Input Filter Design for Feasibility in Constraint-Adaptation Schemes

Gene A. Bunin, * **Grégory François,** * **Bala Srinivasan,** ** **Dominique Bonvin** *

* *Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland (e-mail: dominique.bonvin@epfl.ch).*
** *Department of Chemical Engineering, Ecole Polytechnique de Montreal, Montreal, Canada, H3C 3A7.*

Abstract: The subject of real-time, steady-state optimization under significant uncertainty is addressed in this paper. Specifically, the use of constraint-adaptation schemes is reviewed, and it is shown that, in general, such schemes cannot guarantee process feasibility over the relevant input space during the iterative process. This issue is addressed via the design of a feasibility-guaranteeing input filter, which is easily derived through the use of a Lipschitz bound on the plant behavior. While the proposed approach works to guarantee feasibility for the single-constraint case, early sub-optimal convergence is noted for cases with multiple constraints. In this latter scenario, some constraint violations must be accepted if convergence to the optimum is desired. An illustrative example is given to demonstrate these points.

Keywords: Real-time optimization, optimization under uncertainty, modifier adaptation, filter design, robust optimization.

1. INTRODUCTION

For the past two to three decades, optimization has played an increasingly significant role in the operation of plant-wide processes, and will likely continue to do so for the foreseeable future. Analogous in many ways to its more-developed neighbor field of process control, the role of optimization in the process industry appears to have gone through a similar evolution - with simpler algorithms giving way to more challenging and demanding ones as processes are pushed to do more. As a result, what might have originally been a design step to determine a theoretically optimal set of operating regimes has now become a closed-loop algorithm of its own. This has created a sort of "optimization with feedback" subfield, and has paved the way for the implementation of more complex real-time optimization (RTO) structures (Engell (2007); Tatjewski (2008)). This is important, as model uncertainty is a standard issue in process operation, where optimizing with a faulty model will not only lead to the "wrong" optimum, but possibly to one that is infeasible as well, either from a physical or safety viewpoint.

In view of these difficulties, numerous authors have investigated robustness and feasibility in RTO schemes. Many have looked at the problem as one would look at robust control, and have attempted to guarantee constraint satisfaction with a high probability either through stochastic chance-constraint programming (CCP) (see, e.g., Zhang et al. (2002); Zhang (2010)) or through methods where all the possible cases due to uncertainty are considered, and a penalty slack term is introduced into the objective as a compromise between strict feasibility and optimality (Mulvey et al. (1995); Darlington et al. (1999)). Others have placed emphasis on effectively integrating advanced control layers into the RTO scheme, where robustness may be, once more, cast into the well-known control framework and included as part of the set-point optimization subproblem (Kassmann et al. (2000); Flemming et al. (2007)). However, to the authors' best knowledge, a proper criterion for guaranteed feasibility of real-time optimization schemes has not been given.

This last sentence qualifies precisely the goal of this paper, where such a criterion will be derived for a very general class of systems with Lipschitz-continuous plant constraints. For each of these systems, it will be assumed that some iterative optimization algorithm is at work - collecting measurements at a given steady-state iteration, updating the optimization problem, and then re-optimizing to obtain a new optimum for the next iteration (see Fig. 1). Although this could, in principle, be any black-box algorithm, in this paper the discussion is limited to constraint-adaptation schemes where the constraints of the model are updated from iteration to iteration through bias update terms (Marchetti et al. (2009)).

When a new optimum is calculated by the updated optimization problem, this optimum is generally considered as being more accurate than the previous (as it uses the most recent measurements), and thus better. However, the full calculated input step is usually not taken in practice for safety reasons, and a filtered step is applied instead (Brdys and Tatjewski (2005)). Although this means of updating the input is known to have robustness properties, the actual choice of the filter is generally not discussed, and is rather chosen in an *ad hoc* manner. In this work,

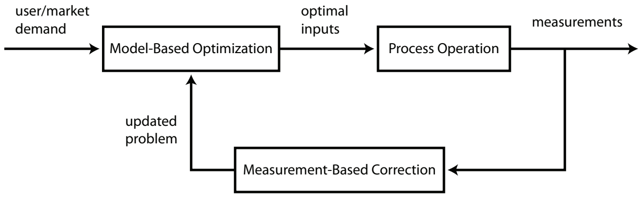

Fig. 1. A general representation of measurement-based iterative optimization.

the value for the filter gain that minimizes the amount of filtering while still guaranteeing feasibility will be given.

The structure of the paper is as follows. Section 2 gives a brief review of the constraint-adaptation scheme and demonstrates how, in general, such a scheme cannot guarantee feasibility from one iteration to the next. To allow for such guarantees, the role of the input filter is then introduced, and a proper derivation of an upper bound for this filter gain (or a lower bound on the amount of filtering) is given in Section 3. Section 4 then presents the problems that arise when trying to use this filter definition for systems with multiple constraints. Section 5 proposes a tentative solution to these problems and offers an illustrative example, and Section 6 concludes the paper.

2. RTO VIA CONSTRAINT ADAPTATION

The optimal inputs obtained by solving an RTO problem with adaptation on the constraints may be written as follows (for more in-depth formulations of this method, see Marchetti et al. (2009))[1]:

$$\mathbf{u}_k^* = \arg\min_{\mathbf{u}} . \quad \varphi(\mathbf{u}) \quad \text{s.t.} \quad \mathbf{G}(\mathbf{u},\theta) + \mathbf{\Lambda}_k \preceq \mathbf{0} \quad , \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ is the vector of inputs, $\varphi \in \mathbb{R}$ is the objective function, $\mathbf{G} \in \mathbb{R}^{n_g}$ is the vector of inequality constraints, $(\cdot)^*$ denotes the optimal input computed by the optimization, and the subscript $(\cdot)_k$ indicates the k^{th} iteration. Uncertainty in the model is represented via a set of uncertain parameters $\theta \in \mathbb{R}^{n_\theta}$, with the vector of correction terms $\mathbf{\Lambda} \in \mathbb{R}^{n_g}$ serving to correct the model. It may be assumed, without loss of generality, that φ is linear and independent of θ by virtue of the epigraph transformation (Boyd and Vanderberghe (2008)), where an uncertain or nonlinear cost may simply be reformulated as an inequality constraint.

In this paper, the correction terms will be defined simply as the plant-model error at the previous iteration:

$$\mathbf{\Lambda}_k = \mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) - \mathbf{G}(\bar{\mathbf{u}}_{k-1},\theta), \tag{2}$$

where $\mathbf{G}_p$ is the vector of measured plant constraints. The overbar $(\bar{\cdot})$ is used to denote the input that is actually applied.

Because applying $\mathbf{u}_k^*$ to the plant directly may result in a step that is too aggressive and leads to an infeasible point for the plant, an input filter - sometimes known as a "relaxation" or "gain" coefficient (Brdys and Tatjewski (2005)) - is generally used:

[1] The symbol $\preceq$ is used to denote a system of less-than-or-equal-to inequalities.

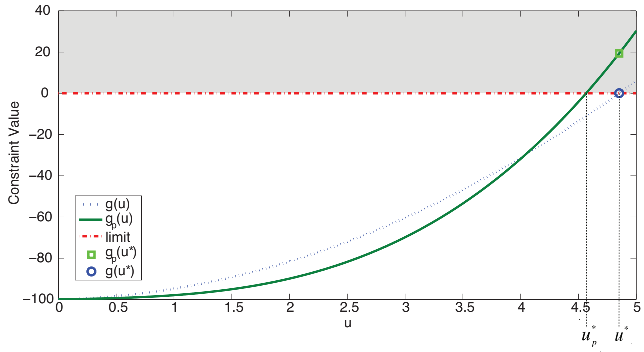

Fig. 2. The plant and model constraints in the optimization problem (4)-(5).

$$\bar{\mathbf{u}}_k = K\mathbf{u}_k^* + (1-K)\,\bar{\mathbf{u}}_{k-1}, \tag{3}$$

where $K \in [0,1]$ is the gain of an exponential filter affecting all the inputs, with $K = 1$ and $K = 0$ corresponding to no filtering and to total filtering, respectively.

A brief example is given next to illustrate the efficacy of this approach.

Example. Consider the following optimization problem:

$$\begin{aligned} \max_u . \quad & u \\ \text{s.t.} \quad & g(u) = 4u^2 + 1.2u - 100 \le 0 \\ & 0 \le u \le 5 \end{aligned} \quad , \tag{4}$$

where the goal is to maximize u, while honoring certain constraints. Assume that the real constraint for the plant is given by:

$$g_p(u) = u^3 + 1.05u - 100. \tag{5}$$

The graphical representation of this problem is given in Fig. 2, from where it is immediately evident that applying the computed optimal input would lead to a significant constraint violation for the plant (compare $g_p(u^*)$ and $g(u^*)$). To get around this issue, one may use the scheme (1)-(3) as follows:

$$\begin{aligned} \Lambda_k &= (\bar{u}_{k-1}^3 + 1.05\bar{u}_{k-1} - 100) \\ &\quad -(4\bar{u}_{k-1}^2 + 1.2\bar{u}_{k-1} - 100) \\ u_k^* &= \arg\max_u . \quad u \\ &\qquad \text{s.t.} \quad 4u^2 + 1.2u - 100 + \Lambda_k \le 0 \\ &\qquad\qquad 0 \le u \le 5 \\ \bar{u}_k &= Ku_k^* + (1-K)\bar{u}_{k-1}. \end{aligned} \tag{6}$$

Choosing the feasible initial input $\bar{u}_0 = 3.5$, one can run this algorithm iteratively and see that convergence to the true optimum is easily achieved in a relatively small number of iterations (see Fig. 3 - only the input is shown for this case, as it is clear that constraint violation, $g(\bar{u}) > 0$, will only occur if the input $\bar{u}$ goes above the optimal value u_p^*).

It is clear from this simple example that the filter can play a crucial role in deciding both the feasibility and convergence speed properties. In the next section, an upper

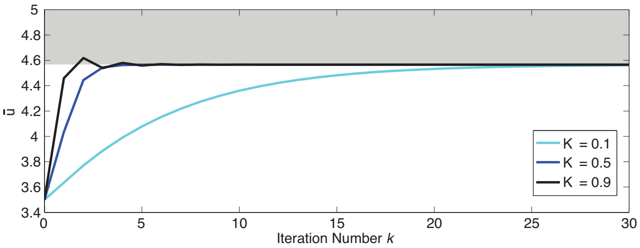

Fig. 3. The solution to (6) with different filter gains.

bound on the filter gain that guarantees feasibility from iteration to iteration is derived.

3. UPPER BOUND ON THE FILTER GAIN

The following theorem is the main result of this work, and derives a bound on the filter gain K_k such that feasibility is guaranteed from iteration to iteration. The subscript $(\cdot)_k$ is added to the filter gain so as to indicate that the value is iteratively adapted (i.e. K_k is the filter gain that will guarantee plant feasibility at iteration k).

Theorem. Assume that all the plant constraints in the vector $\mathbf{G}_p(\mathbf{u}) = \left[\mathbf{g}_{p,1}(\mathbf{u}), \mathbf{g}_{p,2}(\mathbf{u}), ..., \mathbf{g}_{p,n_g}(\mathbf{u})\right]^{\mathrm{T}}$ are Lipschitz-continuous functions, i.e.:

$$\begin{array}{l} \exists \kappa_i \in [0, \infty) : \\ |g_{p,i}(\mathbf{u}_a) - g_{p,i}(\mathbf{u}_b)| \leq \kappa_i \|\mathbf{u}_a - \mathbf{u}_b\| \quad , \\ \forall \mathbf{u}_a, \mathbf{u}_b \in \mathbf{U}; \forall i = 1, ..., n_g \end{array} \tag{7}$$

where $\mathbf{u}_a$ and $\mathbf{u}_b$ are any two points in the input space $\mathbf{U}$.

Then, the RTO method given by (1)-(3) will be feasible at the k^{th} iteration if:

$$K_k \leq \min_i \left[\frac{-g_{p,i}(\bar{\mathbf{u}}_{k-1})}{\kappa_i \|\mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1}\|} \right]. \tag{8}$$

Proof. The proof will use the feasibility of the initial point (assumed) and iteratively extend it to all future iterations. It will be assumed that the system is feasible at the $(k-1)^{\mathrm{st}}$ iteration, i.e.:

$$\mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) \preceq \mathbf{0}, \tag{9}$$

and a condition on K_k that ensures the feasibility at the k^{th} iteration will be determined.

The Lipschitz relation (7) allows bounding the value of the plant constraint at the k^{th} iteration (the absolute value has been removed as it is unnecessary here):

$$\mathbf{G}_p(\bar{\mathbf{u}}_k) \preceq \mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) + \boldsymbol{\kappa} \|\bar{\mathbf{u}}_k - \bar{\mathbf{u}}_{k-1}\|, \tag{10}$$

where $\boldsymbol{\kappa} = \left(\kappa_1, \kappa_2, ..., \kappa_{n_g}\right)^{\mathrm{T}}$. It is clear from (10) that the feasibility criterion at k, $\mathbf{G}_p(\bar{\mathbf{u}}_k) \preceq \mathbf{0}$, will be automatically satisfied if:

$$\mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) + \boldsymbol{\kappa} \|\bar{\mathbf{u}}_k - \bar{\mathbf{u}}_{k-1}\| \preceq \mathbf{0}. \tag{11}$$

Using the filter law (3) for $\bar{\mathbf{u}}_k$ gives:

$$\begin{array}{l} \mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) \\ \quad + \boldsymbol{\kappa} \|K_k \mathbf{u}_k^* + (1 - K_k)\,\bar{\mathbf{u}}_{k-1} - \bar{\mathbf{u}}_{k-1}\| \preceq \mathbf{0} \end{array}, \tag{12}$$

or

$$\mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) + \boldsymbol{\kappa} K_k \|\mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1}\| \preceq \mathbf{0}\,. \tag{13}$$

This is a system of inequalities that, if treated component-wise and rewritten for K_k, will yield the following individual bounds:

$$K_k^{g_i} \leq \frac{-g_{p,i}(\bar{\mathbf{u}}_{k-1})}{\kappa_i \|\mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1}\|}, \tag{14}$$

where $g_{p,i}(\bar{\mathbf{u}}_{k-1})$ is simply the measurement of the ith constraint at the previous iteration $k-1$.

As K_k is a scalar, it suffices to take the component-wise minimum of the $K_k^{g_i}$ values in (14) for (13) to be satisfied. This thus leads to the condition (8) for K_k. □

Two remarks are in order:

(i) Note that (8) represents a *sufficient* condition for feasibility.
(ii) Although this result is illustrated in the context of RTO via constraint adaptation, it can be used for any measurement-based, iterative optimization scheme, provided that the assumption regarding Lipschitz continuity of the plant constraints holds.

4. SUB-OPTIMAL CONVERGENCE IN THE MULTI-CONSTRAINT CASE

It may be tempting to use the derived criterion (8) directly in the iterative optimization when defining a filter gain value, and indeed, this gives excellent results for cases with only a single constraint. Unfortunately, sub-optimal convergence is seen in the case of multiple constraints.

4.1 Single-Constraint Case

Consider, once more, the example given earlier.

The Simple Example Revisited. Returning to Problem (6), one starts by initializing at the feasible point $u_0^* = \bar{u}_0 = 3.5$, and then applying the proposed filter gain calculation at each iteration. The maximum slope of the plant constraint over the relevant input space is used as the Lipschitz bound ($\kappa = 76.05$).

With the convergence criterion defined as $\|\mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1}\|_2 \leq 0.01$, it is shown in Fig. 4 that this is sufficient for the plant to converge safely in 3 iterations. Fig. 4 also shows the performance of the "ideal" filter gain of $K = 0.7$, found here by trial and error, that would yield safe convergence in a single iteration for this one-dimensional case.

4.2 Multi-Constraint Case

Unfortunately, the results obtained above cannot be retained for multi-constraint problems. The intuitive reason for this is simple: some constraints enjoy a faster convergence rate than others, and the constraints do not converge simultaneously. The consequences of this are stated in the following corollary.

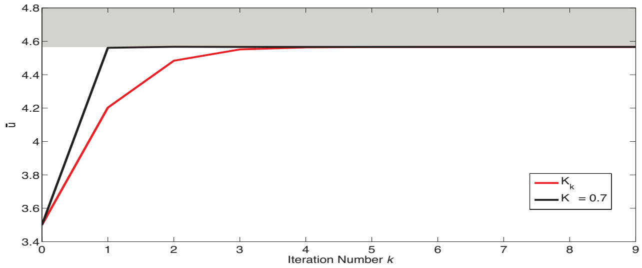

Fig. 4. Solution to (6) using the proposed approach.

Corollary. Let one of the plant constraints becomes active at iteration $k-1$,

$$\begin{aligned} &\exists g_{p,a} \in \mathbf{G}_p : \\ &g_{p,a}(\bar{\mathbf{u}}_{k-1}) = 0. \end{aligned} \tag{15}$$

Then, an algorithm that bounds the filter gain according to (8) will converge to $\bar{\mathbf{u}}_\infty = \bar{\mathbf{u}}_{k-1}$.

Proof. Defining the filter gain by its upper bound (the least conservative gain that guarantees feasibility):

$$K_k := \min_i \left[\frac{-g_{p,i}(\bar{\mathbf{u}}_{k-1})}{\kappa_i \left\| \mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1} \right\|} \right], \tag{16}$$

it is easily seen that $K_k = 0$, since $K_k \in [0,1]$ and

$$K_k^{g_a} = \frac{-g_{p,a}(\bar{\mathbf{u}}_{k-1})}{\kappa_a \left\| \mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1} \right\|} = 0. \tag{17}$$

By the filter law (3),

$$\bar{\mathbf{u}}_k = K_k \mathbf{u}_k^* + (1 - K_k)\, \bar{\mathbf{u}}_{k-1} = \bar{\mathbf{u}}_{k-1}. \tag{18}$$

This may be extended to all future iterations to show that $\bar{\mathbf{u}}_\infty = \bar{\mathbf{u}}_{k-1}$, even when $\bar{\mathbf{u}}_k^* \neq \bar{\mathbf{u}}_{k-1}$, thereby forcing the algorithm to converge prematurely. □

It is clear that premature convergence is impossible for the single-constraint case due to a single filter and a single active constraint, but becomes an issue when multiple active constraints are present.

Take the following example:

$$\begin{aligned} \min_{u_1,u_2} . \quad & u_1 + u_2 \\ \text{s.t.} \quad & -u_1^3 \leq 0 \\ & -u_1 - 2u_2 \leq 0 \end{aligned} \quad , \tag{19}$$

with the initial inputs of (1,1). Assume that the constraints in (19) already include ideal correction terms, and that solving at each iteration will give the true optimum of (0,0). If $\kappa = 5$ is used for both constraints and the filtering scheme is applied, the convergence behavior given in Table 1 is observed.

Clearly, the constraints converge at different rates due to their differing structures (one is cubic, while the other is linear). While g_1 has practically converged after 100 iterations, g_2 has not, and a significant offset is noted from the true optimum as a result. This is because the convergence of a single constraint pushes the filter gain, as defined in (8), to $K_k = 0$. While it appears that K_k never becomes 0 exactly (notice that the algorithm is still moving after 10,000 iterations), it would be impractical to rely on this sort of asymptotic convergence due to both measurement noise and time constraints that would arise in real applications. In a realistic scenario, $g_{p,1}$ would likely be declared as being exactly 0 after 100 iterations, if not sooner, and the algorithm would effectively converge with $K_k = 0$ exactly. $g_{p,2}$ would remain inactive and some optimality losses would be incurred.

Table 1. Convergence behavior for the example in (19).

k	$\bar{u}_1$	$\bar{u}_2$	$g_{p,1}$	$g_{p,2}$	K_k	Opt. Loss
0	1.00	1.00	-1.00	-3.00	0.141	100%
1	0.86	0.86	-0.63	-2.58	0.104	86%
2	0.77	0.77	-0.45	-2.31	0.084	77%
3	0.70	0.70	-0.35	-2.11	0.070	70%
4	0.66	0.66	-0.28	-1.97	0.061	66%
100	0.18	0.18	-0.01	-0.55	0.005	18%
1000	0.06	0.06	-0.00	-0.18	0.001	6%
10000	0.02	0.02	-0.00	-0.06	0.000	2%

For cases where the main reduction in optimality loss may be achieved prior to this sub-optimal convergence (the loss would be 18% in the example above if K_k is defined as 0 after 100 iterations - see Table 1), the amount of sub-optimality may be acceptable. However, for cases where full convergence is desired, an algorithm that trades absolute feasibility for optimality is proposed next.

5. OPTIMALITY VIA STEPWISE CONVERGENCE

5.1 *The Stepwise Convergence Algorithm*

In this scheme, the filter gain is adapted as proposed, and the smallest component-wise filter is applied at each iteration. As stated above, this will generally lead to one constraint converging before the rest. When this occurs, that constraint is simply removed from the subsequent filter calculations. The algorithm may be outlined as follows:

(i) Specify $\boldsymbol{\kappa}$ and choose a feasible $\mathbf{u}_0^*$. Set $\bar{\mathbf{u}}_0 = \mathbf{u}_0^*$ and apply it to the plant. Set $k := 1$.
(ii) Use measurements to compute $\boldsymbol{\Lambda}_k = \mathbf{G}_p(\bar{\mathbf{u}}_{k-1}) - \mathbf{G}(\bar{\mathbf{u}}_{k-1}, \theta)$.
(iii) Solve (1) to obtain $\mathbf{u}_k^*$.
(iv) Compute all the $K_k^{g_i}$ as given by (14).
(v) Check if $-g_{p,i}(\bar{\mathbf{u}}_{k-1}) \leq \delta$, $\forall i \in [1, n_g]$, where δ is some convergence threshold. If yes, ignore the corresponding $K_k^{g_i}$ in the following step.
(vi) Calculate the largest possible filter gain as given by (8), compute $\bar{\mathbf{u}}_k$ according to (3), and apply it to the plant.
(vii) Check if the entire process has converged with respect to the overall criterion $\left\| \mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1} \right\|_2 \leq \delta_u$. If not, set $k := k+1$ and return to Step 2.

Note that absolute feasibility cannot be guaranteed for this scheme, as steps are still being made by the iterative RTO after a single constraint has converged. The maximum violation in this case may be bounded as follows.

Proposition. Let $g_{p,1}(\mathbf{u}), g_{p,2}(\mathbf{u}) \in \mathbf{G}_p(\mathbf{u})$, and let $K_k^{g_1}$ and $K_k^{g_2}$ be filter gain values that, at iteration k, will guarantee feasibility for $g_{p,1}(\mathbf{u})$ and $g_{p,2}(\mathbf{u})$, respectively, as defined by (14). If $K_k = K_k^{g_2}$ when $g_{p,1}(\bar{\mathbf{u}}_{k-1}) = 0$ ($K_k^{g_1}$ is ignored in step (v) of the algorithm above), then the maximum value of $g_{p,1}(\bar{\mathbf{u}}_k)$ will be bounded by the relation:

$$g_{p,1}(\bar{\mathbf{u}}_k) \leq -\frac{\kappa_1}{\kappa_2} g_{p,2}(\bar{\mathbf{u}}_{k-1}), \tag{20}$$

This corresponds to the maximum possible violation in $g_{p,1}(\mathbf{u})$ if a filter gain larger than the minimum component-wise gain is used.

Proof. The proof follows readily from the component-wise version of condition (10) for constraint $g_{p,1}(\mathbf{u})$:

$$g_{p,1}(\bar{\mathbf{u}}_k) \leq g_{p,1}(\bar{\mathbf{u}}_{k-1}) + \kappa_1 \left\| \bar{\mathbf{u}}_k - \bar{\mathbf{u}}_{k-1} \right\|, \tag{21}$$

where substituting the filter law (3) leads to:

$$g_{p,1}(\bar{\mathbf{u}}_k) \leq g_{p,1}(\bar{\mathbf{u}}_{k-1}) + \kappa_1 K_k \left\| \mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1} \right\|. \tag{22}$$

It is now supposed that the filter gain K_k comes from the second constraint:

$$K_k = \frac{-g_{p,2}(\bar{\mathbf{u}}_{k-1})}{\kappa_2 \left\| \mathbf{u}_k^* - \bar{\mathbf{u}}_{k-1} \right\|}. \tag{23}$$

Substituting this filter gain into (22) yields:

$$g_{p,1}(\bar{\mathbf{u}}_k) \leq g_{p,1}(\bar{\mathbf{u}}_{k-1}) - \frac{\kappa_1}{\kappa_2} g_{p,2}(\bar{\mathbf{u}}_{k-1}). \tag{24}$$

This is a useful general result, but the case of interest is for when $g_{p,1}(\bar{\mathbf{u}}_{k-1})$ has converged to 0, which then returns (20). □

For cases when these violations are expected to be small and/or can be offset by a properly decoupled controller, or for cases where a good local model is available for the active constraint, this proposed scheme may be effective in keeping violations to a minimum while avoiding sub-optimal convergence.

5.2 Illustrative Example

The following 6-input-4-constraint case is considered:

$$\begin{aligned} \max_{\mathbf{u}}.\quad & 0.45u_1 + 0.04u_2 + 0.88u_3 + 0.69u_4 \\ & +0.95u_5 + 0.56u_6 \\ \text{s.t.}\quad & g_1(\mathbf{u}) = 2.2u_1^2 + e^{3u_3} + 0.9u_5^2 \\ & \qquad -0.9u_6 - 0.8 \leq 0 \\ & g_2(\mathbf{u}) = 1.1u_2^2 - 1.1u_3 + 0.9u_4^2 \\ & \qquad -u_4 - 2 \leq 0 \\ & g_3(\mathbf{u}) = -2u_2 + e^{1.5u_5} + 0.8u_6^2 - 2 \leq 0 \\ & g_4(\mathbf{u}) = -1.2u_3 + u_4 + 2.5u_6^2 - 4 \leq 0 \\ & 0 \preceq \mathbf{u} \preceq 1 \end{aligned}, \tag{25}$$

with the plant constraints given as:

$$\begin{aligned} g_{p,1}(\mathbf{u}) &= 2u_1^2 + e^{2u_3} + u_5^2 - u_6 - 0.8 \\ g_{p,2}(\mathbf{u}) &= u_2^2 - u_3 + u_4^2 - u_5 - 2 \\ g_{p,3}(\mathbf{u}) &= -u_1 - u_2 + e^{u_5} + u_6^2 - 2 \\ g_{p,4}(\mathbf{u}) &= -u_3 + u_4^2 + 3u_6^2 - 4 \end{aligned}. \tag{26}$$

The constraint-adaptation scheme in (1)-(3) is used for the iterative RTO. The appropriate Lipschitz constants are defined as $\boldsymbol{\kappa} = [14.78, 2, 2.72, 6]^{\mathrm{T}}$ for this example. As will be seen, constraints 1, 3, and 4 are active at the optimum, while constraint 2 remains inactive for the duration of operation. The goal is to start at the initial feasible, sub-optimal point of $\mathbf{u}_0^* = (0.29, 0.94, 0.07, 1.00, 0.43, 0.98)$ and to converge to the optimum governed by a set of active constraints.

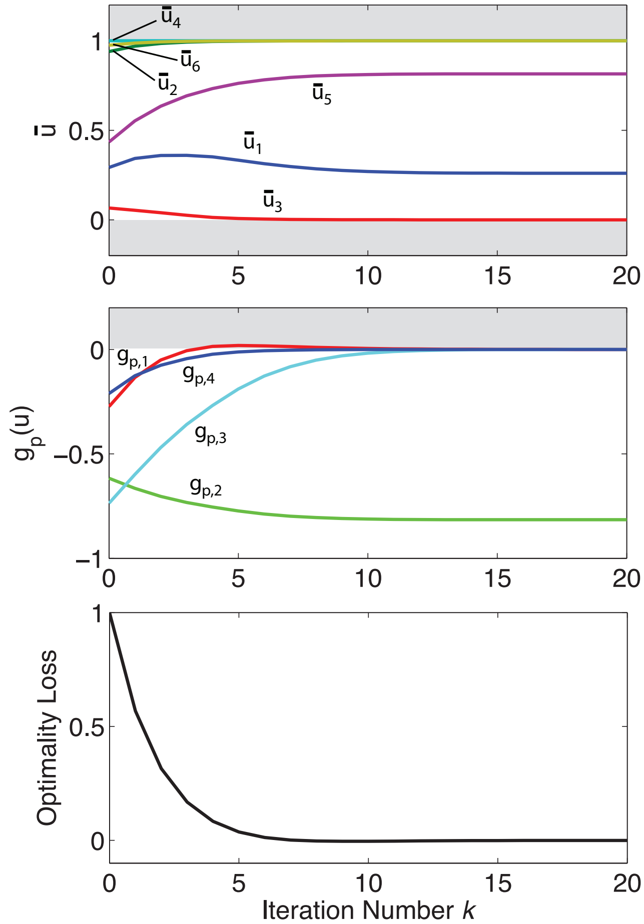

Fig. 5. Convergence results with $K = 0.5$ for the problem (25)-(26). Fast convergence is attained, but feasibility is compromised for $g_{p,1}$.

Two cases are studied. In the first, an *ad hoc* constant filter gain of 0.5 is used, while the second employs the stepwise convergence scheme, with a convergence threshold of $\delta = 0.005$ used to decide when a constraint should be removed from the filter gain analysis.

The results are presented in Figs. 5 and 6 for the two cases, respectively. The use of a constant preset filter gain results in significantly faster convergence than the feasibility-seeking scheme, but compromises the feasibility of the first constraint during convergence. Whether this violation is grievous or whether it is acceptable depends, of course, on the specific context and application. The stepwise approach follows the one-by-one constraint convergence as prescribed, but also incurs some violations during the convergence of $g_{p,4}$ in the second step. As should be expected, the price of improved feasibility is a significant increase in the number of iterations needed to converge.

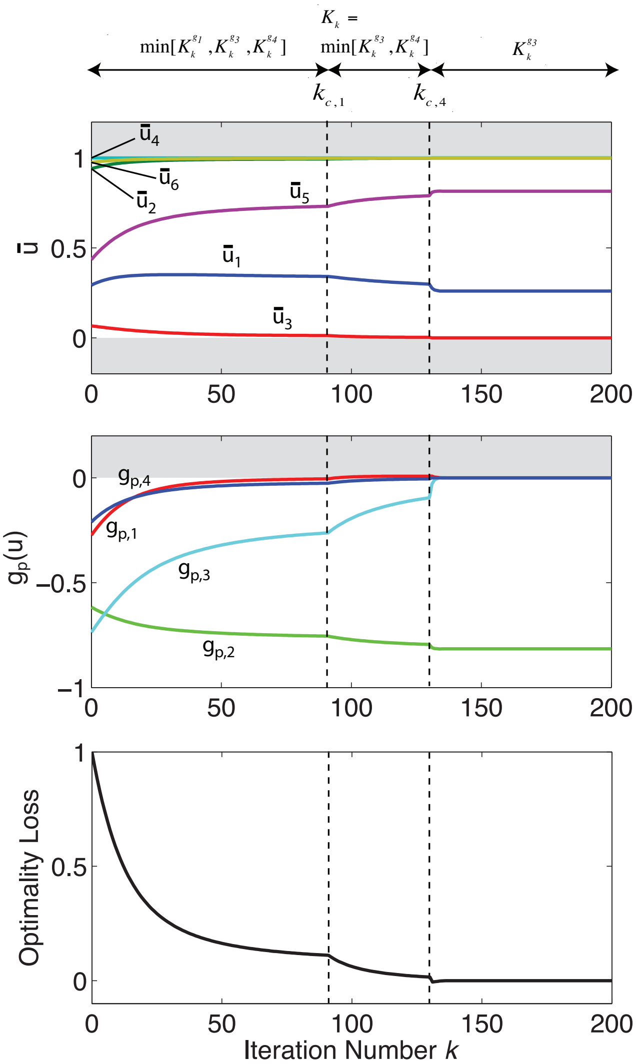

Fig. 6. Convergence results with adaptive filtering and the stepwise convergence scheme for the problem (25)-(26). Note the successive activation of the constraints $g_{p,1}$, $g_{p,4}$, and $g_{p,3}$ ($k_{c,1}$ and $k_{c,4}$ denote the instances where $g_{p,1}$ and $g_{p,4}$ have converged, respectively). A violation in $g_{p,1}$ is seen during the second step.

It is important to note that, if one wanted to guarantee absolute feasibility for this example, it would be sufficient to stop after approximately 90 iterations for the stepwise scheme (the constraint $g_{p,1}$ would approximately reach 0, but its respective filter would *not* be removed from analysis). This would effectively lead to sub-optimal convergence, with both $g_{p,4}$ and $g_{p,3}$ inactive. However, note that the majority of the optimality losses have already been diminished by that point (Fig. 6), and so the sub-optimality would not be great. This point is especially relevant for systems with frequent market perturbations where the cost function is modified often and where the system will not stay at the sub-optimal converged state for long anyway.

6. CONCLUSIONS

This paper has addressed the use of input filtering to guarantee absolute feasibility in iterative, measurement-based RTO algorithms. Using the concepts of Lipschitz continuity, a sufficient feasibility condition was found by placing an upper bound on the filter gain. Although the implementation of this bound into a working algorithm is not entirely straightforward for multi-constraint systems, where it may lead to sub-optimal convergence, one such implementation was proposed here with stepwise constraint-by-constraint convergence and reduced constraint violations. This was demonstrated for a simulated 6-input-4-constraint problem, and it was shown that accepting sub-optimal convergence to maintain absolute feasibility would not lead to major optimality losses for this case.

It is acknowledged that the results, while of potential academic interest, may still require modifications to be implemented practically. This is because feasibility, while important, will not always outweigh the other important performance factor - speed. A relaxation of the absolute feasibility criteria so as to allow for an intuitive, "tunable" tradeoff between feasibility and speed is something yet to be addressed. The results presented here should, however, be of significant use for RTO problems where prolonged constraint violations are costly.

REFERENCES

Boyd, S. and Vanderberghe, L. (2008). *Convex Optimization.* Cambridge University Press.

Brdys, M. and Tatjewski, P. (2005). *Iterative Algorithms for Multilayer Optimizing Control.* Imperial College Press.

Darlington, J., Pantelides, C., Rustem, B., and Tanyi, B. (1999). An algorithm for constrained nonlinear optimization under uncertainty. *Automatica*, 35, 217–228.

Engell, S. (2007). Feedback control for optimal process operation. *J. Process Contr.*, 17, 203–219.

Flemming, T., Bartl, M., and Li, P. (2007). Set-point optimization for closed-loop control systems under uncertainty. *Ind. Eng. Chem. Res.*, 46, 4930–4942.

Kassmann, D., Badgwell, T., and Hawkins, R. (2000). Robust steady-state target calculation for model predictive control. *AIChE Journal*, 46(5), 1007–1024.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *Ind. Eng. Chem. Res.*, 48(13), 6022–6033.

Mulvey, J., Vanderbei, R., and Zenios, S. (1995). Robust optimization of large-scale systems. *Operations Research*, 43(2), 264–281.

Tatjewski, P. (2008). Advanced control and on-line process optimization in multilayer structures. *Annual Reviews in Control*, 32, 71–85.

Zhang, Y. (2010). Robust dynamic real-time optimization under parametric uncertainty for plant-wide processes. In *8th World Congress on Intelligent Control and Automation, in Jinan, China.*

Zhang, Y., Monder, D., and Forbes, J.F. (2002). Real-time optimization under parametric uncertainty: A probability constrained approach. *Journal of Process Control*, 12, 373–389.