.128 L mol^{-1} min^{-1}	$C_B(t_0)$	0.05 mol L^{-1}
k_3	0.028 min^{-1}	$C_C(t_0)$	0.08 mol L^{-1}
k_4	0.001 L mol^{-1} min^{-1}	$C_D(t_0)$	0.01 mol L^{-1}
$C_B{}^{in}$	5 mol L^{-1}	$V(t_0)$	1 L
t_f	250 min	t_0	0 min

To solve the dynamic optimization, the single shooting approach was used to parameterize the optimal trajectory in three arcs, following the same procedure presented in the work of Chachuat

and co-workers [6]. In the first arc (from t_0 to t_s), the feed is at its upper bound; during the second arc (from t_s to t_m), it is assumed that the feed remains constant at an intermediate flow rate F_s; finally, during the third arc (from t_m to t_f), the flow rate of the feed is equal to zero. Therefore, the decision variables of the dynamic finite dimensional optimization problem are t_s, F_s and t_m.

Both the dual methodology and the proposed nested approach have been implemented in this example using the complete set of modifiers, as uncertainty is present in both the objective function and the inequality constraints of the state variables. Although in the original paper the authors only use the bias corrector of the inequality constraint ε, we also used λ and γ to compare the performance of the adaptation strategies, considering that if only ε is updated, then the dual and nested methodologies are the same.

Starting from the optimum of the model, the evolution of the decision variables and the objective function for the dual and nested methods are presented in Figure 12 considering no noise added. As in previous examples, we have tested different degrees of excitation of the system for the gradient-based approach, which is represented in the graphs for different values of δ^L. Regarding the nested implementation, in contrast to previous examples, in this case, the initial guesses of the modifiers for the upper layer were set equal to zero to let the NM algorithm search around the optimum of the model with the simplex in the first iteration.

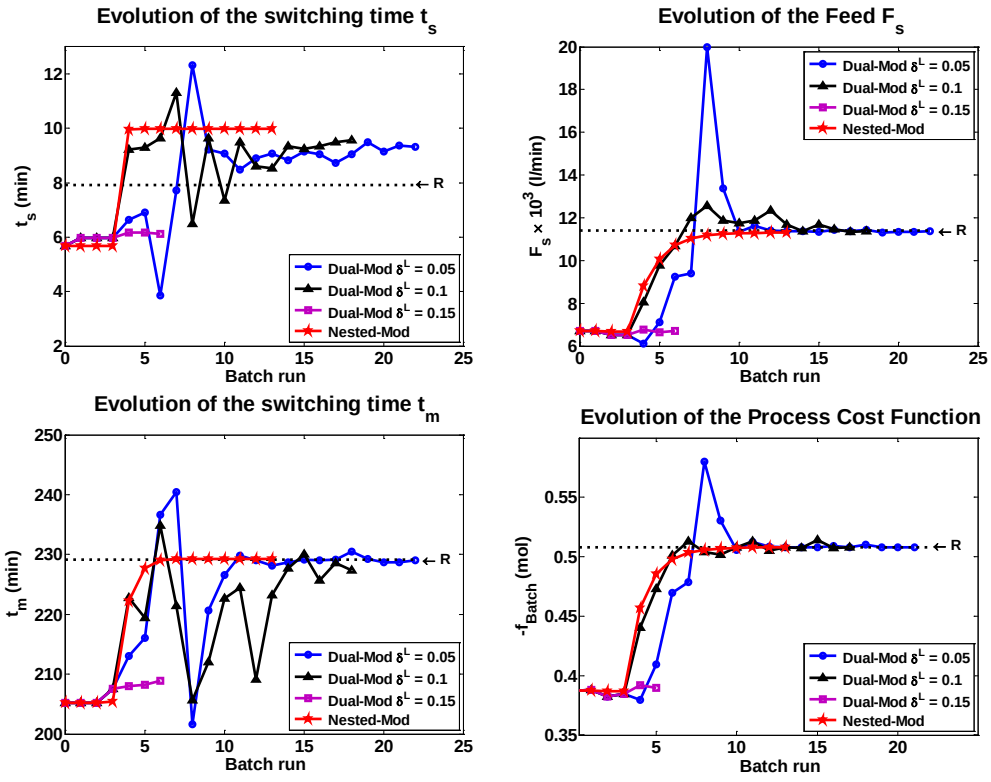

Figure 12. Evolution of the decision variables and the objective function for the dual and the nested modifier-adaptation methods in each batch. The dashed lines show the optimum of the process.

Notice that because the example is implemented in a run-to-run approach, each iteration of the RTO layer is a batch run. The comparison of the evolution of the cost function from Figure 12 shows that

for both methodologies, the RTO layer can find the optimum of the process. Regarding the evolution of the decision variables, the variables from the nested optimization show a smoother and faster evolution than those from the dual approach, where the speed of convergence depends on the value of δ^L. It can also be noted that no strategy was able to locate the real value of t_s, while F_s and t_m were successfully detected. The deterioration in the objective function as a consequence of the incorrect detection of t_s is 0.0056% in the nested approach, implying that the optimal value of the cost function from the process is barely sensitive to this decision variable.

The trajectories of the decision and constrained variables from the process obtained by applying the outcomes from the dual and the nested methods are compared with the expected optimal trajectories from the process in Figure 13.

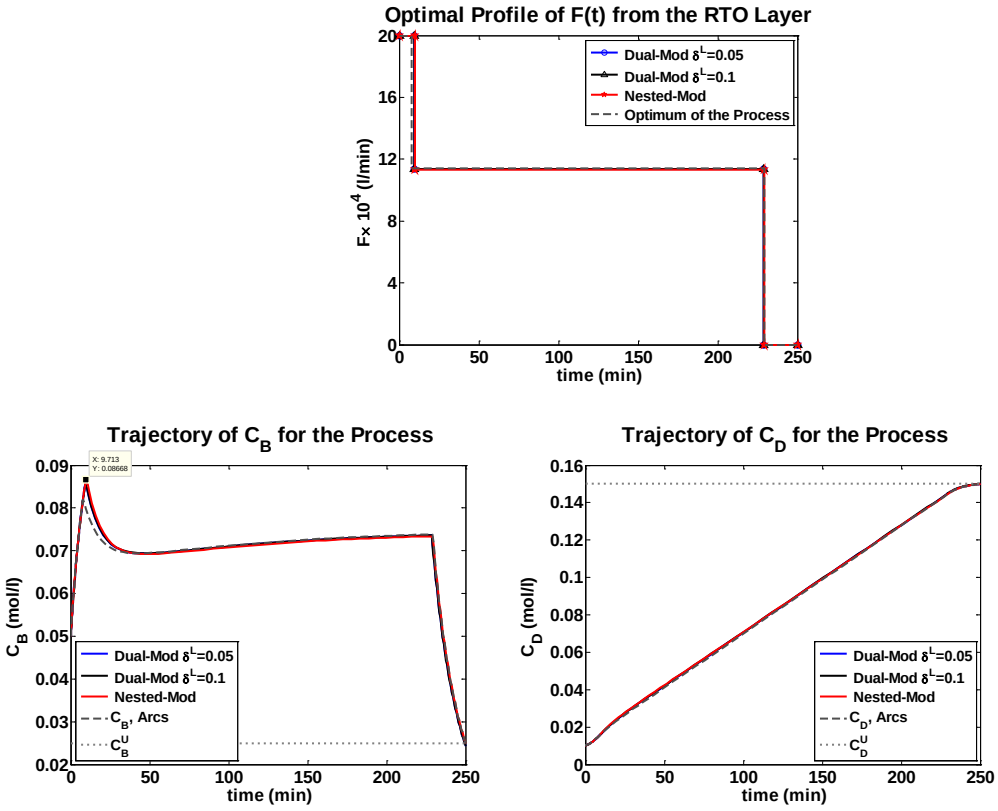

Figure 13. Optimal trajectory of the decision and constrained variables from the RTO layer

It can be seen from previous figures that the only difference in the trajectory of the constrained variables is a slight growth in the concentration peak of C_B (at t = 9.7 min); nevertheless, at t_f, C_B and C_D are at their upper bounds. Therefore, we can say that the RTO approaches were able to converge to the optimum of the process.

Next, an additive measurement noise was simulated in the state variables C_B, C_C and C_D, using a Gaussian zero mean distribution function with a 99.5% confidence interval equivalent to 10% of the expected range of the molar concentrations. The evolution of the objective function as well as the decision variables for the dual approach are presented in Figures 14 and 15 for different values of δ^L, while the nested algorithm is shown in Figure 16. In all figures, 10 evolutions of consecutive

batches starting from the optimum of the model and stopping when the RTO layer has converged to a stationary point are represented, whereas in each of the batches, the stochastic noise has been simulated.

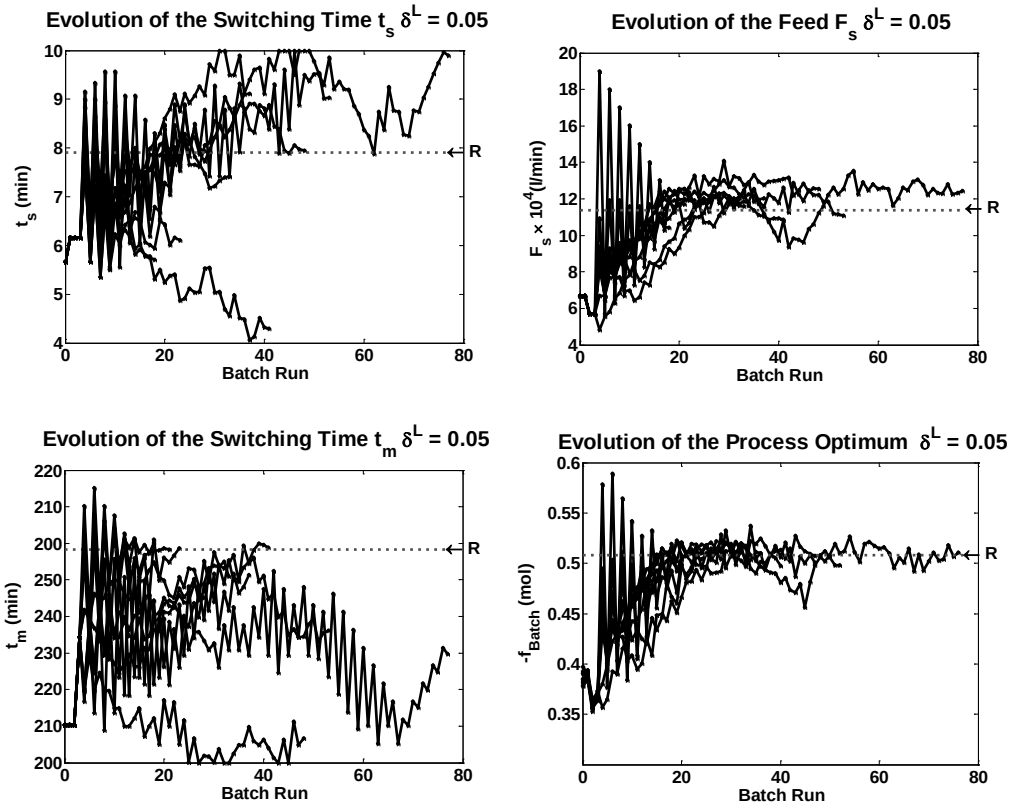

Figure 14. Evolution of the decision variables and the objective function for the dual modifier-adaptation with $\delta^L = 0.05$. The dashed lines show the process optimum.

As expected, in the gradient-based methodology, the evolution of a poorly excited system presents more sensitivity with respect to the measurement noise, showing differences in the capability to converge to the real process depending on the energy to estimate its gradients.

The evolution of the decision variables in the case of $\delta^L = 0.05$ shows how the random nature of the measurements is reflected in the path formed by the dual approach, producing a chaotic progression. Regarding the evolution of the objective function, it can be noted that in some cases, the system can converge to a region close to the optimum of the real system, but in a noisy way. The poor ability of the methodology to estimate the process optimum, as well as the dispersion in its evolution, is the consequence of a poor estimation of the experimental derivatives.

If the bound of the dual constraint is increased, then the process improves its capability to accurately estimate the real gradients, which implies a decrease in the dispersion of the path formed by the decision variables and the objective function with $\delta^L = 0.1$. In this case, the system can detect the process optimum using fewer iterations to reach a region close to the convergence point and following a more regular trajectory than a poorly excited system due to the improved estimation of the experimental derivatives.

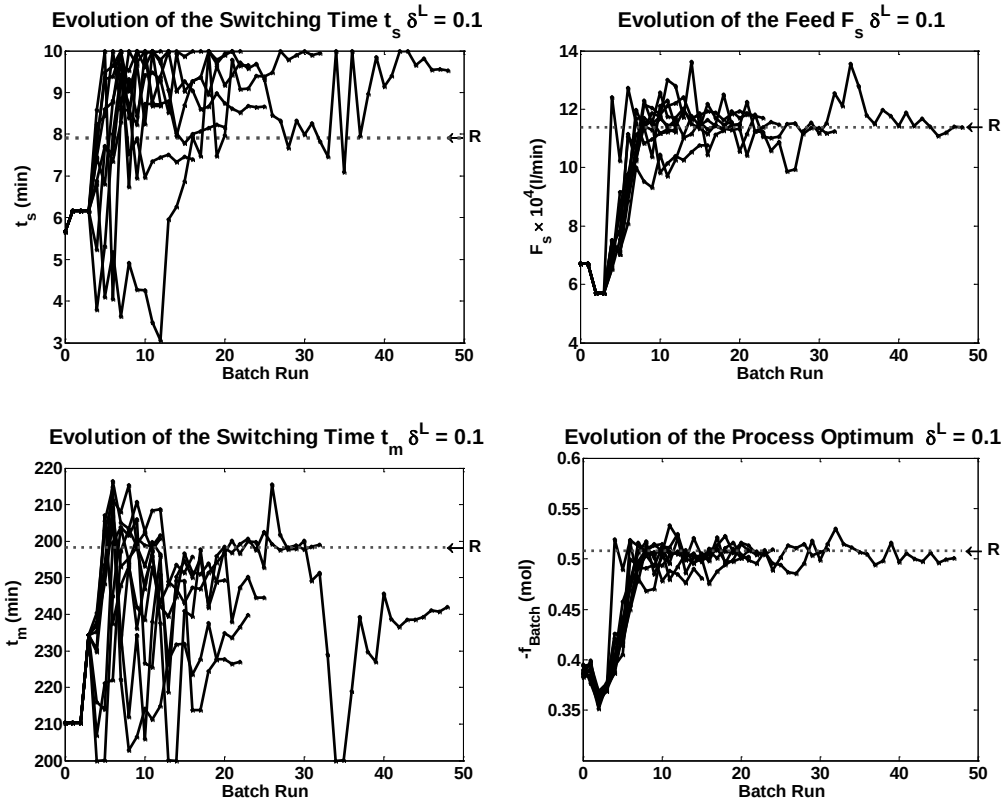

Figure 15. Evolution of the decision variables and the objective function for the dual modifier-adaptation with $\delta^L = 0.1$. Dashed lines show the optimum of the process.

Regarding the performance of the nested approach, as shown in Figure 16, the influence of the noise in the path formed by the decision variables and the objective function is minimal compared to the gradient-based methodology, and the shape of the evolution of the algorithm is quite similar to the noise-free scenario. The algorithm takes approximately 7 iterations to converge to a region close to the optimum of the process, and the remaining 23 iterations are used to improve the objective function. This result can be expected based on the convergence rate of the NM algorithm. It can also be noted that the volume of the simplex proposed by the NM algorithm decreases when the objective function is improved, which means that when the process is near its optimum, the size of the simplex is more affected by the noise of the objective function, worsening the convergence rate to the desired point. This behavior can also explain the small influence of the noise during the first iterations.

As in the noise-free scenario, both methodologies were unable to detect the real value of t_s. However, as previously noted, the limited effect of this decision variable on the value of the optimum makes this fact less important considering the improvements observed in the objective function applying the RTO layer.

Regarding the implementation of the nested methodology in the semi-batch reactor, we note that this method was able to find the real optimum of the process under structural modeling mismatch

and noise-free scenarios in a similar way to the gradient-based methodology but in fewer batch runs. If the measurements are contaminated with random noise, the nested approach is able to estimate a region close to the optimum of the process similarly to the noise-free scenario, which confirms the assumptions of the expected robustness.

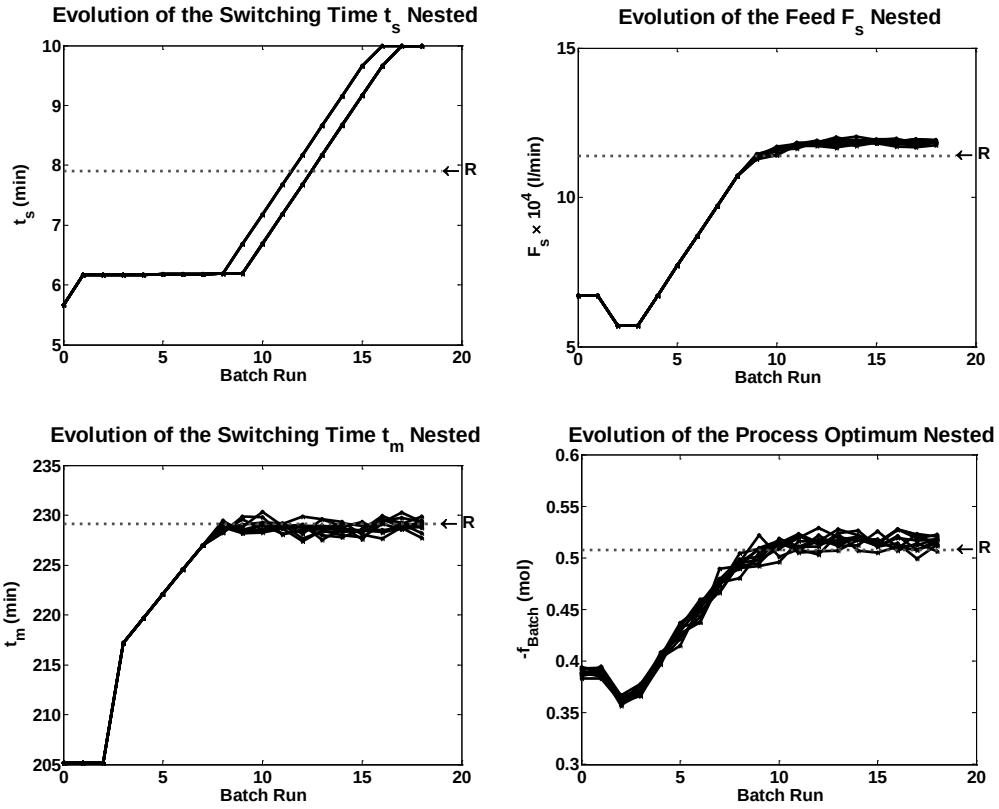

Figure 16. Evolution of the decision variables and the objective function for the nested modifier-adaptation. Dashed lines show the optimum of the process.

6. Conclusions

In this work, an alternative approach to the direct estimation of the process gradient in modifier-adaptation methodology has been proposed to increase its robustness and to simplify the implementation. The idea is to replace the errors in the gradients by a direct correction given by an upper gradient-free optimization layer that can operate without constraints.

Regarding the comparison of the proposed nested methodology with the dual modifier-adaptation methodology, it can be said that the convergence to a KKT point of the plant is quite similar in the cases when a relaxed dual constraint is employed in a noise-free scenario. However, when the process measurements are contaminated with noise, the proposed method seems to be more robust. It can also be said that the implementation of the nested methodology is easier than the implementation of the dual methodology, as one of the most sensitive parameters, the size of the feasible region of the dual constraint, is neglected, as well as the additional non-convexities that have no physical meaning in the real system.

The fact that the proposed methodology finds a KKT point of the real process in a robust way makes the nested modifier-adaptation method an attractive alternative for real implementation.

As a final conclusion of this work, it can be stated that the reformulation of the modifier-adaptation methodology as a nested optimization problem allows a better understanding of the concept of modifiers. This approach permits looking for additional alternatives to update the modifiers, with the idea of finding the best actualization algorithm that can be applied to the system, i.e., if the system has no important influences on the process noise and the process gradient is cheaper to obtain, then a gradient-based methodology can be applied. However, if there is noise in the system and/or the gradient of the process is not available or is expensive to obtain, the use of a direct search algorithm can be a better alternative, similarly to the field of model optimization, where the best optimization algorithm depends on the type of the problem to be solved.

As a final point, it is important to mention the problems that can appear when n_u increases, as the number of decision variables of the upper layer grows by n_u (n_g + 1). This growth can be a drawback for a direct search algorithm because of the well-known scalability problems of some simplicial algorithms [30], even if each new iteration only requires one or two new values of the process cost function. Further research may be needed to propose the implementation of the nested approach for large-scale industrial applications. However, the freedom of selecting different optimization algorithms in the upper layer provided by the reformulation gives additional alternatives to test other gradient-free methods with better performance in high-dimensional problems than the NM algorithm, maintaining the advantage of converging to the KKT point of a plant without any information about the process curvature. The same idea can be mentioned for the type of problems where the NM approach is not recommended, for example, when discontinuities can be produced in a region closer to the optimum of the process, where the application of other gradient-free algorithms can be proposed to address these challenges.

Acknowledgements

The research leading to these results received funding from Project OPEROPT, DPI2012-37859 of the Spanish MINECO and the EU project MORE under GA 604068. The financial support of UTFSM internal project N° 271455 is also greatly appreciated.

BIBLIOGRAPHY

1. Marlin, T. E.; Hrymak, A. N., Real-time Operations Optimization of Continuous Processes. *AICHE SYMPOSIUM SERIES* **1997**, (316), 156-164.
2. Bamberger, W.; Isermann, R., Adaptive on-line steady-state optimization of slow dynamic processes. *Automatica* **1978,** 14, (3), 223-230.
3. Roberts, P. D., AN ALGORITHM FOR STEADY-STATE SYSTEM OPTIMIZATION AND PARAMETER-ESTIMATION. *Int. J. Syst. Sci.* **1979,** 10, (7), 719-734.
4. Tatjewski, P., Iterative Optimizing Set-Point Control - The Basic Principle Redesigned. In *15th Triennial World Congress, Barcelona, Spain*, IFAC: 2002.
5. Gao, W.; Engell, S., Iterative set-point optimization of batch chromatography. *Computers & Chemical Engineering* **2005,** 29, (6), 1401-1409.
6. Chachuat, B.; Srinivasan, B.; Bonvin, D., Adaptation strategies for real-time optimization. *Computers & Chemical Engineering* **2009,** 33, (10), 1557-1567.
7. Marchetti, A.; Chachuat, B.; Bonvin, D., Modifier-Adaptation Methodology for Real-Time Optimization. *Industrial & Engineering Chemistry Research* **2009,** 48, (13), 6022-6033.

8. Marchetti, A.; Chachuat, B.; Bonvin, D., A dual modifier-adaptation approach for real-time optimization. *Journal of Process Control* **2010,** 20, (9), 1027-1037.
9. Marchetti, A. G., A new dual modifier-adaptation approach for iterative process optimization with inaccurate models. *Computers & Chemical Engineering* **2013,** 59, (0), 89-100.
10. Biegler, L. T.; Grossmann, I. E.; Westerberg, A. W., A note on approximation techniques used for process optimization. *Computers & Chemical Engineering* **1985,** 9, (2), 201-206.
11. Forbes, J. F.; Marlin, T. E., Model Accuracy for Economic Optimizing Controllers: The Bias Update Case. *Industrial & Engineering Chemistry Research* **1994,** 33, (8), 1919-1929.
12. Forbes, J. F.; Marlin, T. E.; MacGregor, J. F., Model adequacy requirements for optimizing plant operations. *Computers & Chemical Engineering* **1994,** 18, (6), 497-510.
13. Bunin, G. A.; François, G.; Srinivasan, B.; Bonvin, D. In *Input Filter Design for Feasibility in Constraint-Adaptation Schemes*, 18th IFAC World Congress, 2011; 2011; pp 5585-5590.
14. Brdys, M. A.; Tatjewski, P., *Iterative algorithms for multilayer optimizing control*. Imperial College Press: London, 2005; p xvi, 370 p.
15. Mansour, M.; Ellis, J. E., Comparison of methods for estimating real process derivatives in on-line optimization. *Applied Mathematical Modelling* **2003,** 27, (4), 275-291.
16. Brdys, M.; Tatjewski, P. In *An algorithm for steady-state optimizing dual control of uncertain plants*, 1st IFAC Workshop on New Trends in Design of Control Systems, Smolenice, Slovakia, 1994; IFAC, Ed. IFAC: Smolenice, Slovakia, 1994.
17. Rodger, E.; Chachuat, B., Design Methodology of Modifier Adaptation for On-Line Optimization of Uncertain Processes. In *18th IFAC World Congress*, Bittanti, S.; Cenedese, A.; Zampieri, S., Eds. IFAC: Università Cattolica del Sacro Cuore, Milano, Italy, 2011; Vol. 18, pp 4113-4118.
18. François, G.; Bonvin, D., Use of Transient Measurements for the Optimization of Steady-State Performance via Modifier Adaptation. *Industrial & Engineering Chemistry Research* **2014,** 53, (13), 5148-5159.
19. Bunin, G.; François, G.; Bonvin, D., Exploiting Local Quasiconvexity for Gradient Estimation in Modifier-Adaptation Schemes. In *The 2012 American Control Conference*, Montréal, Canada, 2012; pp 2806-2811.
20. Bunin, G. A.; François, G.; Bonvin, D., From Discrete Measurements to Bounded Gradient Estimates: A Look at Some Regularizing Structures. *Industrial & Engineering Chemistry Research* **2013,** 52, (35), 12500-12513.
21. Gao, W.; Wenzel, S.; Engell, S., Comparison of Modifier Adaptation Schemes in Real-Time Optimization. In *ADCHEM 2015*, IFAC: Whistler, Canada., 2015; Vol. 48, pp 182-187.
22. Gao, W.; Wenzel, S.; Engell, S. In *A reliable modifier adaptation strategy for real time optimization*, PSE 2015/ESCAPE 25, Copenhagen, Denmark, 31 May-4 June 2015, 2015; Copenhagen, Denmark, 2015.
23. Krstic, M.; Wang, H.-H., Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica* **2000,** 36, (4), 595-601.
24. Guay, M.; Zhang, T., Adaptive extremum seeking control of nonlinear dynamic systems with parametric uncertainties. *Automatica* **2003,** 39, (7), 1283-1293.
25. François, G.; Srinivasan, B.; Bonvin, D., Comparison of Six Implicit Real-Time Optimization Schemes. *Journal Européen des Systèmes Automatisés* **2012,** 46, (2-3), 291-305.
26. Faulwasser, T.; Bonvin, D., On the Use of Second-Order Modifiers for Real-Time Optimization. In *19th IFAC World Congress*, Boje, E.; Xia, X., Eds. IFAC: Cape Town, South Africa, 2014; Vol. 19, pp 7622-7628.
27. Bunin, G. A., On the equivalence between the modifier-adaptation and trust-region frameworks. *Computers & Chemical Engineering* **2014,** 71, 154-157.
28. Lagarias, J. C.; Reeds, J. A.; Wright, M. H.; Wright, P. E., Convergence Properties of the Nelder--Mead Simplex Method in Low Dimensions. *SIAM J. on Optimization* **1998,** 9, (1), 112-147.
29. Walters, F. H.; Parker, L. R.; Morgan, S. L.; Stanley, D. N., *Sequential simplex optimization: a technique for improving quality and productivity in research, development, and manufacturing*. CRC Press: Boca Raton, Fla., 1991; p xix, 325 p.
30. Anderson, E.; Ferris, M., A Direct Search Algorithm for Optimization with Noisy Function Evaluations. *SIAM J. on Optimization* **2000,** 11, (3), 837-857.
31. Fiacco, A. V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*. Elsevier Science: 1983.

32. Navia, D.; Gutierrez, G.; de Prada Moraga, C., Nested Modifier-Adaptation Methodology for RTO in the Otto Williams Reactor. In *10th International Symposium on Dynamics and Control Process Systems (DYCOPS 2013)*, IFAC: Mumbai, India, 2013.
33. Navia, D.; Gutierrez, G.; De Prada Moraga, C., Mixed Modifier-Adaptation for RTO in a Continuous Bioreactor. In *19th IFAC World Congress*, Boje, E.; Xia, X., Eds. IFAC: Cape Town, South Africa, 2014; Vol. 19, pp 7635-7640.
34. Golden, M. P.; Ydstie, B. E., Adaptive extremum control using approximate process models. *AIChE Journal* **1989,** 35, (7), 1157-1169.
35. François, G.; Bonvin, D., Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation. *Industrial & Engineering Chemistry Research* **2013,** 52, (33), 11614-11625.
36. Rios, L.; Sahinidis, N., Derivative-free optimization: a review of algorithms and comparison of software implementations. *J Glob Optim* **2013,** 56, (3), 1247-1293.
37. Kolda, T.; Lewis, R.; Torczon, V., Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods. *SIAM Review* **2003,** 45, (3), 385-482.
38. Mathworks, I., *MATLAB Optimization Toolbox 6 user's guide*. The MathWorks: Natick, Mass., 2007.

Appendix A

To evaluate the conditions presented in Theorem 4.1, note that the Hessian of the modified problem (23) is

$$\nabla^2_{\mathbf{uu}}\mathbf{L} = \begin{bmatrix} 2+\mu & 0 \\ 0 & 8 \end{bmatrix} \in S_+ \tag{A.1}$$

Therefore, it is possible to estimate the matrix $\nabla_{\Lambda}\mathbf{u}_{\infty}$ (see Chapter 4.2 in [31] and equation (19)) as

$$\begin{aligned}
&\nabla_{\Lambda}\mathbf{u}_{\infty} = \begin{bmatrix}\nabla_{\gamma}\mathbf{u}_{\infty} & \nabla_{\lambda}\mathbf{u}_{\infty}\end{bmatrix} = \begin{bmatrix}\mu\mathbf{T} & \mathbf{T}\end{bmatrix} \\
&\mathbf{T} := \begin{bmatrix} A & B \\ B & C \end{bmatrix} \\
&A = \frac{1}{2(\mu_{\infty}+1)}\left[\frac{(\gamma_{1\infty}-2\theta_2+2u_{1\infty})^2}{2(\mu+1)\Delta}-1\right] \\
&B = \frac{\mu(\gamma_{2\infty}-4\theta_3)(\gamma_{1\infty}-2\theta_2+2u_{1\infty})}{16(\mu_{\infty}+1)\Delta} \\
&C = \frac{1}{8}\frac{(\gamma_{2\infty}-4\theta_3)^2}{8\Delta}-\frac{1}{8} \\
&\Delta = 2(\mu_{\infty}+1)(\gamma_{1\infty}-2\theta_2+2u_{1\infty})^2+8(\gamma_{2\infty}-4\theta_3)^2
\end{aligned} \tag{A.2}$$

Note that in equation (A.2), the parameters θ are the ones for the model. Replacing the value of the stationary point obtained using the nested procedure, $\mathbf{u}_{\infty}$=[2.976;2.108], λ_{∞}=[-2.607;-0.322], γ_{∞}=[-2.632;-0.306], μ_{∞}=2.04, and matrix **T** gives

$$\mathbf{T}_{\infty} = \begin{bmatrix} -0.164 & 5E-4 \\ 5E-4 & -0.123 \end{bmatrix} \tag{A.3}$$

Matrix $\mathbf{T}_{\infty}$ has an empty null space. Therefore, the stationary condition of the upper layer implies that $\nabla_{\Lambda}\mathbf{L}_{p\infty}$ must be zero. This implication can be easily confirmed by replacing the stationary values in the gradient of the Lagrangian function of problem (22) but using the parameters θ of the

process. The same procedure can be applied for the inequality constraint, giving G = 0. As $\mu_{\infty}>0$, the remaining the KKT conditions of the process are satisfied. Considering that equation (A.1) is also valid for the process, it can be concluded that the nested methodology found a KKT point of the real system.