I&EC research
Industrial & Engineering Chemistry Research

Article

pubs.acs.org/IECR

Including Disjunctions in Real-Time Optimization

Fernán J. Serralunga, Pio A. Aguirre, and Miguel C. Mussati*

INGAR Instituto de Desarrollo y Diseño (CONICET - UTN), Avellaneda 3657 (S3002GJC) Santa Fe, Argentina

Supporting Information

ABSTRACT: Real-time optimization (RTO) is widely used in industry to improve the steady-state performance of a process using the available measurements, reacting to changing prices and demands scenarios and respecting operating, contractual, and environmental constraints. Traditionally, RTO has used nonlinear continuous formulations to model the process. Mixed-integer formulations have not been used in RTO, because of the need of a fast solution (on the order of seconds or a few minutes), and because many discrete decisions, such as startups or shutdowns, are taken with less frequency in a scheduling layer. This work proposes the use of disjunctions in RTO models, listing a series of examples of discrete decisions (different to startups or shutdowns) that can be addressed by RTO. Two model adaptation approaches (the two-step approach and the modifier adaptation strategy) are revised and modified to make them suitable for RTO with discrete decisions. Some common techniques used in RTO (such as filtering the optimal inputs) are also analyzed and adapted for a formulation with disjunctions. The performance of RTO with disjunctions is shown by a case study in which a generic process is optimized. The results show that the performance of a process can be improved by RTO with discrete decisions. The system converges to the vicinity of the real plant optimum when constraints gradients are corrected, even under structural and parametric mismatch.

1. INTRODUCTION

The performance of industrial processes can be continuously improved with the use of computer-aided tools. Among these tools, real-time optimization (RTO) makes use of the online available measurements to maximize a process performance index, while accounting for operative, environmental and contractual constraints under changing scenarios. The results from RTO can be sent directly to the control system or as targets to a model-based predictive control system.

In most cases, RTO makes use of a model to find the optimal process operating point. According to two recent reviews,[1,2] the steps followed in an RTO cycle are steady-state detection,[3,4] data validation (which may include data reconciliation[5,6]), model adaptation, and performance optimization.

A key step in RTO is the model *adaptation*[7] with the available measurements, as a way to reduce the structural and parametric mismatch between the real plant and the model. The way the model is adapted is an important feature of an RTO approach, as it impacts on the feasibility and optimality of the results. Because of this, several adaptation strategies have been developed.

The *two-step* approach[8] is the most common adaptation strategy in industrial applications. It solves two optimization problems: the first one is an error minimization to update model parameters, and the second one is the maximization of the performance index using those updated parameters. This strategy requires a large number of parameters and a special model structure to converge to the real plant optimum.[9] Nevertheless, these requirements are rarely met in practice.

Modifier adaptation[10] eliminates the error minimization step by resorting to the addition of correction terms (*modifiers*) in constraints and objective function. The correction terms can be updated using an exponential filter.[11] When constraints are updated only with a bias,[12] the strategy is called *constraint adaptation*.[13,14] If gradient modifiers are also added, the strategy has the property to match a real plant Karush−Kuhn−Tucker (KKT) point, upon convergence.[9] A recent work shows that, for modifier adaptation schemes, if the plant model is built using convex approximations the model *adequacy* is enforced (i.e., upon convergence, the system will match a real plant local minimum).[15] The use of gradient correction terms was first proposed with the ISOPE technique.[16,17]

Direct input strategies[18,19] emulate the structure of a control system with a set-point that implies that the plant operates at its optimum point. For example, the KKT conditions of the problem can be defined as the target or setpoint to follow.

Modifier adaptation, ISOPE and direct input strategies require the estimation of gradients, which is not a trivial task; several techniques have been developed.[20,21] Recent publications[22−24] have proposed the use of weighted regressions of past data to estimate gradients. Dynamic data can also be used to estimate steady-state gradients.[25]

The results from performance optimization can be validated before being applied to the real plant,[26] or filtered with a so-called *input filter*.[27] The input filter can be designed to guarantee feasibility upon certain conditions.[28]

All the aforementioned references use nonlinear continuous models for RTO. Mixed-integer nonlinear programming (MINLP) or generalized disjunctive programming (GDP) formulations generally have not been used in RTO, for two main reasons. The first one is that RTO requires a fast solution (on the order of seconds or a few minutes) and problems involving discrete variables are more complex to solve. The second is that many discrete decisions, such as start-up or

Special Issue: Jaime Cerdá Festschrift

Received: February 3, 2014
Revised: May 10, 2014
Accepted: May 22, 2014
Published: May 22, 2014

   dx.doi.org/10.1021/ie5004619 | *Ind. Eng. Chem. Res.* 2014, 53, 17200−17213

shutdown of pieces of equipment, cannot be properly addressed with a single-period optimization formulation, and therefore they are reserved to the scheduling optimization layer, which is performed with a lower frequency, in comparison to RTO.

However, the evolution of processing capacity, as well as the development of more efficient MINLP and logic-based algorithms, can make problems of a typical size used in RTO (hundreds of variables and constraints, and a few discrete decisions) solvable in the time required by this application. Recent works show industrial applications of MINLP optimization updating the model and prices with real-time data,[29−32] although they do not include any study of the plant−model mismatch and the strategies to adapt the model.

This work addresses the inclusion of certain types of disjunctions in RTO problems. In addition, it will show how discrete decisions that do not involve startups or shutdowns can appear in an industrial plant and may be addressed by RTO. For example, complex economic contracts, disjoint operative regions, piecewise functions or dual control[11] constraints may require a formulation with disjunctions or discrete variables. The authors have introduced this problem in a recent work, which presents RTO with disjunctions for a heat and power system.[33]

The two-step and modifier adaptation strategies, suitable for nonlinear continuous problems, are extended to make them capable of dealing with discrete decisions. This includes deciding which parameters or modifiers will be updated on each RTO cycle, depending on the values of discrete variables in the plant at that moment; this also includes which past data will be used for gradient estimation, which is dependent on the values of discrete variables in the past. The use of input filter and other common practices in RTO (such as fixing a maximum change in some variables, or adding a *convexifying term*[27] in the objective function) can lead to suboptimal solutions or infeasibilities if disjunctions are present. Because of this, new bounds for the input filter are defined, and a reformulation of the strategies for maximum changes and convexifying terms is also presented.

This paper is organized as follows: Section 2 briefly introduces Generalized Disjunctive Programming (GDP). Section 3 presents five types of disjunctions that can appear in an RTO system. Section 4 presents the RTO problem to solve and shows how different adaptation strategies can be used with GDP/MINLP problems. Section 5 shows the convergence and feasibility problems with some strategies used traditionally in RTO (limiting the maximum change, filtering the optimal inputs), and proposes a modification of these strategies to make them suitable for problems with discrete variables. Section 6 presents and discusses a case study: it consists of the RTO of a generic system formed by three processes with structural plant/model mismatch. Finally, the Conclusions section (section 7) summarizes the main results of this work.

2. GENERALIZED DISJUNCTIVE PROGRAMMING

When discrete and continuous variables are present, an optimization problem can often be formulated using Generalized Disjunctive Programming (GDP).[34] This formulation consists of modeling using algebraic constraints, logic disjunctions, and logic propositions. Its general structure can be stated as

$$\min f(\mathbf{x}) + \sum_d c_d \tag{1}$$

subject to

$$\mathbf{g}_G(\mathbf{x}) \le 0$$

$$\bigvee_{i \in N_d} \begin{bmatrix} z_{id} \\ \mathbf{g}_{id}(\mathbf{x}) \le \mathbf{0} \\ c_d = \gamma_{id} \end{bmatrix}, \; d \in D$$

$$\mathbf{\Omega}(\mathbf{z}) = \text{true}$$

$$\mathbf{x}^{\mathrm{L}} \le \mathbf{x} \le \mathbf{x}^{\mathrm{U}}$$

$$\mathbf{x} \in R^n, \; c_d \in R^1, \; z_{id} \in \{\text{true, false}\}$$

where f is a continuous function of continuous variables $\mathbf{x}$; *global* inequality constraints $\mathbf{g}_G(\mathbf{x}) \le 0$ must always be satisfied. Each *disjunction* $d \in D$ is formed by N_d terms, related by the logical operator OR. Each term $i \in N_d$ is characterized by the Boolean variable z_{id}; if z_{id} = true, the constraints $\mathbf{g}_{id}(\mathbf{x}) \le \mathbf{0}$ must be satisfied, and the fixed cost c_d is equal to γ_{id}. Logical propositions $\mathbf{\Omega}(\mathbf{z})$ = true are constraints that are function of the Boolean variables $\mathbf{z}$ (with $\mathbf{z}$ being the vector of all variables z_{id}).[35]

Once the model is formulated as a GDP problem, it can be transformed to a MINLP problem, using big-M or convex hull reformulations.[36] This way, the problem can be solved with any of the MINLP solvers currently available. Logic-based algorithms have also been developed, which exploit the particular structure of GDP problems.[37,38]

GDP was applied with success in the areas of process synthesis and planning/scheduling optimization. In the following sections of this work, the application of GDP for real-time optimization is detailed formally and illustrated with examples.

3. TYPICAL DISJUNCTIONS IN RTO

In process operation optimization, disjunctions are usually related to startups or shutdowns of process units or pieces of equipment. As mentioned in the Introduction section, real-time optimization may not include startups or shutdowns; these decisions can be addressed more properly in a multiperiod strategy, which can account for transition constraints (warm-up periods, minimum times between startups and shutdowns, penalization of the number of changes).[39] However, there are several aspects of the process—the control strategy, the model, and the RTO implementation—that can require discrete or binary decisions, and therefore they can be modeled using GDP.

The following subsections describe some disjunctions that can be part of a RTO optimization problem.

3.1. Piecewise Functions. Piecewise functions can be included in the optimization problem in order to model the behavior of the process (or the control strategy) with different functions in different operative regions:

$$\begin{bmatrix} z_{d,1} \\ g = g_1(u) \le 0 \\ u_1^{\mathrm{L}} \le u \le u_2^{\mathrm{L}} \end{bmatrix} \vee \cdots \vee \begin{bmatrix} z_{d,p} \\ g = g_p(u) \le 0 \\ u_p^{\mathrm{L}} \le u \le u_{p+1}^{\mathrm{L}} \end{bmatrix} \vee \cdots \vee \begin{bmatrix} z_{d,np} \\ g = g_{np}(u) \le 0 \\ u_{np}^{\mathrm{L}} \le u \le u_{np+1}^{\mathrm{L}} \end{bmatrix} \tag{2}$$

where p indicates the active piece (1 to np), g_p is the function active in piece p, and u_p^{L} is the lower bound for each piece p.

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

3.2. Economic Contracts—Penalties in the Objective Function. If the objective function is the cost (or the profit), it may include fixed costs or penalties that apply under certain conditions. A fixed penalty can be included, for example, in a power purchase contract if power exceeds a maximum value; emissions can be penalized with fixed costs if they override the legal limit. These fixed costs need to be modeled with discrete variables or disjunctions.

Complex contracts included in the objective function can also be modeled by piecewise functions of similar nature to those referred to in section 3.1.

3.3. Disjoint Operating Regions. Depending on the structure of the control system, there may be disjoint feasible regions in the system to optimize. For example, a valve that can either be opened from 5% to 100% or fully closed can be modeled with a disjunction. Operative decisions can also include rules indicating not to operate within a certain region for safety or reliability reasons.

3.4. Dual RTO. Modifier adaptation RTO (described later in section 4.2.2) requires the estimation of the real plant cost and constraints gradients. When this estimation is based on data from past RTO cycles, additional constraints are often added to the cost optimization, so that the new optimal inputs are useful for the experimental gradient calculation. These constraints bound the error generated by noise and by first-order truncation,[10] and they allow evaluating the gradient in all directions.[40]

Real-time optimization that includes these extra constraints is called *dual RTO* or *dual control*,[11] because it follows two objectives: cost minimization and generation of useful points for the gradient estimation problem. The additional constraints generate two disjoint feasible regions at both sides of the hyperplane defined by the last *nu* points in the space of the inputs (where *nu* is the number of inputs of the system). A new disjunction can be included in the RTO problem at each RTO cycle *k*:

$$\begin{bmatrix} z_{\text{dual}} \\ \mathbf{e}^{+}(\mathbf{u}, \mathbf{u}^{k}, ..\mathbf{u}^{k-nu+1}) \le 0 \end{bmatrix} \vee \begin{bmatrix} \neg z_{\text{dual}} \\ \mathbf{e}^{-}(\mathbf{u}, \mathbf{u}^{k}, ..\mathbf{u}^{k-nu+1}) \le 0 \end{bmatrix} \tag{3}$$

where the additional Boolean variable z_{dual} is used to select one of the feasible regions constrained by functions $\mathbf{e}^{+}$ and $\mathbf{e}^{-}$. The only continuous variables in this disjunction are the future inputs $\mathbf{u}$, while the current inputs $\mathbf{u}^{k}$ and the past inputs $\mathbf{u}^{k-1}...\mathbf{u}^{k-nu+1}$ are parameters that define the feasible regions.

The usual practice with nonlinear models is to solve the NLP problem for each region, and then to apply the best feasible solution. If disjunctions are used for other purposes in the model, dual control can also be included as a disjunction, in order to solve the entire problem in a single GDP or MINLP model.

3.5. Minimum Changes. An RTO implementation may be expected to avoid excessive variability. In other cases, some of the inputs may be manually modified by operators, who will probably not want to apply small movements in process inputs. For these situations, RTO can include a constraint on the minimum absolute value of the change allowed in certain process inputs $\mathbf{u}$, which can alternatively keep their current values:

$$\begin{aligned} &[u_i \le u_i^k - \Delta u_{i,\min}] \vee [u_i \ge u_i^k + \Delta u_{i,\min}] \vee [u_i = u_i^k] \\ &\quad i = 1, ..., nu;\ \Delta u_{i,\min} \ge 0 \end{aligned} \tag{4}$$

If the dual control constraints of section 3.4 are also included, the current operating point is infeasible. Moreover, the hyperplane determined by the last *nu* points is not part of the feasible region. Although this is not incompatible with the disjunction of eq 4, an input will not be allowed to stay on the same point for more than *nu* RTO cycles.

To demonstrate this intuitive fact, consider the hyperplane that contains the last *nu* points, defined by equation $\mathbf{a}^{\mathrm{T}}\cdot\mathbf{u} = d$, where $\mathbf{a}$ and d are constant coefficients. These coefficients can be obtained by solving the following equation:

$$\begin{bmatrix} \mathbf{u}^{k} & \mathbf{u}^{k-1} & \cdots & \mathbf{u}^{k-nu-1} \end{bmatrix}^{\mathrm{T}}\cdot\mathbf{a} = d\cdot\mathbf{1} \tag{5}$$

Because the constraints of eq 3 have been applied in previous RTO cycles, $\mathbf{u}^{k}$ does not belong to the hyperplane defined by vectors $(\mathbf{u}^{k-1}, ..., \mathbf{u}^{k-nu})$, and, consequently, it does not belong to the subspace defined by $(\mathbf{u}^{k-1}, ..., \mathbf{u}^{k-nu-1})$. Similarly, $\mathbf{u}^{k-1}$ does not belong to the subspace defined by $(\mathbf{u}^{k-2}, ..., \mathbf{u}^{k-nu-1})$, and the same conclusion can be obtained for the rest of the vectors forming the matrix of eq 5. Therefore, this matrix is of rank *nu* and eq 5 can be solved.

Suppose that element u_1 has the same value in the past *nu* cycles, i.e., $u_1^{k} = u_1^{k-1} = ... = u_1^{k-nu-1}$. Then, the first row of the matrix of eq 5 can be subtracted from all of the other rows, and the following system of equations is obtained:

$$\begin{aligned} &a_2\cdot(u_2^{k-1} - u_2^{k}) + \cdots + a_{nu}\cdot(u_{nu}^{k-1} - u_{nu}^{k}) = 0 \\ &\qquad\vdots \\ &a_2\cdot(u_2^{k-nu-1} - u_2^{k}) + \cdots + a_{nu}\cdot(u_{nu}^{k-nu-1} - u_{nu}^{k}) = 0 \end{aligned} \tag{6}$$

Since this system was obtained by linear combination of rows from eq 5, which was of rank *nu*, eq 6 has a unique solution, i.e., $a_2 = a_3 = \cdots = a_{nu} = 0$. Therefore, the hyperplane defined by the last *nu* points is

$$u_1 = u_1^{k} \tag{7}$$

which is the last disjunction of last term of the disjunction of eq 4, for $i = 1$. Since dual control forces $\mathbf{u}$ to be outside the mentioned hyperplane, this option is infeasible and the system will be forced to select between the two other alternatives of eq 4. The same can be demonstrated for any other element u_i of vector $\mathbf{u}$.

4. REAL-TIME OPTIMIZATION INCLUDING DISJUNCTIONS

4.1. Problem Statement. A real-time optimization problem including disjunctions can be stated as follows:

$$\min_{\mathbf{u}} Q(\mathbf{y}, \mathbf{u}) + \sum_{d\in D} c_d \tag{8}$$

subject to

$$\mathbf{h}_G(\mathbf{y}, \mathbf{u}) = 0$$

$$\mathbf{g}_G(\mathbf{y}, \mathbf{u}) \le 0$$

$$\bigvee_{i\in N_d} \begin{bmatrix} z_{id} \\ \mathbf{h}_{id}(\mathbf{y}, \mathbf{u}) = 0 \\ \mathbf{g}_{id}(\mathbf{y}, \mathbf{u}) \le 0 \\ c_d = \gamma_{id} \end{bmatrix}, d \in D$$

$$\mathbf{\Omega}(\mathbf{z}) = \text{true}$$

$$\mathbf{u}^{\mathrm{L}} \le \mathbf{u} \le \mathbf{u}^{\mathrm{U}},\ z_{id} \in \{\text{true, false}\}$$

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

where $\mathbf{u}$ are the process inputs, $\mathbf{y}$ the process outputs, and $\mathbf{z}$ the logic variables corresponding to the nd disjunctions. "Global" equality and inequality constraints ($\mathbf{h}_G$ and $\mathbf{g}_G$ respectively) are valid, regardless of which disjunction terms are active (i.e., which variables z_{id} are true). In contrast, $\mathbf{h}_{id}$ and $\mathbf{g}_{id}$ are only valid if z_{id} is true. $\boldsymbol{\Omega}$ are the logical constraints.

The real functionality of the process outputs, with respect to the inputs, cannot be known in practice. Instead of this, a process model is available to predict (approximately) functions $\mathbf{h}_G$ and $\mathbf{h}_{id}$:

$$h_G(\mathbf{y}, \mathbf{u}) \approx \mathbf{f}_G(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_G) = 0$$
$$\bigvee_{i \in N_d} \begin{bmatrix} z_{id} \\ \mathbf{h}_{id}(\mathbf{y}, \mathbf{u}) \approx \mathbf{f}_{id}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{id}) = 0 \end{bmatrix}, d \in D \tag{9}$$

where $\boldsymbol{\theta}_G$ and $\boldsymbol{\theta}_{id}$ are adjustable parameters. The process states (commonly named as $\mathbf{x}$) appear implicitly in eq 9 and they will not appear explicitly along this work for the sake of simplicity.

4.2. Adaptation Strategies. *4.2.1. Two-Step Approach.* The adjustable parameters can be estimated at the beginning of each RTO cycle k using measured outputs $\mathbf{y}^k$:

$$\boldsymbol{\theta}_G^k, \boldsymbol{\theta}_{id}^k = \arg\min_{\theta} (\mathbf{y}^k - \mathbf{y})^{\mathrm{T}} \cdot (\mathbf{y}^k - \mathbf{y}) \tag{10}$$

subject to

$$\mathbf{f}_G(\mathbf{y}, \mathbf{u}^k, \boldsymbol{\theta}_G) = 0$$
$$\mathbf{f}_{id}(\mathbf{y}, \mathbf{u}^k, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{id}) = 0 \qquad i \in N_d,\ d \in D/z_{id}^k = \text{true}$$
$$\boldsymbol{\theta}_{id} = \boldsymbol{\theta}_{id}^{k-1} \qquad i \in N_d,\ d \in D/z_{id}^k = \text{false}$$

A common practice to reduce the effect of noise in the variance of the estimated parameters is to filter the results using the values from previous cycles. Among the filtering techniques, the exponential filter is widely used. An implementation for mixed integer real-time optimization can be of the following form:

$$\boldsymbol{\theta}_G^{f,k} = K_G \boldsymbol{\theta}_G^k + (I_G - K_G)\boldsymbol{\theta}_G^{k-1}$$
$$\boldsymbol{\theta}_{id}^{f,k} = K_{id} \boldsymbol{\theta}_{id}^k + (I_{id} - K_{id})\boldsymbol{\theta}_{id}^{k-1}$$
$$\forall\ i \in N_d,\ d \in D/z_d^k = \text{true} \tag{11}$$

where $\boldsymbol{\theta}_G^{f,k}$ and $\boldsymbol{\theta}_{id}^{f,k}$ are the filtered updated parameters, K_G and K_{id} are filtering matrices, and I_G and I_{id} are identity matrices of consistent dimension.

The optimization problem to be solved is therefore formulated as follows:

$$\min_{\mathbf{u}} Q(\mathbf{y}, \mathbf{u}) + \sum_{d \in D} c_d \tag{12}$$

subject to

$$\mathbf{f}_G(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_G^{f,k}) = 0$$
$$\mathbf{g}_G(\mathbf{y}, \mathbf{u}) \le 0$$
$$\bigvee_{i \in N_d} \begin{bmatrix} z_{id} \\ \mathbf{f}_{id}(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_G^{f,k}, \boldsymbol{\theta}_{id}^{f,k}) = 0 \\ \mathbf{g}_{id}(\mathbf{y}, \mathbf{u}) \le 0 \\ c_d = \gamma_{id} \end{bmatrix}, d \in D$$
$$\boldsymbol{\Omega}(\mathbf{z}) \le 0$$
$$\mathbf{u}_l \le \mathbf{u} \le \mathbf{u}_u,\ z_{id} \in \{\text{true}, \text{false}\}$$

4.2.2. Modifier Adaptation. Modifier adaptation techniques avoid the first optimization step for parameter estimation. Instead of this, they keep the model parameters constant, adapting the model constraints with a bias, and correcting the cost and model gradients at RTO cycle k as follows:

$$\begin{aligned}
&\boldsymbol{\beta}_G^k = \mathbf{g}_G(\mathbf{y}^k, \mathbf{u}^k) - \mathbf{g}_G(\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta}), \mathbf{u}^k) \\
&\boldsymbol{\lambda}_G^k = \nabla_{\mathbf{u}} \mathbf{g}_G^{real}(\mathbf{y}^k, \mathbf{u}^k) - \nabla_{\mathbf{u}} \mathbf{g}_G(\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta}), \mathbf{u}^k) \\
&\boldsymbol{\lambda}_Q^k = \nabla_{\mathbf{u}} Q^{real}(\mathbf{y}^k, \mathbf{u}^k) - \nabla_{\mathbf{u}} Q(\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta}), \mathbf{u}^k) \\
&\left.\begin{aligned}
&\boldsymbol{\beta}_{id}^k = \mathbf{g}_{id}(\mathbf{y}^k, \mathbf{u}^k) - \mathbf{g}_{id}(\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta}), \mathbf{u}^k) \\
&\boldsymbol{\lambda}_{id}^k = \nabla_{\mathbf{u}} \mathbf{g}_{id}^{real}(\mathbf{y}^k, \mathbf{u}^k) - \nabla_{\mathbf{u}} \mathbf{g}_{id}(\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta}), \mathbf{u}^k) \\
&\mathbf{u}_{id}^{\mathrm{ref}} = \mathbf{u}^k
\end{aligned}\right\} \quad \forall\ i \in N_d,\ d \in D/z_{id}^k = \text{true} \\
&\left.\begin{aligned}
&\boldsymbol{\beta}_{id}^k = \boldsymbol{\beta}_{id}^{k-1} \\
&\boldsymbol{\lambda}_{id}^k = \boldsymbol{\lambda}_{id}^{k-1}
\end{aligned}\right\} \quad \forall\ i \in N_d,\ d \in D/z_{id}^k = \text{false}
\end{aligned} \tag{13}$$

where $\boldsymbol{\beta}_G^k$ are the global constraint modifiers, $\boldsymbol{\lambda}_G^k$ the global constraint gradient modifiers, and $\boldsymbol{\lambda}_Q^k$ the cost gradient modifier. $\boldsymbol{\beta}_{id}^k$ and $\boldsymbol{\lambda}_{id}^k$ are the constraint modifiers and gradient modifiers corresponding to term $i \in N_d$, $d \in D$. The superindex *real* is used for the true gradient obtained from process data. The outputs obtained using the process model are expressed as $\mathbf{y}(\mathbf{u}^k, \boldsymbol{\theta})$. The global constraint and cost gradient modifiers are used for a linear correction around the current inputs $\mathbf{u}^k$; instead, for gradient modifiers in disjunctions, $\boldsymbol{\lambda}_{id}^k$, the linear correction is built around inputs $\mathbf{u}_{id}^{\mathrm{ref}}$, which are only updated to the current inputs when the Boolean variable z_{id}^k = true.

The modifiers can be smoothed using an exponential filter, as is indicated in section 4.2.1. The expressions are not shown here to avoid repetition; they can be obtained in a straightforward way from eq 11.

Once the modifiers are calculated with eq 13, the corrected objective function and constraints are obtained as follows:

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

$$\begin{aligned}
&\min\ Q(\mathbf{y}, \mathbf{u}) + \sum_{d\in D} c_d + (\boldsymbol{\lambda}_Q^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}^k)\\
&\mathbf{g}_G(\mathbf{y}, \mathbf{u}) + \boldsymbol{\beta}_G^k + (\boldsymbol{\lambda}_G^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}^k) \le 0\\
&\mathbf{g}_{id}(\mathbf{y}, \mathbf{u}) + \boldsymbol{\beta}_{id}^k + (\boldsymbol{\lambda}_{id}^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}_{id}^{\mathrm{ref}}) \le 0 \qquad i \in N_d,\ d \in D
\end{aligned} \tag{14}$$

The adapted RTO problem can then be formulated as follows:

$$\min\ Q(\mathbf{y}, \mathbf{u}) + \sum_{d\in D} c_d + (\boldsymbol{\lambda}_Q^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}^k) \tag{15}$$

subject to

$$\mathbf{f}_G(\mathbf{y}, \mathbf{u}, \boldsymbol{\theta}_G^k) = 0$$

$$\mathbf{g}_G(\mathbf{y}, \mathbf{u}) + \boldsymbol{\beta}_G^k + (\boldsymbol{\lambda}_G^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}^k) \le 0$$

$$\bigvee_{i\in N_d}\begin{bmatrix} z_{id}\\ \mathbf{f}_{id}(\mathbf{y}, \mathbf{u}^k, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{id}) = 0\\ \mathbf{g}_{id}(\mathbf{y}, \mathbf{u}) + \boldsymbol{\beta}_{id}^k + (\boldsymbol{\lambda}_{id}^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}_{id}^{\mathrm{ref}}) \le 0\\ c_d = \gamma_{id}\end{bmatrix}, d \in D$$

$$\begin{bmatrix} z_{\mathrm{dual}}\\ \mathbf{e}^+(\mathbf{u}) \le 0\end{bmatrix} \vee \begin{bmatrix} \neg z_{\mathrm{dual}}\\ \mathbf{e}^-(\mathbf{u}) \le 0\end{bmatrix}$$

$$\boldsymbol{\Omega}(\mathbf{z}, z_{\mathrm{dual}}) \le 0$$

$$\mathbf{u}^{\mathrm{L}} \le \mathbf{u} \le \mathbf{u}^{\mathrm{U}},\ z_{id} \in \{\mathrm{true}, \mathrm{false}\},\ z_{\mathrm{dual}} \in \{\mathrm{true}, \mathrm{false}\}$$

where the additional disjunction z_{dual} generates the two disjoint feasible regions by constraints $\mathbf{e}^+$ and $\mathbf{e}^-$, described in section 3.4. The gradient modifier update in disjunctions d can lead to more dual constraints, based on the current and past values of the inputs $\mathbf{u}_{id}^{\mathrm{ref}}$. If this happens, each disjunction d can be reformulated adding a disjunction inside of it, defined by the Boolean variable $z_{id,\mathrm{dual}}$:

$$\bigvee_{i\in N_d}\begin{bmatrix} z_{id}\\ \mathbf{f}_{id}(\mathbf{y}, \mathbf{u}^k, \boldsymbol{\theta}_G, \boldsymbol{\theta}_{id}) = 0\\ \mathbf{g}_{id}(\mathbf{y}, \mathbf{u}) + \boldsymbol{\beta}_{id}^k + (\boldsymbol{\lambda}_{id}^k)^{\mathrm{T}}\cdot(\mathbf{u} - \mathbf{u}_{id}^{\mathrm{ref}}) \le 0\\ c_d = \gamma_{id}\\ \begin{bmatrix} z_{id,\mathrm{dual}}\\ \mathbf{e}_{id}{}^+(\mathbf{u}) \le 0\end{bmatrix} \vee \begin{bmatrix} \neg z_{id,\mathrm{dual}}\\ \mathbf{e}_{id}{}^-(\mathbf{u}) \le 0\end{bmatrix}\end{bmatrix}, d \in D \tag{16}$$

Equation 16 shows a possible structure of disjunctions in a dual modifier adaptation strategy; a detailed study of dual strategies is beyond the scope of this work.

As mentioned in section 3.4, it can be remarked that dual constraints only appear when gradient estimation is performed using data from the steady states reached after each RTO cycle. If other methods are used for gradient estimation,[21,25] these constraints may not be necessary or should be formulated in a different way.

A variant of the modifier adaptation strategy modifies only the equations of the model that are not known with certainty, and estimates each gradient only with respect to a subset of the inputs or a linear combination of them.[24] The interested reader is referred to Supporting Information to find an extension of this strategy when disjunctions are present.

4.3. Modifier Adaptation of Piecewise Functions. If the piecewise function given by eq 2 has C^0 continuity, the adaptation strategy must keep this property. In the simplest case, a biasing (constraint adaptation) strategy should add the same bias to all the pieces of the function. If a linear (gradient) modifier is included in the current piece (i.e., the piece corresponding to the current operating point), two options are possible: either (i) use the same value of the gradient modifier to all the pieces, or (ii) change the constraint modifier in the other pieces to keep C^0 continuity.

The condition for C^0 continuity in the limit of two pieces is

$$\begin{aligned}
&g_p(u_{p+1}^{\mathrm{L}}) + \beta_p^k + \lambda_p^k\cdot(u_{p+1}^{\mathrm{L}} - u_p^{\mathrm{ref}})\\
&\quad = g_{p+1}(u_{p+1}^{\mathrm{L}}) + \beta_{p+1}^k + \lambda_{p+1}^k\cdot(u_{p+1}^{\mathrm{L}} - u_{p+1}^{\mathrm{ref}})
\end{aligned} \tag{17}$$

As the original model was also C^0 continuous,

$$g_p(u_{p+1}^{\mathrm{L}}) = g_{p+1}(u_{p+1}^{\mathrm{L}}) \tag{18}$$

Therefore, eq 17 implies

$$\beta_p^k + \lambda_p^k\cdot(u_{p+1}^{\mathrm{L}} - u_{p+1}^{\mathrm{ref}}) = \beta_{p+1}^k + \lambda_{p+1}^k\cdot(u_{p+1}^{\mathrm{L}} - u_{p+1}^{\mathrm{ref}}) \tag{19}$$

Once the constraint and gradient correction terms, $\beta_{p_{\mathrm{act}}^k}^k$ and $\lambda_{p_{\mathrm{act}}^k}^k$ respectively, have been updated for the active piece p_{act}^k, using eq 13, the modifiers for all of the other pieces can be updated using the following procedure:

FOR $p := (p_{\mathrm{act}}^k - 1)$ to 1

$$\lambda_p^k = \lambda_p^{k-1}$$

$$\begin{aligned}
\beta_p^k = {}& \beta_{p+1}^k + \lambda_{p+1}^k\cdot(u_{p+1}^{\mathrm{L}} - u_{p+1}^{\mathrm{ref}})\\
&- \lambda_p^k\cdot(u_{p+1}^{\mathrm{L}} - u_p^{\mathrm{ref}})
\end{aligned}$$

LOOP

FOR $p := (p_{\mathrm{act}}^k + 1)$ to np

$$\lambda_p^k = \lambda_p^{k-1}$$

$$\begin{aligned}
\beta_p^k = {}& \beta_{p-1}^k + \lambda_{p-1}^k\cdot(u_p^{\mathrm{L}} - u_{p-1}^{\mathrm{ref}})\\
&- \lambda_p^k\cdot(u_p^{\mathrm{L}} - u_p^{\mathrm{ref}})
\end{aligned}$$

LOOP (20)

5. REVISION AND REFORMULATION OF COMMON RTO PRACTICES

The adaptation strategies presented in section 4.2 generate a model that can be trusted only in the vicinity of the current operating point (defined by the input set $\mathbf{u}^k$). If the new point $\mathbf{u}^{k+1}$ falls in a region where the adapted model is unable to predict the plant values accurately, the application of these inputs to the plant can result in constraint violations or worsening of the objective function.

In practice, many RTO applications apply a strategy to keep the next point $\mathbf{u}^{k+1}$ in the valid region around $\mathbf{u}^k$. Some strategies modify the optimization problem by adding constraints that limit the maximum change in input values, or by penalizing this change

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

in the objective function. Another approach, known as *input filter*, solves the optimization problem to find a new optimum $\mathbf{u}^*$, and then selects the new point by doing a step in the direction of $(\mathbf{u}^* - \mathbf{u}^k)$.

These strategies cannot be applied directly to an RTO system with disjunctions. An extension to each of these strategies (changes in the optimization problem and input filter) is proposed and discussed in the next subsections.

5.1. Input Changes Constrained or Penalized in the Optimization Problem. A common practice in RTO is to limit the change in process inputs by adding a constraint in the optimization problem:[2]

$$\mathbf{u}^k - \Delta\mathbf{u}_{\max} \le \mathbf{u} \le \mathbf{u}^k + \Delta\mathbf{u}_{\max} \tag{21}$$

Brdys and Tatjewski[27] proposed a convexifying term that is added to the objective function in order to penalize changes (and, additionally, to make the objective function strictly convex if the rest of the terms are linear). The convexifying term C_T is defined as

$$C_T = \rho \| \mathbf{u} - \mathbf{u}^k \|^2 \tag{22}$$

where ρ is a parameter selected by the user. Smaller values of ρ allow larger changes in the inputs, while bigger values of ρ produce more robust results but a slower convergence.

Some dual control implementations can also limit the maximum change in process inputs by setting feasible regions with bounds in noise and truncation errors in gradient estimation.[11,40]

If the RTO model includes disjunctions which involve fixed costs or disjoint feasible regions, these strategies may affect the convergence to the real plant optimum (and even to the model optimum). This can be shown through a simple example. The following optimization problem is solved:

$$\min f = 10 - 0.4x + \gamma \tag{23}$$

subject to

$$\begin{bmatrix} z \\ x \ge 5 \\ \gamma = 3.5 + 0.05x \end{bmatrix} \vee \begin{bmatrix} \neg z \\ x \le 5 \\ \gamma = 0 \end{bmatrix}$$

$$2 \le x \le 10$$

It is assumed that eq 23 is a perfect model (i.e., it describes the system to optimize without structural or parametric mismatch). The initial point is $x^0 = 8$.

A constraint in the maximum change is added, as proposed by eq 21:

$$-1 \le x - x^k \le 1 \tag{24}$$

Alternatively, the following convexifying term is added to the objective function, as proposed by eq 22:

$$C_T = 0.3(x - x^k)^2 \tag{25}$$

The optimal solution of eq 23 is ($x = 5, f = 8$). It can be said that an RTO system using eq 23 converges to the optimum in one cycle (there is no mismatch and the result is not filtered). If constraint of eq 24 is included, the system converges within two cycles to a local optimum ($x = 10, f = 10$). If eq 25 is added to the objective function, the system converges to the same local optimum within four RTO cycles. Figure 1 illustrates these results.

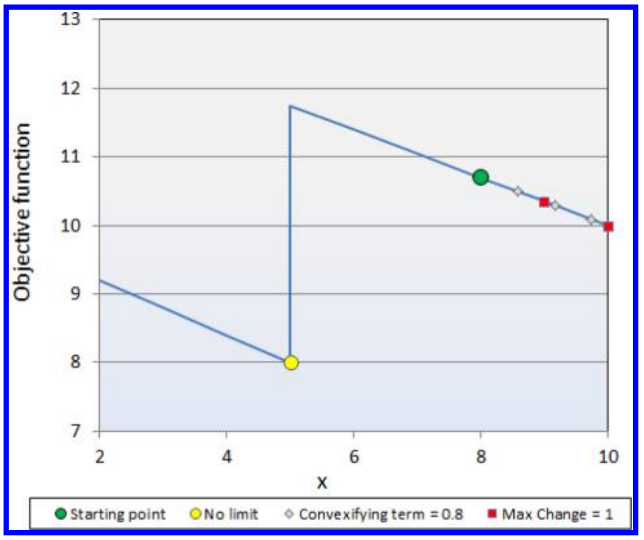

Figure 1. Convergence using maximum change constraints and convexifying terms.

To avoid these suboptimal solutions, the constraint in the maximum changes or the penalty can be included only if the optimal value and the current value of the disjunction are the same (i.e., if $z = z^k$). A change in disjunctions can be penalized with a fixed cost as a way to do the change only if the model predicts a significant benefit by doing it. For example, eq 26 can be written for a convexifying cost constraint:

$$\begin{bmatrix} z \\ \mathbf{g}_1(\mathbf{u}) \le 0 \\ C_T = z^k \rho \| \mathbf{u} - \mathbf{u}^k \|^2 + (1 - z^k)\pi_1 \end{bmatrix} \vee \begin{bmatrix} \neg z \\ \mathbf{g}_2(\mathbf{u}) \le 0 \\ C_T = (1 - z^k)\rho \| \mathbf{u} - \mathbf{u}^k \|^2 + z^k \pi_2 \end{bmatrix} \tag{26}$$

where $\mathbf{g}_1$ and $\mathbf{g}_2$ are the constraints that are valid if z = true and if z = false, respectively. π_1 and π_2 are the fixed costs that penalize a change in z from true to false and from false to true, respectively. In the example of eq 23, adding the disjunction of eq 26, the RTO system converges to the optimum in one iteration if the penalizing costs π_1 and π_2 are <2; otherwise, it converges to the local optimum.

5.2. Input Filter. As mentioned, another common practice in RTO is known as *input filtering*. The optimal results $\mathbf{u}^*$ obtained from solving the optimization problem at cycle k are filtered with the purpose of gaining robustness and helping the convergence to the real plant optimum:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + K \cdot (\mathbf{u}^* - \mathbf{u}^k) \qquad 0 \le K \le 1 \tag{27}$$

Bunin et al.[28] have proposed upper bounds for the filtering gain K, which are based on Lipschitz constants of the constraints and ensure feasibility for RTO with continuous problems.

If disjunctions are present in the RTO model, input filtering can lead to suboptimal or infeasible solutions. This undesired behavior can appear when disjunctions include fixed costs or generate disjoint feasible regions, as is analyzed in the following paragraphs.

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

When disjoint feasible regions are present (see section 3.3), the input filter can lead to inputs outside the feasible region. For example, consider the following problem:

$$\min f = (x-4)^2 \tag{28}$$

subject to

$$\begin{bmatrix} z \\ x \geq 4.5 \end{bmatrix} \vee \begin{bmatrix} \neg z \\ x \leq 7 \end{bmatrix}$$
$$3 \leq x \leq 8$$

A starting point of $x^0 = 8$ and an input filter of $K = 0.4$ are selected. Figure 2 shows the evolution of x and f. It can be observed that four intermediate points are infeasible.

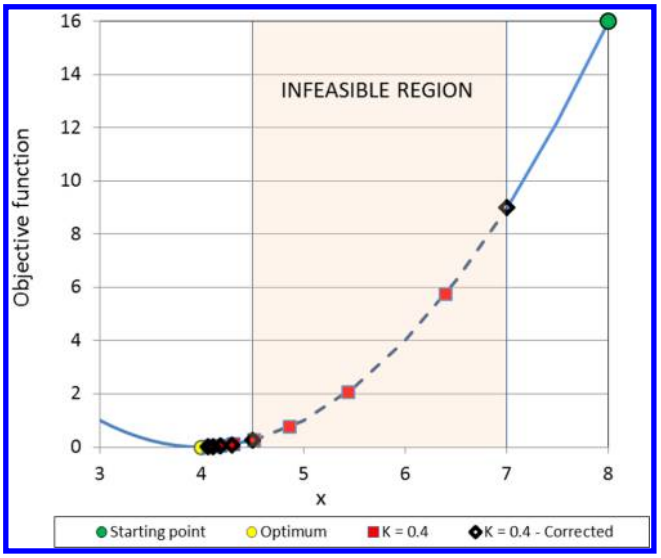

Figure 2. Input filter and disjoint feasible regions.

If the constraints are hard (i.e., temporary violations are not tolerated), the input filter must not allow this violation. Considering a disjunction that generates disjoint feasible regions:

$$\begin{bmatrix} z_d \\ \mathbf{g}_{1d}(\mathbf{u}) \leq 0 \\ c_d = \gamma_d \end{bmatrix} \vee \begin{bmatrix} \neg z_d \\ \mathbf{g}_{2d}(\mathbf{u}) \leq 0 \\ c_d = 0 \end{bmatrix} \tag{29}$$

Upper and lower bounds in the input filter can be obtained for convex constraints:

$$\left.\begin{aligned} &K_{U,d}^{\max} = \arg\max_K K \\ &\text{subject to} \\ &\mathbf{u}^{k+1} = \mathbf{u}^k + K(\mathbf{u}^* - \mathbf{u}^k) \\ &\mathbf{g}_{1d}(\mathbf{u}^{k+1}) \leq 0 \qquad \text{if } (z_d^k) \wedge (\neg z_d^*) \\ &\mathbf{g}_{2d}(\mathbf{u}^{k+1}) \leq 0 \qquad \text{if } (\neg z_d^k) \wedge (z_d^*) \end{aligned}\right\} \forall\, d \in D_U \tag{30}$$

$$\left.\begin{aligned} &K_{U,d}^{\min} = \arg\min_K K \\ &\text{subject to} \\ &\mathbf{u}^{k+1} = \mathbf{u}^k + K(\mathbf{u}^* - \mathbf{u}^k) \\ &\mathbf{g}_{1d}(\mathbf{u}^{k+1}) \leq 0 \qquad \text{if } (\neg z_d^k) \wedge (z_d^*) \\ &\mathbf{g}_{2d}(\mathbf{u}^{k+1}) \leq 0 \qquad \text{if } (z_d^k) \wedge (\neg z_d^*) \end{aligned}\right\} \forall\, d \in D_U \tag{31}$$

where D_U is the subset of constraints that generate disjoint feasible regions. If constraints are convex, for z_d^k = true and z_d^* = false:

$$\begin{aligned} &\mathbf{g}_{1d}(\mathbf{u}^k) \leq 0 \\ &\mathbf{g}_{1d}(\mathbf{u}^k + K_{U,d}^{\max}(\mathbf{u}^* + \mathbf{u}^k)) \leq 0 \end{aligned} \tag{32}$$

Equation 32 implies that

$$\mathbf{g}_{1d}(\mathbf{u}^k + K(\mathbf{u}^* + \mathbf{u}^k)) \leq 0 \qquad \forall\, 0 \leq K \leq K_{U,d}^{\max} \tag{33}$$

At the same time,

$$\begin{aligned} &\mathbf{g}_{2d}(\mathbf{u}^*) \leq 0 \\ &\mathbf{g}_{2d}(\mathbf{u}^k + K_{U,d}^{\min}(\mathbf{u}^* + \mathbf{u}^k)) \leq 0 \end{aligned} \tag{34}$$

Equation 34 implies that

$$\mathbf{g}_{1d}(\mathbf{u}^k + K(\mathbf{u}^* + \mathbf{u}^k)) \leq 0 \qquad \forall\, K_{U,d}^{\min} \leq K \leq 1 \tag{35}$$

A similar proof can be obtained for z_d^k = false and z_d^* = true to show that all values of K greater than $K_{U,d}^{\min}$ or lower than $K_{U,d}^{\max}$ are feasible for the constraints contained in disjunction d.

As result of eqs 30 and 31, the following condition can be obtained for K:

$$(K \leq K_{U,d}^{\max}) \vee (K \geq K_{U,d}^{\min}) \tag{36}$$

Equation 36 provides two possible ranges for K. If the option $(K \leq K_{U,d}^{\max})$ is always selected, the system may get stuck in one value of the disjunction, while the optimization results are indicating that the optimum can be achieved by changing the value of the disjunction. If the option $(K \geq K_{U,d}^{\min})$ is always selected, it can lead to a less-robust solution. A practical criterion to select between the two options is to use a maximum ratio $r_{\max}$ for $K_{U,d}^{\min}/K_{U,d}^{\max}$. For a single disjunction d:

$$\begin{aligned} &\text{IF } (K_{U,d}^{\min}/K_{U,d}^{\max} \leq r_{\max}) \text{ THEN } K \leq K_{U,d}^{\max} \\ &\text{ELSE } K \geq K_{U,d}^{\min} \end{aligned} \tag{37}$$

Figure 2 shows the evolution obtained by setting $r_{\max} = 10$. It can be observed that the RTO system never selects an input that falls in the infeasible region.

When the disjunctions include fixed costs, an RTO system with an input filter can converge to a suboptimal operating point. An example of this behavior can be shown by solving eq 23 and applying the results with an input filter $K < 1$. Figure 3 shows the results for $K = 0.6$. The optimal inputs x^* after each cycle k are always the real optimal input ($x = 5$), but the filtering strategy prevents the system from reaching this point.

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

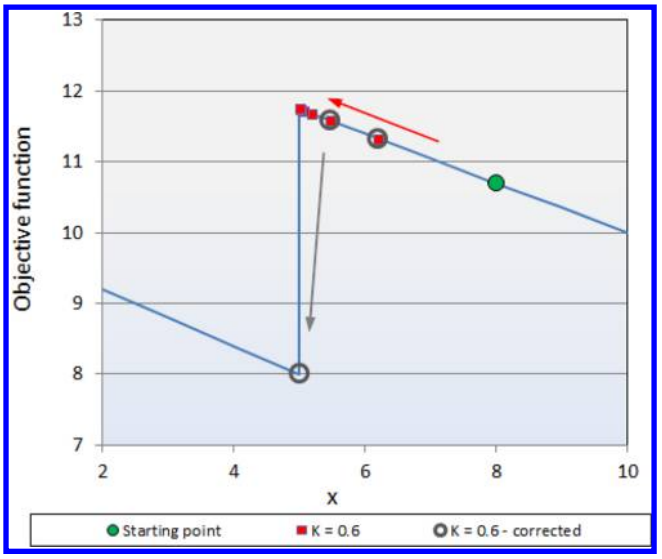

Figure 3. Convergence problems and corrected strategy for the input filter when fixed costs are present.

From a theoretical point of view, the example shown in Figure 3 converges to the maximum possible value of the objective function. In practice, after a certain number of iterations, x^k will be close enough to $x = 5$ (for example, $x^8 = 5.001$), and it could then be assumed that the cost is the optimal ($f = 8$). Nevertheless, the path followed until reaching the optimum leads to an increase in the objective function after each RTO cycle, and thus the value of the objective function can be worse than the starting point for several cycles (which, in the context of RTO, can demand several hours).

A possible solution for this problem is to define a minimum allowed change in the inputs when the optimal value of a Boolean variable associated with a disjunction, z_d^*, is different to its current value, z_d^k. If the filtered inputs generate a step $\|K\cdot(\mathbf{u}^* - \mathbf{u}^k)\|$ which is lower than the minimum allowed change, δ_d, then the Boolean variable z_d is forced to change its value.

First, for each disjunction considered as that of eq 29 (but not necessarily generating disjoint feasible regions), the maximum value of the filtering constant K that makes $z_d^{k+1} = z_d^k$ is calculated as

$$K_d^{\max} = \arg\max_K K \tag{38}$$

subject to

$$\mathbf{u}^{k+1} = \mathbf{u}^k + K(\mathbf{u}^* - \mathbf{u}^k)$$

$$\left.\begin{aligned}\mathbf{g}_{1d}(\mathbf{u}^{k+1}) \le 0\\ \mathbf{g}_{2d}(\mathbf{u}^{k+1}) \ge 0\end{aligned}\right\} \quad \text{if } (z_d^k) \wedge (\neg z_d^*)$$

$$\left.\begin{aligned}\mathbf{g}_{2d}(\mathbf{u}^{k+1}) \le 0\\ \mathbf{g}_{1d}(\mathbf{u}^{k+1}) \ge 0\end{aligned}\right\} \quad \text{if } (\neg z_d^k) \wedge (z_d^*)$$

If the change in $\mathbf{u}$ for $K_d^{\max}$ is lower than the minimum step δ_d (selected as an adjustable parameter), then z_d is forced to be the optimal z_d^*:

$$((z_d^k = \neg z_d^*) \wedge (\|K_d^{\max}\cdot(\mathbf{u}^* - \mathbf{u}^k)\| \le \delta_d)) \Rightarrow z_d^{k+1} = z_d^* \tag{39}$$

Having this result, a minimum value of K can be selected for each disjunction d:

$$\left.\begin{aligned}&K_d^{\min} = \arg\min_K K\\ &\text{subject to}\\ &\mathbf{u}^{k+1} = \mathbf{u}^k + K(\mathbf{u}^* - \mathbf{u}^k)\\ &\mathbf{g}_{1d}(\mathbf{u}^{k+1}) \le 0 \quad \text{if } (\|K_d^{\max}\cdot(\mathbf{u}^* - \mathbf{u}^k)\| \le \delta_d) \wedge (\neg z_d^k) \wedge (z_d^*)\\ &\mathbf{g}_{2d}(\mathbf{u}^{k+1}) \le 0 \quad \text{if } (\|K_d^{\max}\cdot(\mathbf{u}^* - \mathbf{u}^k)\| \le \delta_d) \wedge (z_d^k) \wedge (\neg z_d^*)\end{aligned}\right\} \forall\, d \in D \tag{40}$$

Then, a lower bound can be set for the input filter K:

$$K \ge \max_d (K_d^{\min}) \tag{41}$$

In eq 40, it can be remarked that if there is no change in the Boolean variable z_d or if the δ_d criterion is not fulfilled, then the value of $K_d^{\min}$ is 0 for the corresponding disjunction d.

Applying the δ_d criterion to the example given by eq 23, with a default value of $K = 0.6$, the system converges to the optimum within three RTO cycles if the minimum allowed change δ is set to 1.

If eqs 36 and 41 are combined, taking into account all the disjunctions generating disjoint feasible regions, it is possible to obtain a set V_K of possible values of K:

$$V_K = \{K / \bigwedge_{d\in D} (K \ge K_d^{\min}) \bigwedge_{d\in D_U} ((K \le K_{U,d}^{\max}) \vee (K \ge K_{U,d}^{\min}))\} \tag{42}$$

The criterion of eq 37 must be extended to be applied to the set V_K. The interested reader is referred to the Supporting Information in this article to determine a possible extension of this criterion.

The bounds proposed in eq 37 may be in conflict with other limits for the input filter that have been formulated in recent publications for nonlinear, continuous RTO.[28] At the same time, logical constraints among Boolean variables (eq 1) can reduce the size of the set V_K even more. Therefore, the problem of using the input filter when discrete decisions are involved should be investigated further.

6. CASE STUDY

The application of real-time optimization to systems that include discrete decisions is shown through a case study consisting of a generic system that includes three processes modeled by simple equations.

As usual in RTO literature,[28,41] the performance of RTO techniques is evaluated by using two models. The first model

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

(hereafter called the *real plant*) represents the real system. It is used to simulate the plant behavior, to evaluate the actual cost obtained by optimization and to provide the real outputs (called *measurements*) to the RTO system. The second model, called the *RTO model*, differs structurally and parametrically from the real plant. It is adapted using the available measurements, and used to find the (model) optimal set of inputs. These inputs are processed and validated (for example with an input filter) and applied to the real plant model, which is used to calculate the new steady state and to provide measurements for the next RTO cycle.

The generic process with three interconnected subprocesses is shown in Figure 4. The process input vector $\mathbf{u}$ is formed by the pair of flows (F_1, F_2) and the measured outputs are Q_1, Q_2, and Q_3.

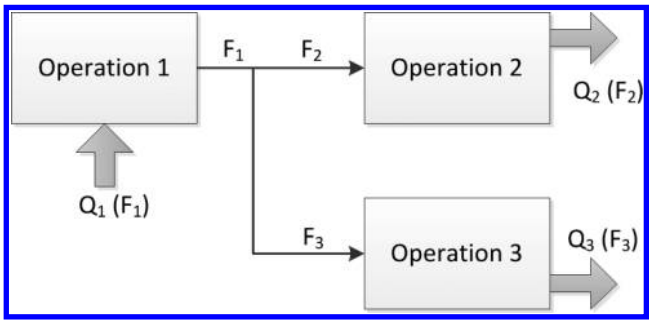

Figure 4. Process of the Case Study.

The functionality of the process outputs is only known approximately for the RTO model. Q_2 is modeled as a piecewise linear approximation. F_3 can be obtained from mass balance as the difference between F_1 and F_2.

Table 1 summarizes the real plant and the RTO model functions for all outputs Q.

A penalty cost C is added to the objective function if F_1 is greater than 100:

$$\begin{bmatrix} z_1 \\ F_1 \geq 100 \\ C = 100 + 0.1(F_1 - 100) \end{bmatrix} \vee \begin{bmatrix} \neg z_1 \\ F_1 \leq 100 \\ C = 0 \end{bmatrix} \tag{43}$$

The process can only operate for values of F_3 lower than 33 or greater than 37:

$$\begin{bmatrix} z_3 \\ F_3 \geq 37 \end{bmatrix} \vee \begin{bmatrix} \neg z_3 \\ F_3 \leq 33 \end{bmatrix} \tag{44}$$

A constraint is added on each RTO cycle k in order to generate information for gradient estimation. Defining the unitary vector $\mathbf{d}$, normal to the line generated by the two past inputs:

$$\mathbf{d}^k = \frac{[F_2^k - F_1^{k-1}, F_1^{k-1} - F_1^k]^T}{\|[F_2^k - F_2^{k-1}, F_1^{k-1} - F_1^k]\|} \tag{45}$$

A disjunction is added to the optimization model:

$$\begin{bmatrix} z_4 \\ [\mathbf{u} - \mathbf{u}^k]^T \cdot \mathbf{d}^k \geq 0.2 \end{bmatrix} \vee \begin{bmatrix} \neg z_4 \\ [\mathbf{u} - \mathbf{u}^k]^T \cdot \mathbf{d}^k \leq -0.2 \end{bmatrix} \tag{46}$$

In order to limit the changes in the inputs, a convexifying cost is included as proposed in section 5.1, with a penalty constant of $\rho = 0.05$. This penalty cost is active only if the optimal values of the Boolean variables of disjunctions 1 and 3 (i.e., z_1 and z_3) do not change with respect to its current value. Otherwise, the changes are not penalized:

$$\begin{bmatrix} z_5 \\ C_T = 0.05((F_1 - F_1^k)^2 + (F_2 - F_2^k)^2) \end{bmatrix} \vee \begin{bmatrix} \neg z_5 \\ C_T = 0 \end{bmatrix}$$
$$((z_1 = z_1^k) \wedge (z_3 = z_3^k)) \Rightarrow z_5 \tag{47}$$

Two modifier adaptation strategies are applied to the system. The first strategy is the simple but effective *constraint adaptation*, which does not include gradient correction terms but only a biasing value to correct the error between the predicted and the real outputs. In the second strategy, constraint and constraint gradient corrections are included, as presented in section 4.2.2.

The modifier adaptation is implemented following the procedure proposed by Rodger and Chachuat.[40] The modified equations are the equations that predict the outputs (i.e., those summarized in Table 1). First, the errors between the RTO model and the real plant outputs are calculated:

$$\begin{aligned} \beta_1^k &= Q_{1,real}^k - Q_1(\mathbf{u}^k) \\ \beta_2^k &= Q_{2,real}^k - Q_2(\mathbf{u}^k) \\ \beta_3^k &= Q_{3,real}^k - Q_3(\mathbf{u}^k) \end{aligned} \tag{48}$$

Then, in the gradient correction case, the gradients are estimated using a Broyden update:

$$\mathbf{BR}_i^k = \mathbf{BR}_i^{k-1} + \frac{Q_i^k - Q_i^{k-1} - (\mathbf{BR}_i^{k-1})^T \cdot (\mathbf{u}^k - \mathbf{u}^{k-1})}{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|^2}(\mathbf{u}^k - \mathbf{u}^{k-1}) \qquad i = 1, 3$$

$$\left.\begin{aligned} \mathbf{BR}_{2,1}^k &= \mathbf{BR}_{2,1}^{k-1} \frac{Q_2^k - Q_{2,1}^{\mathrm{ref}} - (\mathbf{BR}_{2,1}^{k-1})^T \cdot (\mathbf{u}^k - \mathbf{u}_{2,1}^{\mathrm{ref}})}{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|^2}(\mathbf{u}^k - \mathbf{u}_{2,1}^{\mathrm{ref}}) \\ \mathbf{BR}_{2,2}^k &= \mathbf{BR}_{2,2}^{k-1} \end{aligned}\right\} \quad \text{if } z_2 = \text{true}$$

$$\left.\begin{aligned} \mathbf{BR}_{2,2}^k &= \mathbf{BR}_{2,2}^{k-1} \frac{Q_2^k - Q_{2,2}^{\mathrm{ref}} - (\mathbf{BR}_{2,2}^{k-1})^T \cdot (\mathbf{u}^k - \mathbf{u}_{2,2}^{\mathrm{ref}})}{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|^2}(\mathbf{u}^k - \mathbf{u}_{2,2}^{\mathrm{ref}}) \\ \mathbf{BR}_{2,1}^k &= \mathbf{BR}_{2,1}^{k-1} \end{aligned}\right\} \quad \text{if } z_2 = \text{false} \tag{49}$$

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

Table 1. Real Plant and Model Functions

Q	real plant	RTO model
Q_1	$5\exp[-(F_1-80)/18]+60$	$-18.5+0.98F_1$
Q_2	$0.03F_2^2+0.04F_2-60$	$-177+3.79F_2 \quad F_2 \le 70$
		$-250+4.84F_2 \quad F_2 \le 70$
Q_3	$30(F_3-18)^{0.5}-20$	$-46.38+4.08F_3-0.001[(F_1-110)^2+(F_2-80)^2]$

where **BR** is the Broyden gradient estimator.[20,40] For output Q_2, the gradient is updated only for the currently active piece of the piecewise function. $Q_{2,1}^{ref}$ and $\mathbf{u}_{2,1}^{ref}$ are the values of Q_2 and $\mathbf{u}$ from the last time that piece 1 was active. $Q_{2,2}^{ref}$ and $\mathbf{u}_{2,2}^{ref}$ are defined similarly for piece 2. After that, the values of Q_2^{ref} and $\mathbf{u}_2^{ref}$ are updated for the active piece:

$$\left.\begin{aligned}\mathbf{u}_{2,1}^{ref}&=\mathbf{u}^k\\ Q_{2,1}^{ref}&=Q_2^k\end{aligned}\right\}\quad \text{if } z_2=\text{true}$$
$$\left.\begin{aligned}\mathbf{u}_{2,2}^{ref}&=\mathbf{u}^k\\ Q_{2,2}^{ref}&=Q_2^k\end{aligned}\right\}\quad \text{if } z_2=\text{false} \tag{50}$$

The gradient estimators from eq 49 are used to obtain the gradient modifiers according to eq 13. Using the knowledge of the system to optimize, some elements of the gradients can be bounded or neglected.[22,24] In this case study, it is already known that output Q_1 is only a function of F_1, and, therefore, the gradient with respect to F_2 is set to 0. In the same way, Q_2 is only a function of F_2, and the gradient with respect to F_1 is neglected in both pieces of the piecewise approximation. The rest of the gradients elements (partial derivatives) are bounded as

$$\left.\begin{aligned}\left\|\frac{\partial Q_p}{\partial u_i}\right\| &\le \frac{3}{2}\max_{\mathbf{u}}\left\|\frac{\partial Q_p^{real}}{\partial u_i}\right\|\\ \left\|\frac{\partial Q_p}{\partial u_i}\right\| &\ge \frac{2}{3}\min_{\mathbf{u}}\left\|\frac{\partial Q_p^{real}}{\partial u_i}\right\|\end{aligned}\right\}\quad \begin{aligned}i&=1,2\\ p&=1,2,3\end{aligned} \tag{51}$$

where the maximum and minimum values are searched in the feasible region of the RTO optimization problem.

In addition, all gradient modifiers are bounded to lie between a maximum value of 2 and a minimum value of −2.

After that, an exponential filter is applied to the modifiers, according to eq 11. The filtering constants (i.e., the elements of the diagonal of the filtering matrix) for the gradient modifiers λ are all equal to 0.7, while the filtering constants for constraint modifiers β are equal to 1 (no filter is applied).

Equation 20 is used to correct the piece that is not active in cycle k, in order to keep C^0 continuity.

Once the model is adapted, the following optimization problem is solved to estimate the optimal input set:

$$\min\ Q_1-Q_2-Q_3+C+C_T \tag{52}$$

subject to

$$F_1-F_2-F_3=0$$
$$Q_1=-18.5+0.98F_1+\beta_1^{f,k}+(\boldsymbol{\lambda}_1^{f,k})^T\cdot(\mathbf{u}-\mathbf{u}^k)$$
$$Q_3=-46.38+4.083F_3+\beta_3^{f,k}+(\boldsymbol{\lambda}_3^{f,k})^T\cdot(\mathbf{u}-\mathbf{u}^k)$$
$$\begin{bmatrix}z_1\\ F_1\ge 100\\ C=100+0.1(F_1-100)\end{bmatrix}\vee\begin{bmatrix}\neg z_1\\ F_1\ge 100\\ C=0\end{bmatrix}$$
$$\begin{bmatrix}z_{2,1}\\ F_2\ge 70\\ Q_2\le -250+4.84F_2+\beta_{21}^{f,k}+(\boldsymbol{\lambda}_{21}^{f,k})^T\cdot(\mathbf{u}-\mathbf{u}_{2,1}^{ref})\end{bmatrix}\vee\begin{bmatrix}z_{2,2}\\ F_2\ge 70\\ Q_2\le -177+3.79F_2+\beta_{22}^{f,k}+(\boldsymbol{\lambda}_{22}^{f,k})^T\\ \cdot(\mathbf{u}-\mathbf{u}_{2,2}^{ref})\end{bmatrix}$$
$$\begin{bmatrix}z_3\\ F_3\ge 37\end{bmatrix}\vee\begin{bmatrix}\neg z_3\\ F_3\le 33\end{bmatrix}$$
$$\begin{bmatrix}z_4\\ \begin{bmatrix}F_1-F_1^k\\ F_2-F_2^k\end{bmatrix}^T\cdot\mathbf{d}^k\ge 0.2\end{bmatrix}\vee\begin{bmatrix}\neg z_4\\ \begin{bmatrix}F_1-F_1^k\\ F_2-F_2^k\end{bmatrix}^T\cdot\mathbf{d}^k\le -0.2\end{bmatrix}$$
$$\begin{bmatrix}z_5\\ C_T=0.3((F_1-F_1^k)^2+(F_2-F_2^k)^2)\end{bmatrix}\vee\begin{bmatrix}\neg z_5\\ C_T=0\end{bmatrix}$$
$$((z_1=z_1^k)\wedge(z_3=z_3^k))\Rightarrow z_5$$
$$\begin{bmatrix}80\\55\\20\end{bmatrix}\le\begin{bmatrix}F_1\\F_2\\F_3\end{bmatrix}\le\begin{bmatrix}120\\90\\50\end{bmatrix}$$
$$\mathbf{z}\in\{\text{true, false}\}^5$$

The optimal inputs $(F_1^*, F_2^*)^T$ obtained by solving eq 52 are filtered with an initial filtering constant $K=0.6$:

$$(F_1^{k+1}, F_2^{k+1})^T=(F_1^k, F_2^k)^T+0.6(F_1^*-F_1^k, F_2^*-F_2^k)^T \tag{53}$$

Disjunction 1 includes a fixed cost; therefore, a fixed input filter may cause the convergence problems discussed in section 4.2 and shown in Figure 3. The strategy presented in section 4.2 is applied, selecting the parameter value $\delta_1=4$. When necessary, the minimum filtering value K_1^{min} is calculated using eq 40.

Disjunction 3 generates two disjoint feasible regions, which may lead to the behavior discussed in section 4.2 and shown in Figure 2. The strategy formulated in eqs 29−37 (extended in the Supporting Information in this work) is applied to avoid infeasible intermediate points, and the parameter r_{max} is set to 10.

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

The point ($F_1^0 = 110$, $F_2^0 = 80$) was selected as the initial point. The value of the objective function computed by the real plant model (i.e., with the true functionality of the outputs) at this point is −31.65. The real plant optimum is ($F_1^* = 100$, $F_2^* = 69.2$), and the optimal value of the objective function is −98.57. In order to provide information for the initial gradient estimation, the initial point is perturbed with small changes in F_1 and F_2. Then, the first two points are ($F_1^1 = 110.5$, $F_2^1 = 80$) and ($F_1^2 = 110$, $F_2^2 = 80.5$). The Broyden estimates are initialized with the model gradients.

Two scenarios are analyzed for each strategy. In the first, the outputs are "measured" perfectly (i.e., without errors or noise). In the second, Gaussian noise of mean 0 and variance 1.0 is added to the output data before performing the adaptation step.

The RTO model and the real plant model were implemented in GAMS[42] as MINLP problems, using a big-M type formulation, and solved using DICOPT, with XPRESS as MILP solver and CONOPT 3 as NLP solver.[43] The GAMS optimization step for the modifier adaptation strategy involves 16 equations and 14 variables (4 binary variables). For constraint adaptation, the MINLP model involves 14 equations and 13 variables (3 binary variables), as disjunction 4 is not included. Disjunction 5 did not require a binary variable to be modeled, while each of the rest of the disjunctions required one binary variable.

The RTO system was solved for 40 cycles, each cycle including the model update using measurements from the real plant model, modifiers calculation, optimization, and input filter. Each cycle required less than 1 s of CPU time in a 64-bit computer, with a 4-core 2.20 GHz processor and 8 GB of RAM memory.

Figures 5 and 6 show the evolution of inputs for the constraint adaptation and the modifier adaptation strategies. Figure 5 shows

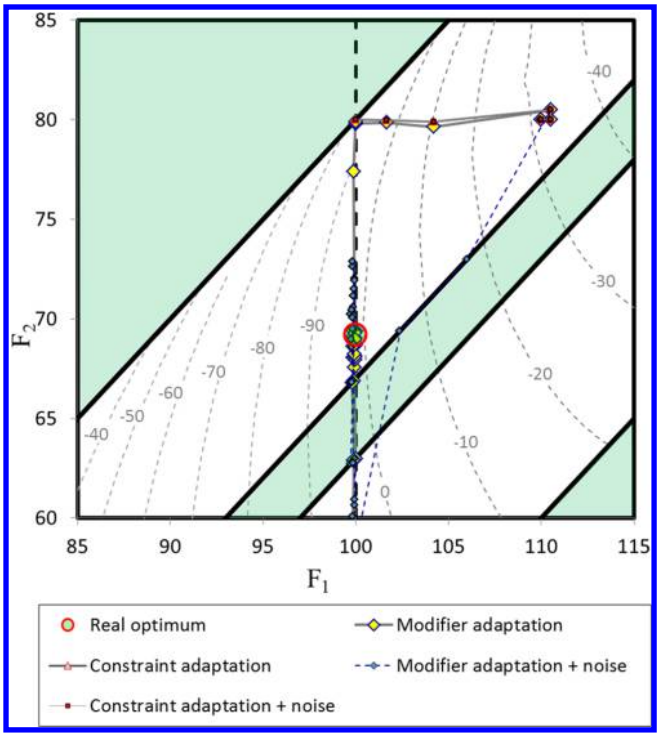

Figure 5. Evolution of inputs for constraint adaptation and modifier adaptation strategies. Initial point: ($F_1^0 = 110$, $F_2^0 = 80$).

the evolution in the space of the inputs, including level curves of the objective function. The infeasible regions are highlighted in green. Figure 6, instead, shows the input values vs the RTO cycle

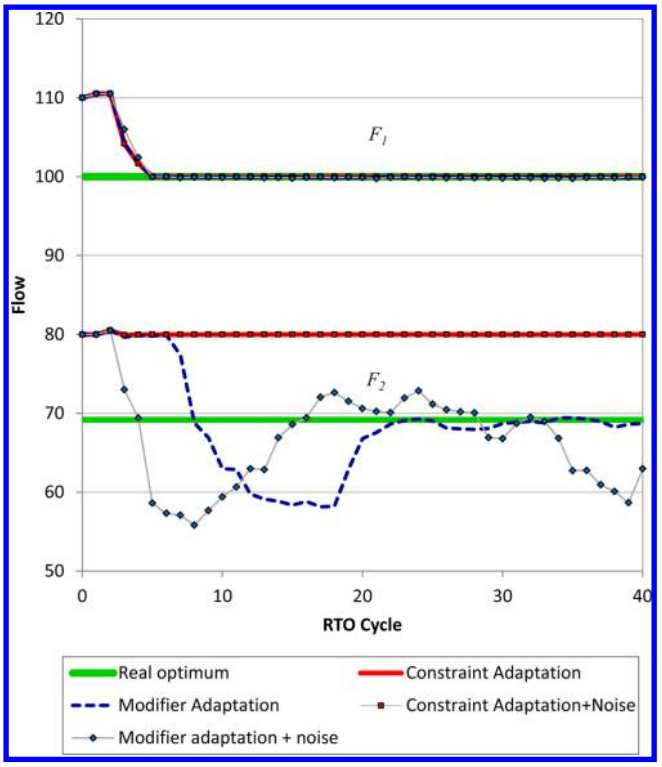

Figure 6. Evolution of the inputs vs RTO cycle.

number. In these two figures, it can be observed that the constraint adaptation strategy shows less variability in the results and converges to a point that improves the objective function significantly, with respect to the initial point. However, since it does not correct gradients, it fails to converge to a KKT point of the real plant. Instead, the full modifier adaptation strategy converges to the vicinity of the real plant optimum. However, at the same time, it shows more variability in the results, because of the errors in gradient estimation.

Interestingly, the evolutions with constraint adaptation are the same for noise-free and Gaussian noise cases. In fact, the optimization results always select the same point, ($F_1 = 100$, $F_2 = 80$), which is not achieved in one iteration, because of the input filter. In addition, the original model without adaptation (not shown in the figures) also finds the same optimum, showing that, in some cases, a model without correction can identify the same active constraints as an adapted model in RTO.

Noise-free modifier adaptation starts following a path that is similar to that of constraint adaptation. After 6 RTO cycles, it starts evolving toward the real plant optimum, and after 20 RTO cycles, the values of F_1 and F_2 converge to the vicinity of the real optimum. The scenario with Gaussian noise starts evolving faster to the optimum. However, for this scenario, the process input F_2 shows a high variability along the entire period under study.

Figure 7 shows the evolution of the objective function for the 40 RTO cycles. It can be observed that the constraint adaptation strategy converges faster, but the final objective function value is not the optimal value. The modifier adaptation strategy requires more cycles to converge, but the achieved objective function value is the optimal. Once convergence is achieved, the objective function value is almost constant for the noise-free case. Despite the variability in the inputs shown by Figure 6, after RTO cycle $k = 5$, the objective function value is always maintained close to its real optimal value.

The Extended Design Cost (EDC) criterion is used to compare the performance of different RTO strategies. It estimates the

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

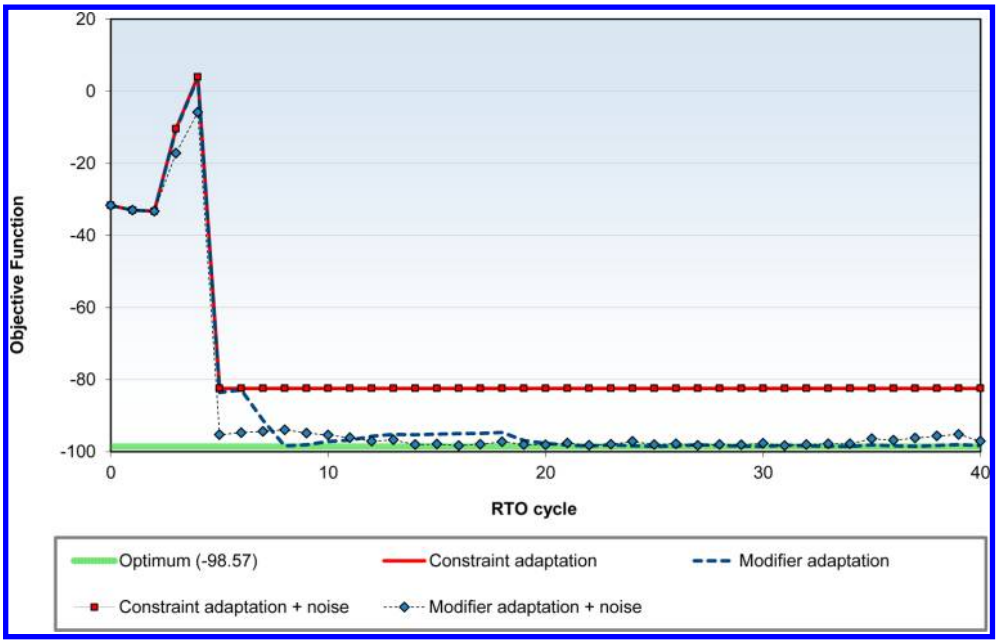

Figure 7. Evolution of objective function for constraint adaptation and modifier adaptation strategies. Initial point: (F_1^0 = 110, F_2^0 = 80).

total loss of profit of the RTO approach, compared to the real plant optimum:[44]

$$\begin{aligned} \mathrm{EDC} &= \int_{T_0}^{T_f} (Q_r^{\,k} - Q_r^{\,*})\,\mathrm{d}t \\ &\approx \frac{1}{2}(Q_r^{\,k_0} - Q_r^{\,*}) + \sum_{k=k_0+1}^{k_f-1} (Q_r^{\,k} - Q_r^{\,*}) \\ &\quad + \frac{1}{2}(Q_r^{\,k_f} - Q_r^{\,*}) \end{aligned} \tag{54}$$

where Q_r^* is the actual optimal cost of the plant and Q_r^k is the actual cost at cycle k. T_0 and T_f are the initial and final times corresponding to RTO cycles k_0 and k_f, respectively.

Table 2 shows the EDC for all of the analyzed scenarios and strategies, including the EDC when no optimization action is applied (i.e., the system operates all the time with the initial input set). The EDC is evaluated for all the cycles (i.e., from k = 0 to 40) and for $k \geq 20$, which is the time when all cycles have reached an objective function value close to their convergence value (as it can be observed in Figure 7). The EDC for $k \geq 20$ evaluates the loss of profit due to variability and to plant−model mismatch, while the first EDC from k = 0 to k = 40 also includes the cost of the evolution to the convergence point. It can be observed that the performance of modifier adaptation is much better than constraint adaptation, because of the structural mismatch correction of the gradient modifiers. Both approaches achieve a significant reduction in the objective function value, and, therefore, these alternatives proved to be potentially useful in a RTO implementation that requires the use of disjunctions or discrete variables.

Table 2. Case Study: Extended Design Costs

	EDC from k = 0 to k = 40		EDC from k = 20 to k = 40	
scenario	absolute	relative	absolute	relative
no action	2676.8	100.0	1338.4	100.0
Noise-Free Scenarios				
constraint adaptation	927.7	34.7	338.7	25.3
modifier adaptation	427.6	16.0	6.3	0.5
Gaussian Noise in the Data				
constraint adaptation	927.7	34.7	338.7	25.3
modifier adaptation	394.3	14.7	23.4	1.7

As mentioned previously, the EDC costs for noise-free and Gaussian noise scenarios are the same with constraint adaptation, since the evolutions are identical.

The EDC from k = 0 to k = 40 for the Gaussian noise scenario, using modifier adaptation, is slightly lower than the noise-free scenario. This is mainly due to the faster initial convergence. Nevertheless, as expected, the EDC for $k \geq 20$ is three times higher for the Gaussian noise case, because of the variability in F_2 that is observed in the figures.

7. CONCLUSIONS

A real-time optimization (RTO) formulation including discrete decisions has been presented and discussed. It has been shown that discrete events of different nature can appear in industrial processes, even if the startup or shutdown of pieces of equipment are not considered.

The two-step adaptation approach and two versions of the modifier adaptation approach were reformulated to make them suitable for problems involving discrete decisions. The reformulation includes the selection of parameters/modifiers to update at each RTO cycle, the exponential filtering, and the past cycles that are used for gradient correction.

New bounds were defined for input filtering. They help to solve two problems that can appear if disjunctions are present: slow and expensive (in terms of the objective function) convergence to the plant optimum, and infeasibility of the selected inputs if disjoint regions are present. These bounds may be in conflict with other bounds formulated in recent publications for nonlinear, continuous RTO. Therefore, this problem should be investigated further.

Maximum change constraints and convexifying terms may cause a convergence to suboptimal points if the problem includes discrete decisions. A strategy to avoid this undesired behavior was presented and illustrated with examples.

dx.doi.org/10.1021/ie5004619 | Ind. Eng. Chem. Res. 2014, 53, 17200−17213

The case study showed that a practical implementation of RTO with discrete decisions is certainly possible. The two analyzed strategies (constraint adaptation and modifier adaptation) showed a behavior similar to that of nonlinear, continuous RTO: convergence to the real plant optimum for gradient adaptation schemes, and less variability in results for constraint adaptation. Both approaches proved to be suitable for an application of RTO with disjunctions.

The CPU times required to solve an RTO cycle (i.e., to adapt the model and to solve the optimization problem) were satisfactory for a RTO implementation. However, the effect of increasing the number of binary variables and the model size must be investigated further.

If the discrete decisions addressed by RTO involve startups or shutdowns, or constraints involving time, a multiperiod RTO formulation may be necessary. The integration of RTO with scheduling optimization, using a multiperiod formulation, will be addressed in a future work.

■ ASSOCIATED CONTENT

Supporting Information

A description of modifier adaptation using performance equations for problems including disjunctions is available in this section. The document also includes a strategy for selecting the input filtering constant for several disjunctions with fixed costs and disjoint feasible regions. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*Tel.: +54-342-4535568. E-mail: mmussati@santafe-conicet.gov.ar.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support from CONICET, Universidad Nacional del Litoral and Soteica Latinoamérica S.A.

■ REFERENCES

(1) Mansour, M.; Ellis, J. E. Methodology of On-Line Optimisation Applied to a Chemical Reactor. *Appl. Math. Modelling* **2008**, *32*, 170−184.

(2) Darby, M. L.; Nikolaou, M.; Jones, J.; Nicholson, D. RTO: An Overview and Assessment of Current Practice. *J. Process Control* **2011**, *21*, 874−884.

(3) Cao, S.; Rhinehart, R. R. An Efficient Method for on-Line Identification of Steady State. *J. Process Control* **1995**, *5*, 363−374.

(4) Roux, G. A. C. L.; Santoro, B. F.; Sotelo, F. F.; Teissier, M.; Joulia, X. Improving Steady-State Identification. *Comput.-Aided Chem. Eng.* **2008**, *25*, 459−464.

(5) Narasimhan, S.; Jordache, C. *Data Reconciliation & Gross Error Detection: An Intelligent Use of Process Data.*; Gulf Professional Publishing: Houston, TX, 1999.

(6) Romagnoli, J.; Sanchez, M. C. *Data Processing and Reconciliation for Chemical Process Operations*; Process Systems Engineering Series; Academic Press: New York, 1999; Vol. 2.

(7) Chachuat, B.; Srinivasan, B.; Bonvin, D. Adaptation Strategies for Real-Time Optimization. *Comput. Chem. Eng.* **2009**, *33*, 1557−1567.

(8) Chen, C. Y.; Joseph, B. On-Line Optimization Using a Two-Phase Approach: An Application Study. *Ind. Eng. Chem. Res.* **1987**, *26*, 1924−1930.

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

(10) Marchetti, A. G. A New Dual Modifier-Adaptation Approach for Iterative Process Optimization with Inaccurate Models. *Comput. Chem. Eng.* **2013**, *59*, 89−100.

(11) Marchetti, A.; Chachuat, B.; Bonvin, D. A Dual Modifier-Adaptation Approach for Real-Time Optimization. *J. Process Control* **2010**, *20*, 1027−1037.

(12) Forbes, J. F.; Marlin, T. E. Model Accuracy for Economic Optimizing Controllers: The Bias Update Case. *Ind. Eng. Chem. Res.* **1994**, *33*, 1919−1929.

(13) Chachuat, B.; Marchetti, A.; Bonvin, D. Process Optimization via Constraints Adaptation. *J. Process Control* **2008**, *18*, 244−257.

(14) Bunin, G. A.; Wuillemin, Z.; François, G.; Nakajo, A.; Tsikonis, L.; Bonvin, D. Experimental Real-Time Optimization of a Solid Oxide Fuel Cell Stack via Constraint Adaptation. *Energy* **2012**, *39*, 54−62.

(15) François, G.; Bonvin, D. Use of Convex Model Approximations for Real-Time Optimization via Modifier Adaptation. *Ind. Eng. Chem. Res.* **2013**, *52*, 11614−11625.

(16) Roberts, P. D. Coping with Model-Reality Differences in Industrial Process Optimisation—A Review of Integrated System Optimisation and Parameter Estimation (ISOPE). *Comput. Ind.* **1995**, *26*, 281−290.

(17) Tatjewski, P. *Iterative Optimizing Set-Point Control—The Basic Principle Redesigned*. In *Proceedings of the 15th International Federation of Automatic Control (IFAC) World Congress*, Barcelona, Spain, July 21−26, 2002; Vol. *15*, p 992.

(18) Srinivasan, B.; Biegler, L. T.; Bonvin, D. Tracking the Necessary Conditions of Optimality with Changing Set of Active Constraints Using a Barrier-Penalty Function. *Comput. Chem. Eng.* **2008**, *32*, 572−579.

(19) Adetola, V.; Guay, M. Parameter Convergence in Adaptive Extremum-Seeking Control. *Automatica* **2007**, *43*, 105−110.

(20) Mansour, M.; Ellis, J. E. Comparison of Methods for Estimating Real Process Derivatives in On-Line Optimization. *Appl. Math. Modelling* **2003**, *27*, 275−291.

(21) Srinivasan, B.; François, G.; Bonvin, D. Comparison of Gradient Estimation Methods for Real-Time Optimization. *Comput.-Aided Chem. Eng.* **2011**, *29*, 607−611.

(22) Bunin, G. A.; François, G. Exploiting Local Quasiconvexity for Gradient Estimation in Modifier-Adaptation Schemes. In *Proceedings of the 2012 American Control Conference*, pp 2806−2811.

(23) Bunin, G. A.; François, G.; Bonvin, D. From Discrete Measurements to Bounded Gradient Estimates: A Look at Some Regularizing Structures. *Ind. Eng. Chem. Res.* **2013**, *52*, 12500−12513.

(24) Serralunga, F. J.; Mussati, M. C.; Aguirre, P. A. Model Adaptation for Real-Time Optimization in Energy Systems. *Ind. Eng. Chem. Res.* **2013**, *52*, 16795−16810.

(25) François, G.; Bonvin, D. Use of Transient Measurements for the Optimization of Steady-State Performance via Modifier Adaptation. *Ind. Eng. Chem. Res.* **2014**, *53*, 5148−5159.

(26) Zhang, Y.; Nadler, D.; Forbes, J. F. Results Analysis for Trust Constrained Real-Time Optimization. *J. Process Control* **2001**, *11*, 329−341.

(27) Brdys, M. A.; Tatjewski, P. *Iterative Algorithms for Multilayer Optimization Control*; 1st ed.; Imperial College Press: London, U.K., 2005.

(28) Bunin, G.; François, G.; Srinivasan, B.; Bonvin, D. Input Filter Design for Feasibility in Constraint-Adaptation Schemes. In *Proceedings of the 18th IFAC World Congress*, Milano, Italy, Aug. 28−Sept. 2, 2011; pp 5585−5590.

(29) Mariani, D. C.; Kihn, M. A.; Ruiz, C. A. Industrial Experience on the Implementation of Real Time On Line Energy Management Systems in Sugar and Alcohol Industry. *Comput.-Aided Chem. Eng.* **2009**, *27*, 459−464.

(30) Ruiz, C. A. Real Time Industrial Process Systems: Experiences from the Field. *Comput.-Aided Chem. Eng.* **2009**, *27*, 133−138.

(31) Ruiz, D.; Ruiz, C.; Serralunga, F. Real-Time Refinery Energy Management. *Pet. Technol. Q.* **2009**, *14*, 115−118.
(32) Puranik, Y.; Sahinidis, N.; Li, T.; Feather, D.; Besancon, B. Real Time Optimization of a Complex Industrial Gas Network. Presented at *AIChE Annual Meeting*, San Francisco, CA, 2013.
(33) Serralunga, F. J.; Mussati, M. C.; Aguirre, P. A. Optimización en tiempo real con disyunciones lógicas. Aplicación a sistemas de calor y potencia (Real-time optimization with logical disjunctions. Application to heat and power systems). In *42 Jornadas Argentinas de Informática - 2° Simposio Argentino de Informática Industrial*, Córdoba, Argentina, 2013; pp 237−248 (ISSN No. 2313-9102).
(34) Raman, R.; Grossmann, I. E. Modelling and Computational Techniques for Logic Based Integer Programming. *Comput. Chem. Eng.* **1994**, *18*, 563−578.
(35) Grossmann, I.; Ruiz, J. Generalized Disjunctive Programming: A Framework for Formulation and Alternative Algorithms for MINLP Optimization. In *Mixed Integer Nonlinear Programming*; Lee, J., Leyffer, S., Eds.; The IMA Volumes in Mathematics and Its Applications; Springer: New York, 2012; Vol. *154*, pp 93−115.
(36) Lee, S.; Grossmann, I. E. New Algorithms for Nonlinear Generalized Disjunctive Programming. *Comput. Chem. Eng.* **2000**, *24*, 2125−2141.
(37) Vecchietti, A.; Grossmann, I. E. LOGMIP: A Disjunctive 0−1 Non-Linear Optimizer for Process System Models. *Comput. Chem. Eng.* **1999**, *23*, 555−565.
(38) Türkay, M.; Grossmann, I. E. Structural Flowsheet Optimization with Complex Investment Cost Functions. *Comput. Chem. Eng.* **1998**, *22*, 673−686.
(39) Mitra, S.; Sun, L.; Grossmann, I. E. Optimal Scheduling of Industrial Combined Heat and Power Plants under Time-Sensitive Electricity Prices. *Energy* **2013**, *54*, 194−211.
(40) Rodger, E.; Chachuat, B. Design Methodology of Modifier Adaptation for On-Line Optimization of Uncertain Processes. In *Proceedings of 18th IFAC World Congress*, Milano, Italy, 2011.
(41) Yip, W. S.; Marlin, T. E. The Effect of Model Fidelity on Real-Time Optimization Performance. *Comput. Chem. Eng.* **2004**, *28*, 267−280.
(42) GAMS Development Corporation. *GAMS Language Guide, RELEASE 2.25, Version 92*; Washington, DC, 1997.
(43) GAMS Development Corporation. *GAMS—The Solver Manuals*; Washington, DC, 2008.
(44) Zhang, Y.; Forbes, J. F. Extended Design Cost: A Performance Criterion for Real-Time Optimization Systems. *Comput. Chem. Eng.* **2000**, *24*, 1829−1841.