On Handling Cost Gradient Uncertainty in Real-Time Optimization

Martand Singhal, Timm Faulwasser and Dominique Bonvin

Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne
CH-1015 Lausanne, Switzerland
e-mail: {martand.singhal, timm.faulwasser, dominique.bonvin}@epfl.ch

Abstract: This paper deals with the real-time optimization of uncertain plants and proposes an approach based on surrogate models to reach the *plant* optimum when the plant cost gradient is imperfectly known. It is shown that, for processes with only box constraints, the optimum is reached upon convergence if the multiplicative gradient uncertainty lies within some bounded interval. For the case of general constraints, conditions are derived that guarantee plant feasibility and, in principle, allow enforcing cost decrease at each iteration.

Keywords: real-time optimization, uncertain gradients, convergence, monotonic cost decrease

1. INTRODUCTION

Steadily increasing economic competition and growing environmental concern explain the need for safe plant operation close to constraints. The goal of real-time optimization (RTO) is to enforce plant optimality in the presence of uncertainty such as plant-model mismatch and disturbances. Instead of searching for a robust solution to the problem, RTO methods rely on measurements to push the *plant* toward optimality.

There exist three main classes of RTO schemes: (i) one can update the model parameters and repeat the optimization with the updated model (Jang et al., 1987; Chachuat et al., 2009); (ii) alternatively, one can compute correction terms and modify the optimization problem accordingly; for example, the use of first-order correction terms allows enforcing plant optimality upon convergence (Gao and Engell, 2005; Marchetti et al., 2009); or, (iii) one can use feedback control and adapt the inputs directly (Skogestad, 2000; Srinivasan and Bonvin, 2007).

This paper investigates feasibility and optimality features for Class-(ii) schemes, which use possibly inaccurate surrogate models. In our particular case, the surrogate model consists of a convex quadratic approximation to the plant cost. The constraints in the surrogate model are linear functions constructed with the help of Lipschitz constants for the plant constraints. Similar to modifier adaptation (Marchetti et al., 2009, 2010; Gao and Engell, 2005), the proposed scheme requires information on plant gradients. Since the estimation of plant gradients from measurements is quite challenging in practice (Marchetti et al., 2010), one can regard the common assumption of exactly known plant gradients as being very restrictive. Here, we try to relax this assumption, at least partially, and we consider uncertainty in the cost-gradient estimates. We establish monotonic cost decrease in the presence of component-wise multiplicative gradient uncertainty for the case of only box constraints. Furthermore, we show that iterative plant feasibility can be guaranteed for the case of general constraints, provided, the gradients and the Lipschitz constants for the plant constraints are known.

The paper is structured as follows. Section 2 briefly discusses the RTO problem formulation. Section 3 shows how uncertainty on the cost gradient can be handled in RTO. A novel RTO approach guaranteeing plant feasibility is presented in Section 4. The same section also discusses conditions under which RTO generates successive iterates with a decrease in the plant cost.

2. PROBLEM STATEMENT

Steady-state performance improvement can be formulated mathematically as a nonlinear program

$$\min_{u} \phi(u, \mathfrak{d}) \tag{1a}$$

$$\text{s.t.} \quad g_j(u, \mathfrak{d}) \leq 0, \qquad j{=}1,..,n_g \tag{1b}$$

$$u^L \preceq u \preceq u^U, \tag{1c}$$

where u is the n_u-dimensional input vector, $\phi : \mathbb{R}^{n_u} \to \mathbb{R}$ is the cost, and $g_j : \mathbb{R}^{n_u} \to \mathbb{R}$ is the j^{th} constraint and $\mathfrak{d}$ is the n_d dimensional vector of disturbances. The subscript $(\prec)$ denotes component-wise inequality of vectors. The disturbance $\mathfrak{d}$ models the fact that for real systems the plant cost and constraints may change with time. For the rest of the paper we do not deal with the disturbance $\mathfrak{d}$ explicitly. Rather, we assume that the plant cost ϕ and the plant constraints g_j are not exactly known. Therefore, only approximate gradient information is available which is key to finding $u^\star$, the optimum of the RTO problem (1).

This paper investigates a novel RTO strategy for handling uncertain gradient information for the cost function. The scheme is based on a quadratic upper bound on the cost that can deal explicitly with uncertain gradients.

3. RTO WITH BOX CONSTRAINTS

We first address the case of the RTO problem (1) without the general constraints (1b), that is, only with the box constraints (1c). Let u_k be the input applied to the plant at the k^{th} RTO iteration and $\Delta_{k+1} = (u_{k+1} - u_k)$ the difference between two successive RTO inputs. The following lemma will be crucial for our later developments.

Lemma 1. Let $\phi : \mathbb{R}^{n_u} \to \mathbb{R}$ be twice continuously differentiable over the compact set $\mathcal{U} \subset \mathbb{R}^{n_u}$ such that

$$-M_{ij} < \left.\frac{\partial^2 \phi}{\partial u_i \partial u_j}\right|_u < M_{ij}, \quad \forall u \in \mathcal{U}, \quad i,j = 1, ..., n_u.$$

Then, the change in ϕ between u_k and u_{k+1} is bounded as

$$\phi(u_{k+1}) \leq \phi(u_k) + \nabla\phi(u_k)^T \Delta_{k+1} + \frac{1}{2}\Delta_{k+1}^T \bar{Q} \Delta_{k+1}, \quad (2)$$

where $\bar{Q} \succ 0$ is a diagonal matrix with the diagonal elements $\bar{Q}_{ii} = \sum_{j=1}^{n_u} M_{ij}, \quad i = 1, ..., n_u$.

Proof. The proof can be found in Bunin et al. (2013). □

The matrix $\bar{Q}$ is called a quadratic upper bound on ϕ. Based on Lemma 1, we subsequently develop a simple RTO strategy that accounts for gradient uncertainty. We propose to solve the following problem

$$\min_{\Delta_{k+1}} \nabla\hat{\phi}_k^T \Delta_{k+1} + \frac{1}{2}\Delta_{k+1}^T \bar{Q} \Delta_{k+1}, \quad (3)$$

where $\nabla\hat{\phi}_k$ is the estimated gradient of the plant cost w.r.t. u_k. Problem (3) has the analytical solution

$$\Delta_{k+1}^\star = -\bar{Q}^{-1}\nabla\hat{\phi}_k, \quad (4)$$

which allows setting up the next RTO iterate as

$$u_{k+1} = u_k - \bar{Q}^{-1}\nabla\hat{\phi}_k. \quad (5)$$

We show next that the RTO iterates given by (5) converge to a KKT point for the plant cost. For this, let us introduce three assumptions and a lemma.

Assumption 1. ($\mathcal{C}^2$ cost). The cost ϕ is twice continuously differentiable on an open set containing the input space $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : u^L \preceq u \preceq u^U\}$. Furthermore, a global quadratic upper bound $\bar{Q}$ is available.

Assumption 2. The solution $\Delta_{k+1}^\star$ obtained in (4) is such that $\Delta_{k+1}^\star \in \mathcal{U}_k$, with

$$\mathcal{U}_k = \left\{\Delta \in \mathbb{R}^{n_u} : \Delta \in \left[u^L - u_k, u^U - u_k\right]\right\}. \quad (6)$$

Assumption 2 implies that the RTO input u_{k+1} given by (5) satisfies the box constraints given in (1c). This assumption may sound restrictive. But note that, if the update u_{k+1} given by (5) does not satisfy the box constraints, we can multiply (4) by a scalar $0 < \alpha < 1$ so that the update $u_{k+1} \in \partial\mathcal{U}$, where $\partial\mathcal{U}$ is the boundary of the set $\mathcal{U}$.

For the third assumption we consider a vector $\gamma_k = (\gamma_{k,1}, ...\gamma_{k,n_u})^T$ and define the set $\mathcal{G}_k \subset \mathbb{R}^{n_u}$ as

$$\mathcal{G}_k = \left\{\operatorname{diag}(\gamma_k)\nabla\phi_k, \gamma_{k,i} \in [\varepsilon, 2-\varepsilon], i = 1, .., n_u\right\}, \quad (7)$$

where $\varepsilon \in (0, 1)$. Let $\Gamma_k \nabla\phi_k \in \mathcal{G}_k$ represent an element of the set $\mathcal{G}_k$, with $\Gamma_k = \operatorname{diag}(\gamma_k)$.

Assumption 3. (Bounded gradient estimate). For all $k \in \mathbb{N}$, the estimate of the plant cost gradient at u_k, $\nabla\hat{\phi}_k \in \mathbb{R}^{n_u}$, satisfies $\nabla\hat{\phi}_k \in \mathcal{G}_k$, with the set $\mathcal{G}_k$ describing the gradient uncertainty. Any gradient estimate $\nabla\hat{\phi}_k$ satisfying $\nabla\hat{\phi}_k \in \mathcal{G}_k$ is said to be an admissible gradient vector.

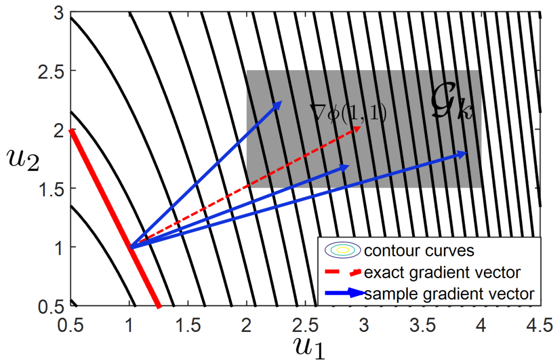

Fig. 1. The set $\mathcal{G}_k$ at $u = (1,1)^T$ for (8) is shown as the shaded area.

To illustrate the set $\mathcal{G}_k$, consider the plant cost

$$\phi(u) = u_1^2 + u_2. \quad (8)$$

The cost gradient at $u = (1,1)^T$ is $\nabla\phi(u) = (2,1)^T$. Figure 1 illustrates $\mathcal{G}_k$ as the shaded grey region corresponding to the gradient vector shown by dotted red arrow. For illustration purposes, we choose here a large value $\varepsilon = 0.5$. The vectors in blue solid arrows are sample vectors that belong to $\mathcal{G}_k$. The descent directions are obtained by reflecting any vector $v \in \mathcal{G}_k$ across the separating hyperplane constructed at $u = (1,1)^T$. The separating hyperplane given by $2u_1 + u_2 = 3$ is shown with solid red line in Figure 1. The contour curves are depicted by solid black curves. This observation leads to the following technical lemma.

Lemma 2. Consider the set of descent directions of ϕ at u_k

$$\mathcal{D}_k = \left\{d \in \mathbb{R}^{n_u} : \nabla\phi_k^T d < 0\right\}, \quad (9)$$

and the set

$$\mathcal{G}_k^- = \left\{v \in \mathbb{R}^{n_u} : -v \in \mathcal{G}_k\right\}. \quad (10)$$

Then, the inclusion $\mathcal{G}_k^- \subset \mathcal{D}_k$ holds.

Proof. From (10) and (7), the inner product of the gradient vector $\nabla\phi_k$ with $v \in \mathcal{G}_k^-$, satisfies

$$\nabla\phi_k^T v = \nabla\phi_k^T(-\Gamma_k\nabla\phi_k).$$

Furthermore, since Γ_k is a positive definite matrix, we have $\nabla\phi_k^T v = -\nabla\phi_k^T\Gamma_k\nabla\phi_k < 0, \forall v \in \mathcal{G}_k^-$ and therefore, from (9), $v \in \mathcal{D}_k$. Hence, $\mathcal{G}_k^- \subset \mathcal{D}_k$. □

The message of Lemma 2 is that $-\nabla\hat{\phi}_k \in \mathcal{G}_k^-$ implies that $-\nabla\hat{\phi}_k \in \mathcal{D}_k$. In other words, the negative of the estimated gradient satisfying Assumption 3 is a descent direction for the cost ϕ at u_k. One may conclude that moving in this direction guarantees cost decrease. However, this is not generally true, even for convex functions. The crucial parameter to be determined is the step length. The next proposition shows that taking iterations according to (5) not only ensures moving in the descent direction but also guarantees that the step lengths are controlled and result in cost decrease.

Proposition 1. (Convergence with uncertain gradients). Consider Problem (1) without the constraints (1b). Let Assumptions 1–3 hold, the initial input be $u_0 \in \mathcal{U}$, and the RTO iterates be computed according to (5). Then,

(i) the plant cost decreases monotonically at each iteration, and
(ii) upon convergence to $u^\star \in int(\mathcal{U})$, $u^\star$ is a KKT point of the plant.

Proof. From(2) and (4) we have,

$$\phi(u_{k+1}) - \phi(u_k) \leq -\nabla\phi_k^T \bar{Q}^{-1}\nabla\hat{\phi}_k + \frac{1}{2}\nabla\hat{\phi}_k^T \bar{Q}^{-1}\nabla\hat{\phi}_k.$$

The definition of the set $\mathcal{G}_k$ in (7) allows writing the gradient estimate $\nabla\hat{\phi}_k$ in terms of the true gradient $\nabla\phi_k$

$$\nabla\hat{\phi}_k = \Gamma_k\nabla\phi_k \in \mathcal{G}_k. \tag{11}$$

Therefore,

$$\phi(u_{k+1}) - \phi(u_k) \leq -\nabla\phi_k^T \bar{Q}^{-1}\Gamma_k\nabla\phi_k + \frac{1}{2}\nabla\phi_k^T\Gamma_k \bar{Q}^{-1}\Gamma_k\nabla\phi_k$$

or,

$$\phi(u_{k+1}) - \phi(u_k) \leq -\nabla\phi_k^T(\Gamma_k - \frac{1}{2}\Gamma_k^2)\bar{Q}^{-1}\nabla\phi_k. \tag{12}$$

Note that $\Gamma_k - \frac{1}{2}\Gamma_k^2$ and $\bar{Q}^{-1}$ are positive definite. Hence, the right-hand side of (12) is strictly negative. This proves the monotonic decrease of the plant cost. If the sequence $\{u_k\}$ converges to $u^\star \in int(\mathcal{U})$, then, since the plant cost ϕ is continuous, the sequence $\{\phi(u_k)\}$ converges to $\phi(u^\star)$. This implies that $\phi(u_{k+1}) - \phi(u_k) \to 0$ and, from (12), $\nabla\phi_k \to 0$. Hence, upon convergence, a KKT point of the plant is reached. □

Proposition 1 certifies that the knowledge of a quadratic upper bound on the plant cost can be used to guarantee cost decrease in the presence of bounded gradient uncertainty.

4. RTO WITH GENERAL CONSTRAINTS

We now consider RTO problems with both the general constraints (1b) and the box constraints (1c). Note that these constraints are hard constraints that need to be respected.

We start with a condition that provides a bound on the evolution of the constraint functions.

Lemma 3. Let $g_j : \mathbb{R}^{n_u} \to \mathbb{R}$ be continuously differentiable over the compact set $\mathcal{U} \subset \mathbb{R}^{n_u}$ such that

$$-\lambda_{i,j} < \left.\frac{\partial g_j}{\partial u_i}\right|_u < \lambda_{i,j}, \quad \forall u \in \mathcal{U}, \quad i = 1, ..., n_u, \tag{13}$$

where λ are the univariate Lipschitz constants of g_j. Then, the evolution of g_j between two successive inputs u_k and u_{k+1} is bounded by

$$g_j(u_{k+1}) \leq g_j(u_k) + \sum_{i=1}^{n_u}\lambda_{i,j}|u_{k+1,i} - u_{k,i}|. \tag{14}$$

Proof. The proof can be found in Bunin et al. (2013). □

Condition (14) can be used to enforce feasibility by computing u_{k+1} that satisfies

$$g_j(u_k) + \sum_{i=1}^{n_u}\lambda_{i,j}|u_{k+1,i} - u_{k,i}| \leq 0. \tag{15}$$

Consider an RTO scheme that enforces (15) at each iteration. Then, as a plant constraint gets nearly active, i.e., $g_j(u_k) \to 0$, the step size for the next iteration becomes smaller and smaller. Eventually, $u_{k+1} - u_k \to 0$, which may result in very slow or even premature convergence to a suboptimal point. To avoid this, we use the following concepts that were introduced by Bunin et al. (2013).

Definition 1. (ϵ-active constraint). The constraint $g_j(u_k)$ is said to be ϵ_j-active at iteration k if, for $\epsilon_j > 0$, $-\epsilon_j \leq g_j(u_k) \leq 0$. The set of ϵ-active constraints is denoted by

$$\mathcal{J}_k = \{j \in \{1, \ldots, n_g\} : -\epsilon_j \leq g_j(u_k) \leq 0\}.$$

Definition 2. (δ-strict descent halfspace). The local strict descent halfspace of the constraint g_j at u_k is defined as the set $\{u \in \mathbb{R}^{n_u} : \nabla g_j(u_k)^T(u - u_k) < 0\}$. Furthermore, we define the δ_j-strict descent halfspace of the j^{th} constraint, with $\delta_j > 0$, as

$$\mathcal{D}_{j,k}^g = \left\{d \in \mathbb{R}^{n_u} : \nabla g_j(u_k)^T d \leq -\delta_j\right\}. \tag{16}$$

Note that the set representing the δ_j-strict descent halfspace is a subset of the strict descent halfspace of the same constraint.

The reason for the introduction of the ϵ-active constraints is to prevent getting too close to a constraint unless absolutely necessary. The δ_j-strict descent halfspace ensures that, once a constraint becomes ϵ_j-active, the iterates stay away from the constraint and avoid convergence to a suboptimal point.

4.1 RTO scheme

We propose an iterative RTO scheme that is based on the following assumptions:

Assumption 4. The initial input u_0 is strictly feasible with respect to the constraint functions g_j, i.e., $g_j(u_0) < 0$, $j = 1, ..., n_u$.

Assumption 5. The plant constraint functions g_j, $j = 1, ..., n_u$, are continuously differentiable on an open set containing the set $\mathcal{U} = \{u \in \mathbb{R}^{n_u} : u^L \preceq u \preceq u^U\}$.

Assumption 6. The exact values and gradients of the plant constraints g_j are available.

Assumption 7. The univariate Lipschitz constants $\lambda_{i,j}$ are available for the plant constraint functions.

The proposed iterative RTO scheme uses the following convex quadratic program

$$\min_{\Delta_{k+1}} \nabla\hat{\phi}_k^T\Delta_{k+1} + \frac{1}{2}\Delta_{k+1}^T\bar{Q}\Delta_{k+1} \tag{17a}$$

subject to

$$g_j(u_k) + \sum_{i=1}^{n_u}\lambda_{i,j}|\Delta_{k+1,i}| \leq 0, j = 1, ..., n_g \tag{17b}$$

$$\nabla g_j(u_k)^T\Delta_{k+1} \leq -\delta_j, \ \forall j \in \mathcal{J}_k \tag{17c}$$

$$u^L - u_k \preceq \Delta_{k+1} \preceq u^U - u_k, \tag{17d}$$

The next input update then becomes

$$u_{k+1} = u_k + \Delta_{k+1}^\star, \tag{18}$$

where $\Delta_{k+1}^\star$ denotes an optimal solution to (17). Note that $\delta_j, \epsilon_j > 0$ are user-defined tuning parameters. Also, constraint gradients are needed only when $\mathcal{J}_k \neq \emptyset$.

4.2 Feasibility of the RTO scheme

We analyze here the feasibility of the RTO scheme (17)-(18). We consider first the constraints (17b) and describe the feasible set for the input increment Δ_{k+1} with respect to the j^{th} constraint. By using the concept of Lipschitz constants introduced in (15), one can write

$$\mathcal{L}_{j,k} := \Big\{\Delta \in \mathbb{R}^{n_u} : \|\Lambda_j \Delta\|_1 \leq -g_j(u_k)\Big\}, \quad (19)$$

where $\|\cdot\|_1$ represents the ℓ_1-norm and $\Lambda_j = \text{diag}(\lambda_{i,j}) \in \mathbb{R}^{n_u \times n_u}$. Let $\mathcal{F}_k$ denote the feasible set of Problem (17) at u_k. This set can be written as

$$\mathcal{F}_k := \Big(\bigcap_{j=1}^{n_g} \mathcal{L}_{j,k}\Big) \cap \Big(\bigcap_{j\in\mathcal{J}_k} \mathcal{D}^g_{j,k}\Big) \cap \mathcal{U}_k. \quad (20)$$

Similarly, let $\mathcal{F} \subset \mathbb{R}^{n_u}$ be the feasible set of Problem (1). Using $\mathcal{L} := \Big\{u \in \mathbb{R}^{n_u} : g_j(u) \leq 0, j = 1, ..., n_g\Big\}$, we have

$$\mathcal{F} := \mathcal{U} \cap \mathcal{L}.$$

We present next three lemmas that will be used to prove recursive feasibility of the RTO scheme (17)–(18).

Lemma 4. If $u_k \notin int(\mathcal{L})$, then $\mathcal{F}_k = \emptyset$.

Proof. Consider first the case with $u_k \in \partial\mathcal{L}$. It follows that $g_j(u_k) = 0$ for at least one $j \in \{1, ..., n_g\}$. Hence, from (19), we get

$$\bigcap_{j=1}^{n_g} \mathcal{L}_{j,k} = 0^{n_u},$$

where 0^{n_u} is the zero vector of $\mathbb{R}^{n_u}$. It follows from $g_j(u_k) = 0$ that $\mathcal{J}_k \neq \emptyset$. From (16) we have $0^{n_u} \notin \mathcal{D}^g_{j,k}$, which gives

$$\Big(\bigcap_{j=1}^{n_g} \mathcal{L}_{j,k}\Big) \cap \Big(\bigcap_{j\in\mathcal{J}_k} \mathcal{D}^g_{j,k}\Big) = \emptyset,$$

and with (20), $\mathcal{F}_k = \emptyset$. Next, consider the case where $u_k \in (\mathbb{R}^{n_u} \setminus \mathcal{L})$, which implies that, for at least one $j \in \{1, ..., n_g\}$, $g_j(u_k) > 0$, and $\mathcal{L}_{j,k} = \emptyset$ follows from (19). Hence, $\mathcal{F}_k = \emptyset$. □

The above lemma shows that the necessary (but not sufficient) condition for (17) to be recursively feasible is that, for each iterate k, $u_k \in int(\mathcal{L})$. The next lemma helps in establishing the plant feasibility of the input update, which is found by solving (17)-(18).

Lemma 5. Consider (17) with Assumptions 5–7. Let $\mathcal{F}_k \neq \emptyset$ and the RTO iterate u_{k+1} be computed as

$$u_{k+1} = u_k + \Delta_{k+1}, \quad \Delta_{k+1} \in \mathcal{F}_k. \quad (21)$$

Then, $u_{k+1} \in \mathcal{F}$, i.e., u_{k+1} is a feasible input for the plant.

Proof. Since $\Delta_{k+1} \in \mathcal{F}_k$, then, $\forall j = 1, ..., n_u$,

$$g_j(u_k) + \sum_{i=1}^{n_u} \lambda_{i,j}|\Delta_{k+1,i}| \leq 0,$$

and, from (21) and (14),

$$g_j(u_{k+1}) \leq g_j(u_k) + \sum_{i=1}^{n_u} \lambda_{i,j}|u_{k+1,i} - u_{k,i}| \leq 0. \quad (22)$$

It follows from (17d) and (21) that $u^L \preceq u_{k+1} \preceq u^U$. Hence, $u_{k+1} \in \mathcal{F}$. □

The following lemma helps further develop the recursive feasibility of the RTO scheme.

Lemma 6. Consider Problem (17) with Assumptions 5–7. If $\mathcal{F}_k \neq \emptyset$ and the RTO iterate u_{k+1} be computed using (21). Then, $u_{k+1} \in int(\mathcal{L})$.

Proof. From (14) we get, for $j = 1, .., n_u$,

$$g_j(u_{k+1}) - g_j(u_k) \leq \sum_{i=1}^{n_u} \lambda_{i,j}|u_{k+1,i} - u_{k,i}|,$$

the right-hand side of which is always positive except for $u_{k+1} = u_k$. Therefore, from (22) and for $u_{k+1} \neq u_k$,

$$g_j(u_{k+1}) < 0 \quad or \quad u_{k+1} \in int(\mathcal{L}). \quad □$$

The Lemma 6 says that if the input update generated by the RTO scheme (17)-(18) is strictly feasible w.r.t. the constraints (1b) then, as discussed in Lemma 4, the necessary condition for the the feasibility of (17) is satisfied for that input update. The following proposition states the conditions for recursive feasibility of the RTO scheme.

Proposition 2. Let Assumptions 4–7 hold and the RTO scheme (17)-(18) be initialized at u_0. Then, the sequence $\{u_k\}_{k\geq 1}$ generated by the RTO scheme satisfies

$$u_k \in \mathcal{F} \cap int(\mathcal{L}), \forall k \in \mathbb{N}.$$

Proof. $u_0 \in \mathcal{F} \cap int(\mathcal{L})$ follows from Assumption 4. It follows from Lemma 5 and from Lemma 6 that
$u_k \in \mathcal{F} \cap int(\mathcal{L}), \forall k \in \mathbb{N}$. □

We have shown that, given strict feasibility of the initial input u_0, we can ensure plant feasibility for the subsequent inputs given by (18). Strict feasibility of the initial input also provides recursive feasibility of the RTO scheme (17)–(18). The tuning parameters δ_j and ϵ_j dictate the feasibility of the RTO scheme through (17c). The larger the values of these parameters, the smaller the feasible region. Hence, in concrete terms, one can choose smaller values of the tuning parameters to enlarge the feasible region, thereby ensuring that the RTO scheme (17)–(18) remains feasible. The price for that will be smaller steps and slower convergence.

4.3 Conditions for cost decreasing iterations

This section provides conditions under which plant cost decrease can be guaranteed at an iteration. To this end, we define the set

$$\mathcal{D}_{\mathcal{G}_k^-} = \Big\{d \in \mathbb{R}^{n_u} : \exists\beta > 0, \beta d \in \mathcal{G}_k^-\Big\}. \quad (23)$$

Note that $\mathcal{G}_k^- \subset \mathcal{D}_{\mathcal{G}_k^-} \subset \mathcal{D}_k$.

Proposition 3. Let the solution to (17) be such that $\Delta^\star_{k+1} \in \mathcal{D}_{\mathcal{G}_k^-}$. Then, the RTO scheme (17)–(18) will give $\phi(u_{k+1}) < \phi(u_k)$.

Proof. Consider the cost of Problem (17)

$$\phi_k := \nabla\hat{\phi}_k^T \Delta_{k+1} + \frac{1}{2}\Delta_{k+1}^T \bar{Q} \Delta_{k+1}.$$

The solution to (17) can be expressed as $\Delta^\star_{k+1} = \alpha^\star d^\star_k$, $\alpha^\star > 0$, $d^\star_k \in \mathbb{R}^{n_u}, \|d^\star_k\|_1 = 1$. The largest value that $\alpha^\star$ can take, $\alpha^\star_{max}$, can be found by minimizing $\phi_k(\alpha d^\star_k)$ over $\alpha > 0$,

$$0 < \alpha^\star \leq \alpha^\star_{max} = -\frac{\nabla\hat{\phi}_k^T d^\star_k}{(d^\star_k)^T \bar{Q} (d^\star_k)}.$$

It follows from $\Delta^\star_{k+1} \in \mathcal{D}_{\mathcal{G}_k^-}$ that $d^\star_k \in \mathcal{D}_{\mathcal{G}_k^-}$. Therefore, $\beta d^\star_k \in \mathcal{G}_k^-$ for some $\beta > 0$ and

$$d^\star_k = -\frac{1}{\beta}\hat{\Gamma}_k \nabla\phi_k, \quad \hat{\Gamma}_k \nabla\phi_k \in \mathcal{G}_k. \quad (24)$$

Since $\nabla\hat{\phi}_k \in \mathcal{G}_k$, $\alpha^\star_{max}$ can be written as

$$\alpha^\star_{max} = \frac{\beta(\Gamma_k \nabla\phi_k)^T(\hat{\Gamma}_k \nabla\phi_k)}{(\hat{\Gamma}_k \nabla\phi_k)^T \bar{Q}(\hat{\Gamma}_k \nabla\phi_k)}.$$

Note that $\hat{\Gamma}_k$ may be different from Γ_k. It follows from (7) that the maximum eigenvalue of Γ_k is $2-\varepsilon$ for some fixed scalar $\varepsilon > 0$. Hence, $\alpha^\star_{max}$ can be upper bounded as

$$\alpha^\star_{max} < \frac{2\beta(\nabla\phi_k)^T(\hat{\Gamma}_k \nabla\phi_k)}{(\hat{\Gamma}_k \nabla\phi_k)^T \bar{Q}(\hat{\Gamma}_k \nabla\phi_k)}. \tag{25}$$

For the plant cost, consider the step length $\alpha > 0$ in the direction $d^\star_k$. Next, we find the condition on α that guarantees a decrease in the plant cost. Applying Lemma 1 to the plant cost and with $\Delta_{k+1} = \alpha d^\star_k$, we get

$$\phi(u_{k+1}) - \phi(u_k) \le \alpha \nabla\phi_k^T d^\star_k + \frac{1}{2}\alpha^2 d^{\star T}_k \bar{Q} d^\star_k.$$

In the direction $d^\star_k$ given by (24), the condition on α for which the right-hand side of the above inequality is strictly negative is

$$\alpha < \frac{2\beta(\nabla\phi_k)^T(\hat{\Gamma}_k \nabla\phi_k)}{(\hat{\Gamma}_k \nabla\phi_k)^T \bar{Q}(\hat{\Gamma}_k \nabla\phi_k)}.$$

From (25), $\alpha^\star_{max}$ satisfies the above condition. Therefore, $\phi(u_{k+1}) < \phi(u_k)$. □

Proposition 3 implies that, if $\Delta^\star_{k+1}$ points in a direction $d \in \mathcal{D}_{\mathcal{G}^-_k} \subset \mathcal{D}_k$, the plant cost will decrease. Also note that, if the solution $\Delta^\star_{k+1} \in int(\mathcal{F}_k)$, then Proposition 1 guarantees that the plant cost decreases by applying the input update (18).

4.4 Choice of tuning parameters ϵ and δ

The tuning parameter ϵ_j defines the value of the corresponding plant constraint at which we start projecting the RTO iterates onto the constrained δ-strict descent half-space through (17c). We propose to start with a reasonably large values of ϵ_j and then decrease its value iteratively when (17) become infeasible for a particular value of ϵ_j. Similarly, we adapt δ_j. Also, once a feasible solution is obtained, we recommend setting the value of δ_j back to its initial value as the step length can get very small with small values of δ_j. In summary, we propose to solve the RTO problem (1) by iteratively solving the RTO scheme (17)–(18) using Algorithm 1.

Remark 1. The RTO scheme (17)–(18) is based on the exact same fundamentals, i.e., Lemmas 1 and 3, as the experimental optimization scheme proposed in Bunin et al. (2014). Moreover, to avoid premature convergence, the RTO scheme (17)–(18) uses the same concept of ϵ-activity that is used in Bunin et al. (2014). Yet, there is a fundamental difference between the two. The experimental optimization scheme is analogous to the exact line search method, while the RTO scheme (17)–(18) is analogous to the trust-region method. The scheme in Bunin et al. (2014) includes two steps, namely, projection followed by filtering. In the projection step, the method chooses the direction in which it will move next, while the filtering step selects the step length. Hence, the analogy.

Similar to trust-region methods, the feasible region defined by the constraints (17b)–(17d) is searched to find the best combination of direction and step length *simultaneously*.

Algorithm 1 Proposed RTO algorithm

DATA: u_0, $\lambda_{i,j}$, $\bar{Q}_\phi$, $\bar{\epsilon}$, $\bar{\delta}$, u^U, u^L, η_ϵ , η_δ.
INITIALIZE $k = 0$, $u_k = u_0$, $\epsilon_j = \bar{\epsilon}$, $\delta_j = \bar{\delta}$
STEP 1.
GET $g_j(u_k)$, $\nabla\hat{\phi}(u_k)$
for all j=1 to n_g **do**
  if $g_j(u_k) > -\epsilon_j$ **then** *GET* $\nabla g_j(u_k)$
  end if
end for
STEP 2.
while $\epsilon_j > \eta_\epsilon$ **do**
  while $\delta_j > \eta_\delta$ **do** *SOLVE* (17)
    if (17) = '$infeasible$' **then**
      for all j=1 to n_g **do** $\delta_j = \delta_j/2$
      end for
    else
      for all j=1 to n_g **do** $\delta_j = \bar{\delta}$
      end for, break
    end if
  end while
  if (17) = '$infeasible$' **then**
    for all j=1 to n_g **do** $\epsilon_j = \epsilon_j/2$
    end for
  else break
  end if
end while
if (17) = '$infeasible$' **then** $\Delta^\star_{k+1} = 0$
end if
APPLY $u_{k+1} = u_k + \Delta^\star_{k+1}$, $k \to k+1$.
GOTO STEP 1.

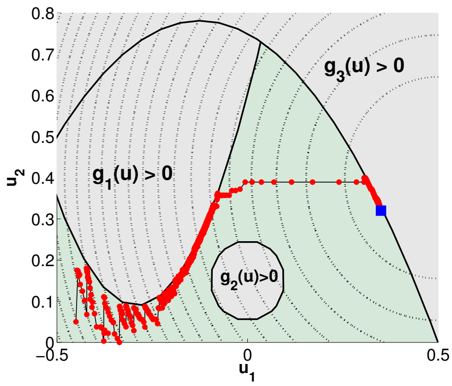

Fig. 2. One run of scheme (17)–(18) applied to (26).

4.5 Numerical Example

We introduce the following numerical example and apply Algorithm 1 to solve

$$\begin{aligned} \min_{u_1,u_2} \phi &= (u_1 - 0.5)^2 + (u_2 - 0.4)^2 \\ \text{subject to } g_1 &= -6u_1^2 - 3.5u_1 + u_2 - 0.6 \le 0 \\ g_2 &= 2u_1^2 + 0.5u_1 + u_2 - 0.75 \le 0 \\ g_3 &= -u_1^2 - (u_2 - 0.5)^2 + 0.01 \le 0 \\ &-0.5 \le u_1 \le 0.5,\ 0 \le u_2 \le 0.8 \end{aligned} \tag{26}$$

with the initial input $u_0 = (-0.45, 0.05)^T$. It is assumed that the constraint gradients are known exactly. Furthermore, the Lipschitz constants of the constraints and the quadratic upper bond on the cost are $\lambda_{1,1} = 10.45$,

$\lambda_{1,2} = \lambda_{2,2} = \lambda_{3,1} = 1.1$, $\lambda_{2,1} = 2.75$, $\lambda_{3,2} = 1.43$ and $\bar{Q} = diag(4.05, 4.05)$.

To account for uncertain cost gradient, we perform a Monte Carlo simulation consisting of 100 runs, each with a different cost gradient estimate. These estimates are generated randomly as follows. At each iteration k and for each gradient component i, a random number $\gamma_{k,i}$ is generated from a uniform distribution in the interval $(\varepsilon, 2-\varepsilon)$, with $\varepsilon = 0.002$. Then, each true gradient component is multiplied by the corresponding $\gamma_{k,i}$. Note that the estimates obtained in this manner satisfies Assumption 3.

For all three constraints, we choose the same initial values of the tuning parameters, $\bar{\epsilon} = 0.11$, $\bar{\delta} = 0.0002$. Algorithm 1 does not update the input when (17) becomes infeasible and the tuning parameters have reached their threshold values $\eta_\epsilon = 10^{-3}, \eta_\delta = 10^{-10}$. However, uncertainty in cost gradient causes the algorithm to keep searching in the vicinity of the plant optimum $u^\star = (0.35, 0.32)^T$. To prevent this from happening, the optimization is considered complete once the cost has reached the value 0.03025, which is 10% more than the optimal cost. One of the 100 runs is depicted in Figure 2. One can see that the RTO iterations, marked by red dots, converge to the plant optimum, plotted as a blue square. The contour curves for the plant cost are shown as dotted circles. Note that the RTO iterates traverse only through the strictly-feasible region highlighted in green. Table 1 indicates the number of RTO iterations needed to converge to the 10% neighbourhood of the true optimum, for the 3 runs with the minimum, median and maximum numbers of iterations.

Figure 3 depicts the evolution of the cost for these 3 runs. One also sees that the proposed RTO algorithm does converge to the 10% neighborhood of the minimal cost. Table 1 lists the number of gradient evaluations resulting from a constraint becoming ϵ-active. The constraint g_1 requires the most gradient evaluations. One sees that the constraint gradients are not evaluated at each iteration; for example, for the run R_{min}, gradient of g_1 is evaluated 86 times, that of g_2 8 times and that of g_3 56 times. Hence, it suffices to compute the gradients of the constraints only when they become ϵ-active. Note that this is a clear advantage over modifier adaptation, which requires all constraint gradients to be evaluated at each iteration, cf. Marchetti et al. (2009, 2010).

Table 1. Summary of the three RTO runs

Run	RTO Iterations	Gradient Evaluations for		
		g_1	g_2	g_3
R_{min}	221	86	8	56
R_{median}	537	406	18	4
R_{max}	649	511	27	3

5. CONCLUSION

This paper has presented a RTO scheme based on a surrogate model build around a quadratic upper bound on the plant cost. The proposed scheme ensures plant feasibility and provides conditions under which cost decreasing iterates are obtained despite the presence of uncertain gradients for the plant cost.

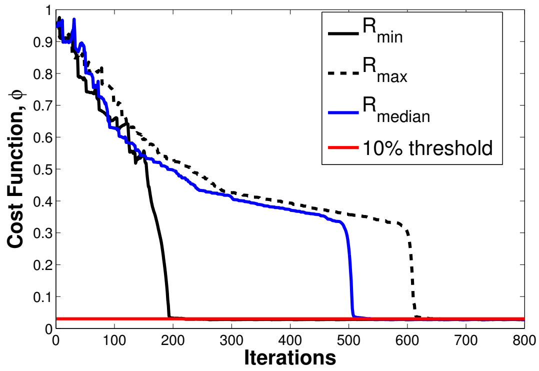

Fig. 3. Plant cost decrease for the three runs of Table 1.

REFERENCES

Bunin, G., François, G., and Bonvin, D. (2013). Sufficient conditions for feasibility and optimality of real-time optimization schemes - i. Theoretical foundations. ArXiv:1308.2620.

Bunin, G., François, G., and Bonvin, D. (2014). Feasible-side global convergence in experimental optimization. *SIAM J. Optimization (submitted)*. EPFL-ARTICLE-199753.

Chachuat, B., Srinivasan, B., and Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Comput. Chem. Engng*, 33(10), 1557–1567.

Gao, W. and Engell, S. (2005). Iterative set-point optimization of batch chromatography. *Comput. Chem. Engng*, 29(6), 1401–1409.

Jang, S.S., Joseph, B., and Mukai, H. (1987). On-line optimization of constrained multivariable chemical processes. *AIChE J.*, 33(1), 26–35.

Marchetti, A., Chachuat, B., and Bonvin, D. (2009). Modifier-adaptation methodology for real-time optimization. *I&EC Research*, 48(13), 6022–6033. doi: 10.1021/ie801352x.

Marchetti, A., Chachuat, B., and Bonvin, D. (2010). A dual modifier-adaptation approach for real-time optimization. *J. Process Control*, 20(9), 1027–1037.

Skogestad, S. (2000). Plantwide control: The search for the self-optimizing control structure. *J. Process Control*, 10(5), 487–507.

Srinivasan, B. and Bonvin, D. (2007). Real-time optimization of batch processes by tracking the necessary conditions of optimality. *I&EC Research*, 46(2), 492–504.