

Real-Time Optimization via Adaptation and Control of the Constraints

Alejandro Marchetti, Benoît Chachuat, Dominique Bonvin

*Laboratoire d'Automatique,
Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland*

Abstract

In the framework of real-time optimization, measurements are used to compensate for effects of uncertainty. The main approach uses measurements to update the parameters of a process model. In contrast, the constraint-adaptation scheme uses the measurements to bias the constraints in the optimization problem. In this paper, an algorithm combining constraint adaptation with a constraint controller is presented. The former detects shifts in the set of active constraints and passes the set points of the active constraints to the latter. In order to avoid constraint violation, the set points are moved gradually during the iterative process. Moreover, the constraint controller manipulates linear combinations of the original input variables. The approach is illustrated for a simple case study.

Keywords: Real-time optimization, constraint control, constraint adaptation.

1. Introduction

Throughout the petroleum and chemicals industry, the control and optimization of many large-scale systems is organized in a hierarchical structure. At the real-time optimization level (RTO), decisions are made on a time scale of hours to a few days by a so-called real-time optimizer that determines the optimal operating point under changing conditions. The RTO is typically a nonlinear program (NLP) minimizing cost or maximizing economic productivity subject to constraints derived from steady-state mass and energy balances and physical relationships. At a lower level, the process control system implements the RTO results, including product qualities, production rates and active constraints (Marlin and Hrymak, 1997).

Because accurate mathematical models are unavailable for most industrial applications, RTO classically proceeds by a two-step approach, namely an identification step followed by an optimization step. Variants of this two-step approach such as ISOPE (Roberts and Williams, 1981; Brdys and Tatjewski, 2005) have also been proposed for improving the synergy between the identification and optimization steps.

Parameter identification is complicated by several factors: (i) the complexity of the models and the nonconvexity of the parameter estimation problems, and (ii) the need for the model parameters to be identifiable from the available measurements. Moreover, in the presence of structural plant-model mismatch, parameter identification does not necessarily lead to model improvement. In order to avoid the task of identifying a model on-line, fixed-model methods have been proposed. The idea therein is to utilize both the available measurements and a (possibly inaccurate) steady-state model to drive the process towards a desirable operating point. In constraint-adaptation schemes (Forbes and Marlin, 1994; Chachuat et al., 2007), for instance, the measurements are used to correct the constraint functions in the RTO problem, whereas a process model is used to

estimate the gradients of the cost and constraint functions. This way, the iterates are guaranteed to reach a feasible, yet suboptimal, operating point upon convergence.

Two types of transients can be distinguished in RTO systems: at the lower level, the dynamic response of the controlled plant between successive steady-state operating points generated by RTO; at the upper level, the transient produced by the iterates of the RTO algorithm. Most RTO algorithms do not ensure feasibility during these transient periods, thus resulting in conservative implementations with significant constraint backoffs and limited changes in operating point between successive RTO periods. Constraint violations during both types of transients can be avoided by controlling the active constraints that define optimal operation (Brdys and Tatjewski, 2005). The implementation of constraint control can significantly decrease the constraint backoffs required in the RTO optimization problem, resulting in increased cost. The set of active constraints might change due to process disturbances and changing operating conditions, thus resulting in different constraint-control schemes (Maarleveld and Rijnsdorp, 1970; Garcia and Morari, 1984).

In this work, a constraint-adaptation scheme is combined with a constraint controller. Special emphasis is placed on selecting the set points and the manipulated variables used in the constraint controller at each RTO period. The effect of the constraint controller on the feasibility of intermediate operating points is studied, under the assumption of an ideal constraint controller.

The paper is organized as follows. Section 2 formulates the optimization problem. The RTO scheme combining constraint adaptation and constraint control is presented in Section 3. The behavior of the proposed scheme, with and without the constraint controller, is illustrated for a simple quadratic programming (QP) problem in Section 4. Finally, Section 5 concludes the paper.

2. Problem Formulation

The optimization problem for the plant can be formulated as follows:

$$\min_{\mathbf{u}} \bar{\Phi}(\mathbf{u}) := \phi(\mathbf{u}, \bar{\mathbf{y}}(\mathbf{u})) \quad (1)$$

$$\text{s.t. } \bar{\mathbf{G}}(\mathbf{u}) := \mathbf{g}(\mathbf{u}, \bar{\mathbf{y}}(\mathbf{u})) \leq \mathbf{G}_{\max}, \quad (2)$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the vector of decision (or input) variables, and $\bar{\mathbf{y}} \in \mathbb{R}^{n_y}$ is the vector of controlled (or output) variables; $\phi: \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is the scalar cost function to be minimized; and $\mathbf{g}_i: \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$, $i=1, \dots, n_g$, is the set of operating constraints. Throughout the paper, the notation $(-)$ is used for the variables that are associated with the plant and (\cdot) for those of the process model.

The steady-state mapping of the plant, $\bar{\mathbf{y}}(\mathbf{u})$, is assumed to be unknown, and only an approximate model $\mathbf{F}(\mathbf{u}, \mathbf{y}, \theta) = \mathbf{0}$ is available for it, where $\theta \in \mathbb{R}^{n_\theta}$ is the set of model parameters. Assuming that the model outputs \mathbf{y} can be expressed explicitly as functions of \mathbf{u} and θ , the cost function and the operating constraints predicted by the model can be written as $\Phi(\mathbf{u}, \theta) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u}, \theta))$ and $\mathbf{G}(\mathbf{u}, \theta) := \mathbf{g}(\mathbf{u}, \mathbf{y}(\mathbf{u}, \theta))$, respectively.

3. Real-Time Optimization Scheme

3.1. Constraint Adaptation

In the presence of uncertainty, the constraint values predicted by the model do not quite match those of the plant. The idea behind constraint adaptation is to modify the optimization problem by adding a correction term to the constraint functions. At each RTO iteration, a model-based optimization problem of the following form is solved:

$$\begin{aligned} \min_{\mathbf{u}_k} \quad & \Phi(\mathbf{u}_k, \theta) \\ \text{s.t.} \quad & \mathbf{G}(\mathbf{u}_k, \theta) + \boldsymbol{\varepsilon}_k \leq \mathbf{G}_{\max}, \end{aligned} \quad (3)$$

where $\boldsymbol{\varepsilon}_k \in \mathbb{R}^{n_g}$ denotes the vector of constraint correction factors. Under the assumption that measurements are available for every constrained quantity at the end of each RTO period, the correction factors can be updated recursively as:

$$\boldsymbol{\varepsilon}_{k+1} = (\mathbf{I} - \mathbf{B})\boldsymbol{\varepsilon}_k + \mathbf{B}[\bar{\mathbf{G}}(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k, \theta)], \quad (4)$$

where $\mathbf{B} \in \mathbb{R}^{n_g \times n_g}$ is a diagonal gain matrix with diagonal elements in the interval $(0, 1]$. An important property of the constraint-adaptation algorithm is that the iterates are guaranteed to reach a feasible, yet suboptimal, operating point upon convergence (Forbes and Marlin, 1994). However, the constraints can be violated during the iterations, which calls for using constraint backoffs and limiting operating point changes between successive RTO periods.

Constraint adaptation (3-4) represents the ‘‘classical’’ constraint-adaptation scheme (Forbes and Marlin, 1994; Chachuat et al., 2007). In this paper, a novel way of adapting the constraints is proposed:

$$\min_{\mathbf{u}_k} \quad \Phi(\mathbf{u}_k, \theta) \quad (5)$$

$$\text{s.t.} \quad \mathbf{G}(\mathbf{u}_k, \theta) + \boldsymbol{\gamma}_k \leq \mathbf{G}_{\max, k}, \quad (6)$$

where the correction term $\boldsymbol{\gamma}_k := \bar{\mathbf{G}}(\mathbf{u}_{k-1}) - \mathbf{G}(\mathbf{u}_{k-1}, \theta)$ stands for the difference between the measured and predicted values at the previous RTO period. The maximum values $\mathbf{G}_{\max, k}$ for the constraints are calculated as:

$$\mathbf{G}_{\max, k} = \bar{\mathbf{G}}(\mathbf{u}_{k-1}) + \mathbf{B}[\mathbf{G}_{\max} - \bar{\mathbf{G}}(\mathbf{u}_{k-1})]. \quad (7)$$

For the combination with constraint control, constraint adaptation (6-7) is preferred because it gives the ability to vary the set points $\mathbf{G}_{\max, k}$ passed to the constraint controller. Upon convergence of this algorithm, the set points reach the original constraint bounds \mathbf{G}_{\max} . Let \mathbf{u}_0^* denote the optimal solution for the process model with $\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}_0$ in (3). It can be shown that the constraint-adaptation schemes (3-4) and (6-7) produce the same iterates when initialized with $\boldsymbol{\varepsilon}_0$ and \mathbf{u}_0^* , respectively, and the same diagonal gain matrix \mathbf{B} is used, provided the set of active constraints does not change.

At each RTO period, a set of optimal inputs, \mathbf{u}_k^* , and corresponding Lagrange multipliers, $\boldsymbol{\lambda}_k^*$, are obtained from the numerical solution of (5-6). Let $\mathbf{G}_k^a \in \mathbb{R}^{n_k^a}$ denote the vector of active constraints at \mathbf{u}_k^* . It is assumed that the Jacobian matrix of the active constraints, $\mathbf{G}_{\mathbf{u}, k}^a \in \mathbb{R}^{n_k^a \times n_u}$, has full row rank at \mathbf{u}_k^* , i.e. the constraints satisfy a regularity condition. It follows that the input space can be split into the n_k^a -dimensional subspace of *constraint-seeking directions* and the $(n_u - n_k^a)$ -dimensional subspace of *sensitivity-seeking directions*. These subspaces are spanned by the columns of the orthogonal matrices \mathbf{V}_k^c and \mathbf{V}_k^s , respectively, as obtained from singular-value decomposition (SVD) of $\mathbf{G}_{\mathbf{u}, k}^a$:

$$\mathbf{G}_{\mathbf{u}, k}^a = [\mathbf{U}_k^c \quad \mathbf{U}_k^s][\boldsymbol{\Sigma}_k^c \quad \mathbf{0}][\mathbf{V}_k^c \quad \mathbf{V}_k^s]^T. \quad (8)$$

3.2. Combination with Constraint Control

At the constraint-control level, the variables are considered as time-dependent signals. In this work, the constraint controller is designed so as to track the iteratively-updated active constraints by varying the process inputs along the constraint-seeking directions.

More precisely, the manipulated variables (MVs) in the constraint controller correspond to the corrections $\delta \mathbf{u}_k^c(t) \in \mathbb{R}^{n_k^c}$ along the directions \mathbf{V}_k^c , from the model optimum \mathbf{u}_k^* . Observe that the MVs may change from RTO period to RTO period, e.g. when the active set of (5-6) changes. At each time instant, the inputs $\mathbf{u}_k(t)$ are then reconstructed from the values of $\delta \mathbf{u}_k^c(t)$, based on the knowledge of \mathbf{u}_k^* and \mathbf{V}_k^c , as:

$$\mathbf{u}_k(t) = \mathcal{U}(\mathbf{u}_k^*, \mathbf{V}_k^c, \delta \mathbf{u}_k^c(t)) := \mathbf{u}_k^* + \mathbf{V}_k^c \delta \mathbf{u}_k^c(t). \quad (9)$$

The set points in the constraint controller correspond to the active constraints, $\mathbf{G}_{\max,k}^a \in \mathbb{R}^{n_k^a}$, determined at the RTO level. Finally, the controlled variables (CVs) are the active constraints $\bar{\mathbf{G}}_k^a(t) := \mathbf{g}^a(\mathbf{u}_k(t), \bar{\mathbf{y}}_k(t))$ for the plant.

At the initial time t_{k-1} of the k -th RTO period, the constraint controller is started from $\delta \mathbf{u}_k^c(t_{k-1}) = \mathbf{V}_k^{cT}(\mathbf{u}_{k-1} - \mathbf{u}_k^*)$. At the terminal time t_k of that period, the constraint controller yields a new steady-state operation, which corresponds to the set points $\mathbf{G}_{\max,k}^a$. The corresponding steady-state inputs \mathbf{u}_k are obtained from (9) as $\mathbf{u}_k = \mathcal{U}(\mathbf{u}_k^*, \mathbf{V}_k^c, \delta \mathbf{u}_k^c(t_k))$.

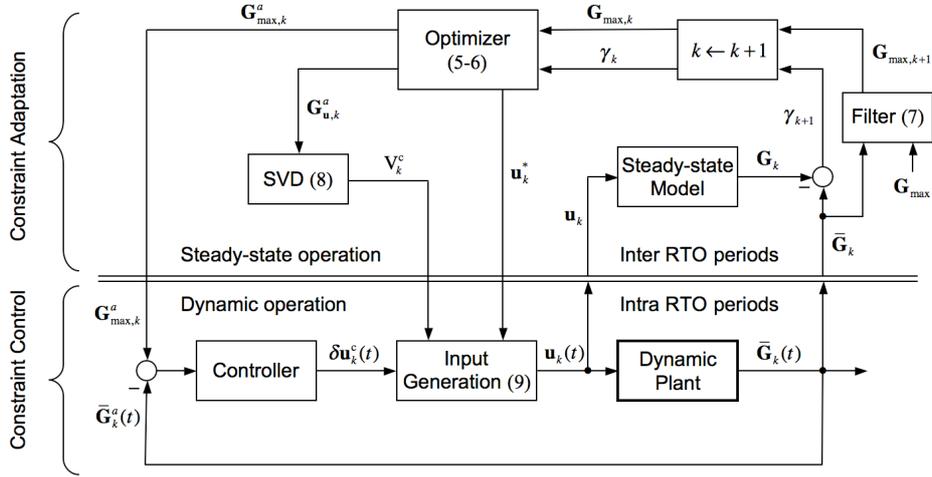


Figure 1. Scheme combining constraint adaptation and constraint control.

The overall optimization and control scheme is illustrated in Fig. 1, and the procedure can be summarized as follows:

1. Set $k = 0$. Initialize B. Start from a feasible (conservative) operating point \mathbf{u}_0 (without the constraint controller).
2. At steady state, measure $\bar{\mathbf{G}}(\mathbf{u}_k)$ and compute $\mathbf{G}_{\max,k+1}^a$ from (7). Set $k := k + 1$.
3. Calculate the solution \mathbf{u}_k^* of (5-6).
4. Determine the constraint-seeking directions \mathbf{V}_k^c from SVD of the Jacobian matrix $\mathbf{G}_{\mathbf{u},k}^a$ of active constraints at \mathbf{u}_k^* .
5. Formulate a square constraint-control problem where the MVs are the values of $\delta \mathbf{u}_k^c(t)$, the CVs are the active constraints $\bar{\mathbf{G}}_k^a(t)$, and the set points are the values $\mathbf{G}_{\max,k}^a$ of the active constraints identified in Step 3.
6. Apply the constraint controller to the plant and get the inputs \mathbf{u}_k corresponding to the new steady-state operation. Go to Step 2.

3.3. Implementation Aspects

The approach assumes that all the constrained variables can be measured or estimated on-line at a sampling period much smaller than the time constant of the controlled plant. Notice that the decision variables \mathbf{u} in the RTO problem may very well be set points of feedback controllers acting directly on the plant manipulated variables. In this case, the constraint controller can be viewed as a primary controller in a cascade control configuration that corrects the set points produced at the RTO level.

The constraint-control problem is a multivariable square control problem, and various controllers can be used, such as a discrete integral controller or a model predictive controller.

In order to avoid overshoots, the set-point corrections can be implemented by ramps rather than steps. Also, small overshoots can usually be accommodated during the first few iterations, i.e. when the set points $\mathbf{G}_{\max,k}^a$ are conservative with respect to the actual bounds \mathbf{G}_{\max} .

4. Illustrative Example

Consider the following QP problem:

$$\min \quad \Phi(\mathbf{u}, \boldsymbol{\theta}) := (u_1 - 1)^2 + (u_2 - 1)^2 \quad (10)$$

$$\text{s.t.} \quad G_1 := \theta_1 + \theta_2 u_1 - u_2 \leq 0, \quad G_2 := \theta_3 + \theta_4 u_1 + u_2 \leq 0, \quad (11)$$

with two decision variables $\mathbf{u} = [u_1 \ u_2]^T$, four model parameters $\boldsymbol{\theta} = [\theta_1, \dots, \theta_4]^T$, and two uncertain constraints G_1 and G_2 . The parameter values for the model and the simulated reality are reported in Table 1. Note that the operating point determined from the model, without constraint adaptation, leads to constraint violation.

Table 1. Values of the parameters $\boldsymbol{\theta}$ for the model and the simulated reality.

	θ_1	θ_2	θ_3	θ_4
Reality	0.4	0.8	-1.8	1.9
Model	0.9	0.4	-2.0	1.4

In this simple QP problem, an ideal constraint controller is assumed, i.e. the controller determines $\delta \mathbf{u}_k^c(t_k)$ such that $\bar{\mathbf{G}}^a \left[\mathcal{U}(\mathbf{u}_k^*, \mathbf{V}_k^c, \delta \mathbf{u}_k^c(t_k)) \right] = \mathbf{G}_{\max,k}^a$. The objective here is to illustrate the effect of constraint control on the feasibility of the steady-state intermediates.

Both constraints are active at the optimum either for the reality or for the model. The constraint-adaptation algorithm is applied with and without constraint control, starting from $\mathbf{u}_0 = [0 \ 1.4]^T$ and with a diagonal gain matrix $\mathbf{B} = b \mathbf{I}_{2 \times 2}$ with $b \in (0, 1]$. The results obtained with $b = 0.7$ are shown in Fig. 2. It can be seen that, without constraint control, the iterates converge by following an infeasible path (left plot). In fact, the iterates can be shown to follow an infeasible path for any value of $b \in (0, 1]$; the constraint violation is reduced by decreasing the value of b , but this is at the expense of a slower convergence. With constraint control, on the other hand, the iterates converge without violating the constraints (right plot), irrespectively of the value of b . Both constraints are found to be active at the solution point of the optimization problem (5-6) for all iterations. Since the number of active constraints is equal to the number of decision variables, the constraint-seeking directions span the whole input space here.

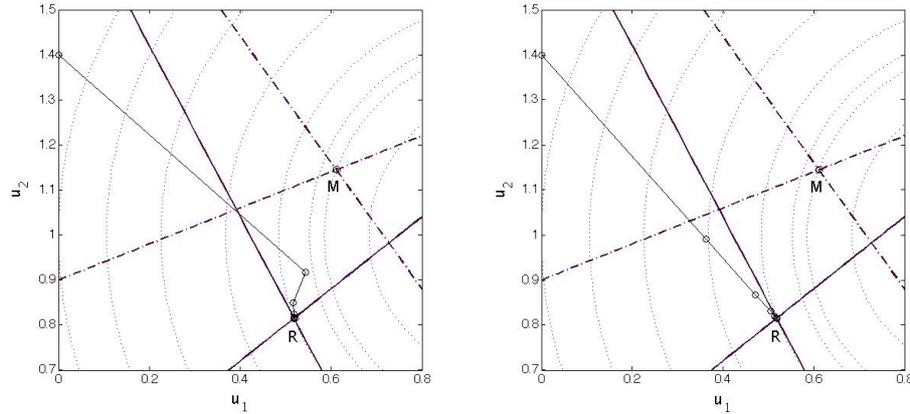


Figure 2. Illustration of the proposed algorithm for Problem (10-11). **Left plot:** Without constraint control; **Right plot:** With constraint control; **Thick solid lines:** constraint bounds for the simulated reality; **Thick dash-dotted lines:** constraint bounds predicted by the model without adaptation; **Dotted lines:** contours of the cost function; **Light solid lines:** iterates; **Point R:** optimum for the simulated reality; **Point M:** optimum for the model without adaptation.

5. Conclusions

An optimization scheme combining constraint adaptation with constraint control has been proposed. This scheme presents two important features: (i) the constraint controller tracks the active constraint determined at the RTO level by adapting the inputs in the subspace of constraint-seeking directions, and (ii) the set points for the active constraints in the constraint controller are updated at each iteration and reach the actual constraint bounds upon convergence.

In future work, this combined scheme will be compared to other existing approaches (e.g. Ying and Joseph, 1999). The combination of more involved RTO schemes with constraint control (e.g. Gao and Engell, 2005) will also be considered.

References

- M. A. Brdys and P. Tatjewski, 2005, *Iterative algorithms for multilayer optimizing control*. Imperial College Press, London, UK.
- B. Chachuat, A. Marchetti, and D. Bonvin, 2007, Process optimization via constraint adaptation, *J. Process Contr.*, In press.
- J. F. Forbes and T. E. Marlin, 1994, Model accuracy for economic optimizing controllers: The bias update case, *Ind. Eng. Chem. Res.*, **33**, 1919-1929.
- W. Gao and S. Engell, 2005, Iterative set-point optimization of batch chromatography, *Comp. Chem. Eng.*, **29**, 1401-1409.
- C. E. Garcia and M. Morari, 1984, Optimal operation of integrated processing systems. Part II: Closed-loop on-line optimizing control, *AIChE J.*, **30**, 2, 226-234.
- A. Maarleveld and J. E. Rijnsdorp, 1970, Constraint control on distillation columns, *Automatica*, **6**, 51-58.
- T. E. Marlin and A. N. Hrymak, 1997, Real-time operations optimization of continuous processes, *AIChE Symp. Ser.*, **93**, 156-164.
- P. D. Roberts and T. W. C. Williams, 1981, On an algorithm for combined system optimization and parameter estimation, *Automatica*, **17**, 1, 199-209.
- C.-M. Ying and B. Joseph, 1999, Performance and stability analysis of LP-MPC and QP-MPC Cascade Control Systems, *AIChE J.*, **45**, 7, 1521-1534.