2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

Exploiting Local Quasiconvexity for Gradient Estimation in Modifier-Adaptation Schemes

Gene A. Bunin, Grégory François, Dominique Bonvin

***Abstract*— A new approach for gradient estimation in the context of real-time optimization under uncertainty is proposed in this paper. While this estimation problem is often a difficult one, it is shown that it can be simplified significantly if an assumption on the local quasiconvexity of the process is made and the resulting constraints on the gradient are exploited. To do this, the estimation problem is formulated as a constrained weighted least-squares problem with appropriate choice of the weights. Two numerical examples illustrate the effectiveness of the proposed method in converging to the true process optimum, even in the case of significant measurement noise.**

I. INTRODUCTION

In the steady-state, real-time optimization of continuous processes, it is often the case that the model used to carry out the on-line optimization is an inaccurate approximation of the true process. This makes the nominal solution unreliable, since it is either a solution that is suboptimal or, in the worst case, infeasible and likely to damage the process. While the standard approach for many years has been to iteratively improve on this solution by collecting measurements and updating the model parameters [1], there has been a recent trend to use measurements to add auxiliary "correction" terms to the model instead [2], [3], [4]. Based on the original work of Roberts [5], these algorithms do not necessarily update the process parameters but are designed around updating the correction terms so as to iteratively drive the optimum given by the modified model to a Karush-Kuhn-Tucker (KKT) point of the real process. These "modifier-adaptation" schemes are advantageous in that they are able to reject both structural and parametric errors in the model [4].

A major bottleneck of these approaches, however, is the need to estimate the real process derivatives (the local process gradient) at each operating point in the iterative optimization procedure. This is a difficult problem, since not only must the contributions of the different derivatives be decoupled at each point, but also because the measured outputs are often corrupted by noise. As basic approaches, one may fall back on finite-difference approximations or derivative-free (e.g. trust-region) methods [6], but these – despite being reliable – generally require an unacceptably large number of samples as the problem dimensionality grows. "Dual-modifier" methods have consequently emerged [2], [7], [8] to combat this issue, and are analogous to trust-region-with-linear-model approaches that require only n_u+1 operating points to estimate the gradient for a process with n_u inputs. Though promising, these methods require additional constraints to ensure robustness against truncation errors and noise, which often conflicts with the main task of optimization and results in algorithms that do not scale well with an increase in dimensionality.

In this work, a new framework that appears to avoid, or at least simplify, a lot of these issues is presented. Primarily, it is shown that assuming local process quasiconvexity leads to the introduction of new constraints on the gradient and may significantly simplify the estimation problem. In order to incorporate these constraints into the estimation, a formulation of the estimation problem as a weighted least-squares problem is proposed, and a possible choice of the weighing scheme is discussed. Two numerical examples serve to illustrate the proposed method, which appears to be quite effective even in the presence of high measurement noise.

The paper is structured as follows. First, the modifier-adaptation methodology is quickly reviewed in Section II. The additional constraints on the gradient that arise from assuming local convexity are detailed in Section III, and the estimation framework is outlined in Section IV. Section V then presents the numerical examples, and Section VI concludes the work.

II. REVIEW OF MODIFIER ADAPTATION

The steady-state, static optimization problem that needs to be solved is the following:[1]

$$\begin{aligned} \min_{\mathbf{u}} \; & \phi(\mathbf{u}) \\ \text{s.t.} \; & \mathbf{G}_p(\mathbf{u}) \preceq \mathbf{0} \end{aligned} \quad , \tag{1}$$

with $\mathbf{u} \in \mathbb{R}^{n_u}$ the vector of inputs, $\phi : \mathbb{R}^{n_u} \to \mathbb{R}$ the process objective, and $\mathbf{G}_p$ a set of n_g inequality constraints $g_{p,1} \cdots g_{p,n_g} : \mathbb{R}^{n_u} \to \mathbb{R}$.

Realistically, any or all of the $\mathbf{G}_p$ functions may not be known exactly in practice, and only an approximate model will be available:[2]

$$\begin{aligned} \min_{\mathbf{u}} \; & \phi(\mathbf{u}) \\ \text{s.t.} \; & \mathbf{G}(\mathbf{u}) \preceq \mathbf{0} \end{aligned} \quad . \tag{2}$$

The key idea of modifier adaptation is to add the minimum number of correction terms, or "modifiers", to force the KKT

Gene A. Bunin, Grégory François, and Dominique Bonvin are all with the Laboratoire d'Automatique of the Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne. gene.bunin@epfl.ch, gregory.francois@epfl.ch, dominique.bonvin@epfl.ch

[1]The symbol $\preceq$ denotes component-wise inequality.

[2]Without loss of generality, it may be assumed, via the epigraph transformation [9], that the cost is linear and known. It should also be noted that while the methodology extends to equality constraints, these have been left out for simplicity.

conditions of the modified model to match the KKT conditions of the process [4]. A scheme that partially achieves this goal is:

$$\begin{aligned} \min_{\mathbf{u}} \; & \phi(\mathbf{u}) \\ \text{s.t.} \; & \mathbf{G}(\mathbf{u}) + \boldsymbol{\varepsilon}_k \preceq \mathbf{0} \end{aligned} \quad , \tag{3}$$

where $\boldsymbol{\varepsilon}_k \in \mathbb{R}^{n_g \times 1}$ is a vector of 0^{th}-order correction terms at iteration k. In the simplest case, $\boldsymbol{\varepsilon}_k$ may be defined as the error between the process and the model:

$$\boldsymbol{\varepsilon}_k = \mathbf{G}_p(\mathbf{u}_k) - \mathbf{G}(\mathbf{u}_k). \tag{4}$$

When run iteratively, this "constraint-adaptation" scheme will, in many cases, converge to a feasible sub-optimal point for the real process [4]. It is attractive as it is easy to implement, does not require any gradient estimates, and has been demonstrated to be effective in practice [10]. When the sub-optimality of this scheme is negligible, it is an excellent first choice for the optimization of uncertain processes.

However, the degree of sub-optimality for a given process is generally unknown until it is observed. To truly guarantee that the process, upon convergence, reaches a KKT point, a 1^{st}-order modifier scheme is needed:

$$\begin{aligned} \min_{\mathbf{u}} \; & \phi(\mathbf{u}) \\ \text{s.t.} \; & \mathbf{G}(\mathbf{u}) + \boldsymbol{\varepsilon}_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k) \preceq \mathbf{0} \end{aligned} \quad , \tag{5}$$

where $\boldsymbol{\lambda}_k \in \mathbb{R}^{n_u \times n_g}$ are the 1^{st}-order modifiers at iteration k that essentially serve to correct the gradient errors between the process and the model and may, in the simplest case, be defined as the error in the Jacobian:

$$\boldsymbol{\lambda}_k^T = \frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}\bigg|_{\mathbf{u}_k} - \frac{\partial \mathbf{G}}{\partial \mathbf{u}}\bigg|_{\mathbf{u}_k}. \tag{6}$$

To ensure a certain degree of robustness, one would usually not apply the entire corrections in (4) and (6) as the terms are only local, and would filter the modifiers from iteration to iteration [4]. Alternatively, one could also filter the inputs [11] and apply only a fraction of the input steps calculated by (3) or (5).

When the process derivatives $\frac{\partial \mathbf{G}_p}{\partial \mathbf{u}}$ can be known accurately, it has been shown that the KKT conditions of the real process will be satisfied by the modifier scheme (5) upon convergence. For a proof of this statement, as well as for the local necessary conditions regarding convergence and a detailed overview of the scheme, the interested reader is referred to [4].

As already stated, the major weakness of modifier adaptation is that these experimental gradients are often not known and are difficult to estimate, making the resulting algorithms that rely on uncertain estimates unreliable. In the theory that follows, it is shown how assuming local quasiconvexity of the functions $\mathbf{G}_p(\mathbf{u})$ may alleviate these problems.

III. EXPLOITING LOCAL QUASICONVEXITY

As all of the analysis in this section applies individually to each of the constraints, a single function $g_p(\mathbf{u}) \in \mathbf{G}_p(\mathbf{u})$ will be considered throughout. Start by assuming that $g_p(\mathbf{u})$ is quasiconvex in some ball of radius δ around the current operating point $\mathbf{u}_k$, defined as $\mathcal{B}_k = \{\mathbf{u} : \|\mathbf{u} - \mathbf{u}_k\| \leq \delta\}$. Let $\mathcal{U}_k$ denote a set of points in $\mathcal{B}_k$ whose function values are inferior to $g_p(\mathbf{u}_k)$:

$$\mathcal{U}_k = \{\mathbf{u} : g_p(\mathbf{u}) \leq g_p(\mathbf{u}_k), \mathbf{u} \in \mathcal{B}_k\} \quad . \tag{7}$$

Then, the first-order condition for quasiconvexity states that:

$$\nabla g_p(\mathbf{u}_k)^T(\mathbf{u} - \mathbf{u}_k) \leq 0, \quad \forall \mathbf{u} \in \mathcal{U}_k \quad . \tag{8}$$

Likewise, if quasiconcavity were assumed instead and $\mathcal{B}_k$ denoted a quasiconcave region, the following would hold:

$$\nabla g_p(\mathbf{u}_k)^T(\mathbf{u} - \mathbf{u}_k) \geq 0, \quad \forall \mathbf{u} \in \mathcal{B}_k \setminus \text{int}(\mathcal{U}_k) \quad . \tag{9}$$

The derivation of (8) follows from applying Jensen's inequality to the convex sublevel sets $\mathcal{U}_k$ and differentiating, with the derivation of (9) following the same steps (see, e.g., [12]). Finally, if an assumption of quasilinearity is made, both sets of conditions are valid. For both simplicity and consistency, only quasiconvexity is explicitly discussed in what follows, but it remains implicit that everything that is mentioned in this paper is equally applicable to quasiconcavity and quasilinearity as well.

Practically, Condition (8) introduces linear bounds for estimating the gradient $\nabla g_p(\mathbf{u}_k)$ at iteration k. In other words, if there are previous measurements whose values are inferior to the current measured value $g_p(\mathbf{u}_k)$ and which are local enough to satisfy the quasiconvexity assumption, then (8) provides constraints on the gradient. A geometric interpretation is given in Fig. 1.

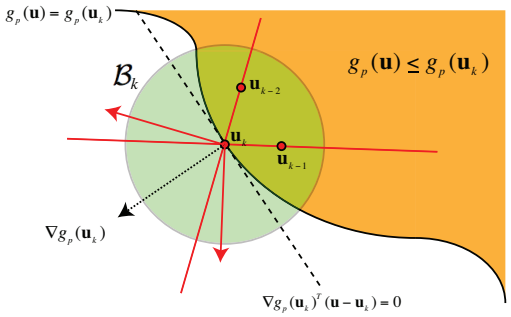

Fig. 1. Geometric interpretation of the first-order quasiconvexity condition constraints on the local gradient. Because $g_p(\mathbf{u}_{k-1})$ and $g_p(\mathbf{u}_{k-2})$ are both less than $g_p(\mathbf{u}_k)$ and are sampled in $\mathcal{B}_k$, the hyperplane defined around $\mathbf{u}_k$ must contain the two previous operating points on its negative side. The hyperplane is therefore constrained to be between the two extreme deviations shown in red solid lines. It follows that the gradient, which is orthogonal to the hyperplane, is constrained accordingly by the solid arrows.

As virtually all real processes are continuous, it is possible to extend the first-order quasiconvexity conditions to other points so as to achieve even tighter constraints by assuming Lipschitz constants on the components of $\nabla g_p(\mathbf{u})$. These are denoted by $\boldsymbol{\kappa} = [\kappa_1 \; \ldots \; \kappa_{n_u}]^T$ and are defined as follows:[3]

[3]The natural choice for these constants are the maximal bounds on the derivatives.

$\kappa_j \in [0, \infty)$:

$$g_p(\mathbf{u}_a) - g_p(\mathbf{u}_b) \leq \sum_{j=1}^{n_u} \kappa_j |u_{a,j} - u_{b,j}|, \quad \forall \mathbf{u}_a, \mathbf{u}_b \quad . \quad (10)$$

The following definition is now proposed.

Definition (The i,k-Lipschitz Polytope w.r.t. Function g_p) *The i,k-*Lipschitz polytope w.r.t. function g_p*, denoted by $\mathcal{L}_{i,k}^{g_p}$, is defined as the following set:*

$$\mathcal{L}_{i,k}^{g_p} = \{\mathbf{u} : g_p(\mathbf{u}_i) + \sum_{j=1}^{n_u} \kappa_j |u_j - u_{i,j}| \leq g_p(\mathbf{u}_k)\}. \quad (11)$$

Whenever $\mathbf{u}_i \in \text{int}(\mathcal{U}_k)$, this is a nonempty n_u-dimensional rhombus with $2n_u$ vertices that are easily obtained from $\mathbf{u}_i$, the (positive) difference $g_p(\mathbf{u}_k) - g_p(\mathbf{u}_i)$, and the Lipschitz constants $\boldsymbol{\kappa}$. Furthermore, it is clear that $\mathbf{u}_i \in \mathcal{L}_{i,k}^{g_p}$, and that $\{\mathcal{L}_{i,k}^{g_p} \cap \mathcal{B}_k\} \subseteq \mathcal{U}_k$. The implication of this is that Condition (8) must hold for all of $\mathcal{L}_{i,k}^{g_p} \cap \mathcal{B}_k$. Practically, this may be implemented by considering only the vertices of $\mathcal{L}_{i,k}^{g_p}$ that belong to $\mathcal{B}_k$.

A nice characteristic of this approach is that, regardless of how conservative the Lipschitz constants may be, the resulting conditions will *always* lead to tighter gradient estimates (by the simple virtue of having additional, non-redundant constraints). This is geometrically illustrated in Fig. 2.

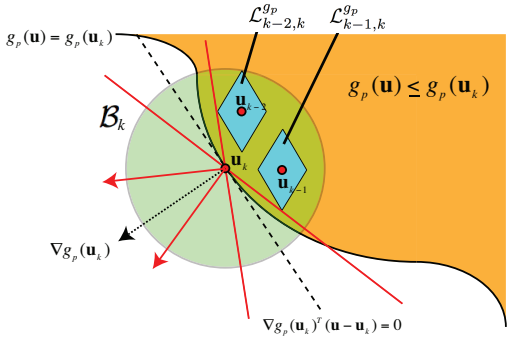

Fig. 2. The polytopes $\mathcal{L}_{k-1,k}^{g_p}$ and $\mathcal{L}_{k-2,k}^{g_p}$ around the past points $\mathbf{u}_{k-1}$ and $\mathbf{u}_{k-2}$ must also lie on the negative side of the hyperplane defined by the gradient at $\mathbf{u}_k$.

The overall potential of these concepts is clear: as the number of iterations increases and more points become available, the possible set of estimates may become so constrained that it becomes impossible to have a "bad" estimate. The estimation method that implements this idea is presented next.

IV. GRADIENT ESTIMATION METHOD

As the assumption of local quasiconvexity leads to linear constraints in the estimation problem, it is natural to place the estimation in the optimization framework where such constraints can be appropriately accommodated. Again, the procedure can be applied individually to each constraint, and so the method will be described for a single function $g_p(\mathbf{u}) \in \mathbf{G}_p(\mathbf{u})$.

A. Weighted least-squares problem

In the proposed method, the estimated gradient is computed as the solution to the following weighted least-squares problem, with $\mathbf{z}_k$ used to denote the gradient at the k^{th} iteration:

$$\begin{aligned} \min_{\mathbf{z}_k} . \quad & \sum_{i=1}^{k} w_i \left[\frac{\mathbf{z}_k^T(\mathbf{u}_i - \mathbf{u}_{i-1})}{\|\mathbf{u}_i - \mathbf{u}_{i-1}\|_2} - \frac{g_p(\mathbf{u}_i) - g_p(\mathbf{u}_{i-1})}{\|\mathbf{u}_i - \mathbf{u}_{i-1}\|_2} \right]^2 \\ \text{s.t.} \quad & \mathbf{z}_k^T(\mathbf{u} - \mathbf{u}_k) < 0, \\ & \forall \mathbf{u} \in \{\mathcal{L}_{0,k}^{g_p} \cup ... \cup \mathcal{L}_{k-1,k}^{g_p}\} \cap \mathcal{B}_k \\ & -\boldsymbol{\kappa} \preceq \mathbf{z}_k \preceq \boldsymbol{\kappa} \end{aligned} \quad . \quad (12)$$

The first set of constraints are the first-order quasiconvexity conditions applied to the union of Lipschitz polytopes that correspond to the previous data and lies in $\mathcal{B}_k$. Note that these inequality constraints are made *strict* so as to avoid the degenerate solution $\mathbf{0}$ (in implementation, it is a non-strict inequality with a small tolerance). There are also box constraints on the gradient estimates that are simply the Lipschitz constants, which, while generally not defining the solution, are nevertheless useful as they ensure that the estimates are always bounded.

As in dual-modifier schemes, the idea is to estimate the current gradient from data collected in the previous operating points, and this is reflected in the objective. The gradient $\mathbf{z}_k$ estimated at iteration k is run through all previous data – from iteration 1 to the current iteration k – to predict the increment $\mathbf{z}_k^T(\mathbf{u}_i - \mathbf{u}_{i-1})$ that is compared to the measured increment $g_p(\mathbf{u}_i) - g_p(\mathbf{u}_{i-1})$. The step size between successive iterations in the denominator is used as it provides an important scaling effect (otherwise, iterations with larger step sizes tend to dominate the cost and thus bias the estimation – see, e.g., [13]).

An important difference with dual-modifier approaches is that the solution to this problem may be obtained regardless of the conditioning of the step-size matrix or the effect of the noise. It is therefore not necessary to add constraints to the optimization problem to make it work. However, some quality control is needed to give more weight to iterations with useful information concerning the *local* gradient and less weight to iterations that are either heavily corrupted by noise or are too far away from the current operating point to offer any benefit. The weights w_i are designed with these goals in mind.

B. Choice of weights

As is often the case, designing the "optimal" weighting scheme is an art that is subjective and requires some experience. Nevertheless, some reasonable guidelines for choosing such a scheme *a priori* are given here. This choice is based on the following important criteria: 1) iterations where the noise is likely to overwhelm the measurement should be given very low weight, 2) iterations that are "far" from the current operating point should be given low weight as they

have little relation to the local gradient, 3) on the contrary, iterations that are close to the current point and are expected to have a good signal-to-noise ratio should be given high weight. Each weight is composed of two terms:

$$w_i = w_{s,i}\, w_{d,i} \quad , \tag{13}$$

with $w_{s,i}$ reflecting the signal-to-noise ratio and $w_{d,i}$ reflecting the distance from the current operating point. Both are defined as functions taking values between 0 and 1, which essentially guarantees that weight is distributed based on the specifications given above.

In designing $w_{s,i}$, it is necessary to have some sort of expected signal-to-noise ratio for a given step. Several heuristic approaches could be proposed to estimate the signal. In this paper, this estimation is done crudely using the Lipschitz constants introduced earlier:

$$|g_p(\mathbf{u}_i) - g_p(\mathbf{u}_{i-1})| \approx \sum_{j=1}^{n_u} \kappa_j |u_{i,j} - u_{i-1,j}| \quad . \tag{14}$$

For the noise, the worst-case noise magnitude σ is assumed. The resulting signal-to-noise ratio r_i is thus a "best-signal-worst-noise" approximation:

$$r_i = \frac{\sum_{j=1}^{n_u} \kappa_j |u_{i,j} - u_{i-1,j}|}{\sigma} \quad . \tag{15}$$

The weight is then defined as:

$$w_{s,i} = \min(0.5 r_i, 1) \quad , \tag{16}$$

i.e. it is a linear function of the estimated signal-to-noise r_i for $0 \leq r_i < 2$, and then constant at 1 for $r_i \geq 2$.

For designing $w_{d,i}$, the distance between the current point $\mathbf{u}_k$ and the points involved in the i^{th} iteration is first defined:

$$d_i = \|0.5(\mathbf{u}_{i-1} + \mathbf{u}_i) - \mathbf{u}_k\|_2 \quad , \tag{17}$$

which is the Euclidean distance between the current point and the midpoint in the i^{th} iteration. $w_{d,i}$ is then defined as:

$$w_{d,i} = 1 - \frac{d_i}{\max\limits_{q=1\ldots k}(d_q)} \quad , \tag{18}$$

i.e. the distance is scaled with respect to the maximal distance.

If the linear penalization is not strict enough (for either $w_{d,i}$ and $w_{s,i}$), the weights could be raised to a power (e.g. $w_i = w_{s,i}^2 w_{d,i}^2$) for stricter penalties. The tradeoff here lies between quantity and quality – whether it is better to have a lot of partially irrelevant data or a smaller amount of relevant data – and is not addressed in this paper.

C. Effect of noise on the constraints

When noise with the worst-case magnitude σ is present, it may not be clear which previous operating points the constraints should be valid for, since it may not be possible to tell which previous constraint measurements are in fact inferior to the current one. In this paper, the worst-case scenario is always assumed, i.e. $\mathcal{U}_k$ is redefined as:

$$\mathcal{U}_k = \{\mathbf{u} : g_p(\mathbf{u}) + \sigma \leq g_p(\mathbf{u}_k) - \sigma, \mathbf{u} \in \mathcal{B}_k\} \quad . \tag{19}$$

The Lipschitz polytope definition is also modified accordingly:

$$\mathcal{L}_{i,k}^{g_p} = \{\mathbf{u} : g_p(\mathbf{u}_i) + \sigma + \sum_{j=1}^{n_u} \kappa_j |u_j - u_{i,j}| \leq g_p(\mathbf{u}_k) - \sigma\} \quad , \tag{20}$$

which guarantees that the resulting linear constraints will still be valid in the noisy case.

V. ILLUSTRATIVE EXAMPLES

Two examples illustrate the effectiveness of the proposed gradient estimation method for modifier-adaptation schemes. The first example has two inputs and serves to illustrate the real-time optimization scheme when the results can be interpreted geometrically. The second example is more involved and encompasses six inputs. Both processes are globally convex, and thus the local quasiconvexity assumption holds by default with $\mathcal{B}_k = \mathbb{R}^{n_u}$ for all k.

A. 2-input case

Consider the following problem:

$$\begin{aligned} \max_{\mathbf{u}} \; & u_1 + u_2 \\ \text{s.t.} \; & g_p(\mathbf{u}) = (u_1 - 2.5)^4 + 1.6u_2 - 3.6 \leq 0 \\ & 1 \leq u_1 \leq 3.5 \\ & 1 \leq u_2 \leq 2.5 \end{aligned} \quad . \tag{21}$$

The nonlinear constraint used in the optimization problem is

$$g(\mathbf{u}) = (u_1 - 2)^4 + 2u_2 - 4 \leq 0 \quad . \tag{22}$$

The process is initialized at $\mathbf{u} = [2 \;\; 1.5]^T$ and the modifier-adaptation scheme described in Section 2 is launched. The inputs are filtered with the gain 0.5 to prevent the algorithm from taking overly aggressive steps, that is, at each iteration the solution of the modified problem is averaged with the previous operating point. The Lipschitz vector $\boldsymbol{\kappa} = [13.5 \;\; 1.6]^T$ is chosen, corresponding to the largest derivative values over the relevant input space. In practice, these would not be known and so more conservative values would be chosen.

Fig. 3 shows the optimization results for the cases with $\sigma = 0, 0.05, 0.10$, and 0.20 (since the value of $g_p(\mathbf{u})$ ranges between -1 and 0 in this example, this may be roughly thought of as 0, 5, 10, and 20% measurement noise). A large number of iterations is simulated for all cases so as to show that not only does the algorithm converge quickly in, generally, less than 20 iterations, but that it is able to maintain the optimum upon convergence.

As expected, the performance declines with rising noise, but the algorithm remains quite robust (only in the 20% case is there a noticeable offset). Since noise not only corrupts the estimation but also reduces the number of useful constraints that can be added (note the reduced size of the polytopes for cases with large σ), it is natural that offsets begin to occur as the noise grows large.

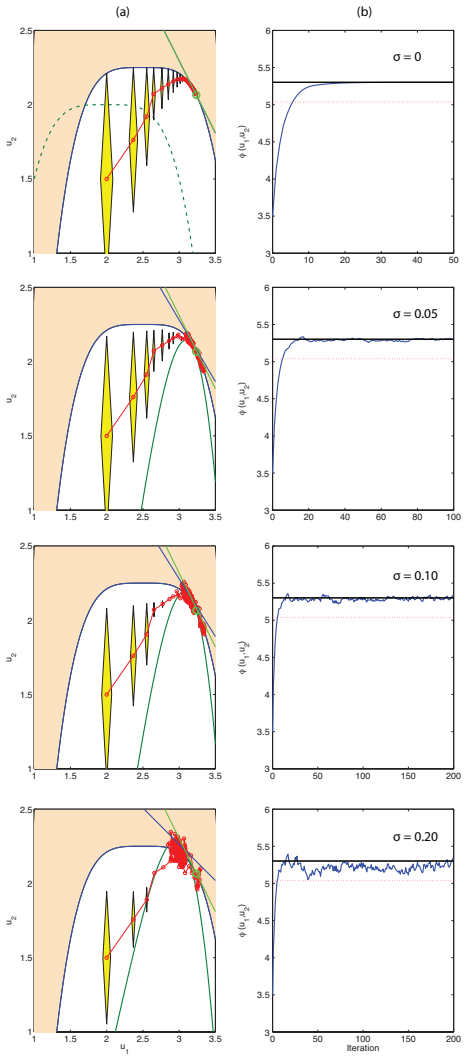

Fig. 3. Solution of Problem (21) using modifier adaptation with the proposed gradient estimation approach for cases with different noise levels. a) The green and blue curves represent the contours of $g(\mathbf{u})+\varepsilon_k+\boldsymbol{\lambda}_k^T(\mathbf{u}-\mathbf{u}_k)$ and $g_p(\mathbf{u})$, respectively (note that for the case of $\sigma = 0$, the green, dashed contour is given for $k = 0$ with $\varepsilon_0 = 0$ and $\boldsymbol{\lambda}_0 = \mathbf{0}$ so as to represent $g(\mathbf{u})$ and illustrate the initial process-model mismatch; otherwise, modifiers corresponding to the final iteration are used). Straight lines of the respective colors show the supporting hyperplanes of the modified-model and process functions upon convergence. The Lipschitz polytopes are also shown, and can be seen to all lie on the correct side of the hyperplane. The non-shaded region represents the feasible space and the green circle represents the process optimum. b) The black line at the very top gives the objective function value at the true process optimum, while the red dashed line below gives the objective value attained if the gradients were not estimated and only constraint adaptation (3) were used.

In order to illustrate the benefits brought by the quasi-convexity assumption, results for a case where no additional constraints (8) are included are given in Fig. 4. It is seen that this version of the algorithm, despite finding the optimal region, may drift very easily from the desired optimum (it does, however, work perfectly well in the absence of noise).

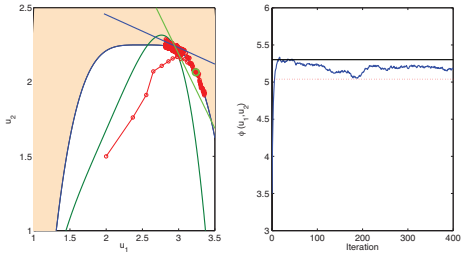

Fig. 4. Solution of Problem (21) for $\sigma = 0.05$ when local quasiconvexity is not assumed.

As a final test, the case with $\sigma = 0.10$ and a major input perturbation is considered ($\mathbf{u}$ is set to $[1.5\ 2]^T$ every 500 iterations). It is seen that the algorithm brings the inputs back to the correct region and maintains them there (Fig. 5).

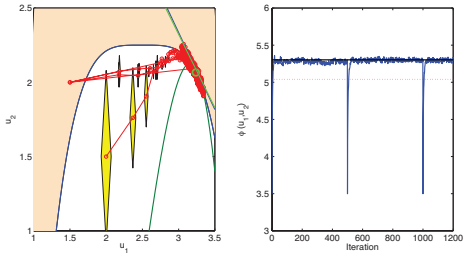

Fig. 5. Solution of Problem (21) with $\sigma = 0.10$ and a major input perturbation applied every 500 iterations.

B. 6-input case

The following optimization problem is solved:

$$\begin{aligned} \max_{\mathbf{u}} \quad & \mathbf{c}^T\mathbf{u} \\ \text{s.t.} \quad & g_{p,1}(\mathbf{u}) = 2u_1^4 + e^{2u_3} + u_5^2 - u_6 - .8 \le 0 \\ & g_{p,2}(\mathbf{u}) = u_2^2 - 2u_3 + u_4^2 + u_5 - 2 \le 0 \\ & g_{p,3}(\mathbf{u}) = -u_1 - u_2 + .5e^{u_5} + u_6^2 - 2 \le 0 \\ & g_{p,4}(\mathbf{u}) = -u_3 + u_4^2 + 6u_6^2 - 4 \le 0 \\ & 0 \preceq \mathbf{u} \preceq 1 \end{aligned} \quad , \tag{23}$$

where $\mathbf{c} = [0.45\ 0.04\ 0.88\ 0.69\ 0.95\ 0.56]^T$ and the modeled constraints are:

$$\begin{aligned} g_1(\mathbf{u}) &= 2.2u_1^2 + e^{3u_3} + 0.9u_5^2 - 0.9u_6 - 0.8 \\ g_2(\mathbf{u}) &= 1.1u_2^2 - 1.1u_3 + 0.9u_4^2 - u_4 - 2 \\ g_3(\mathbf{u}) &= -2u_2 + e^{1.5u_5} + 0.8u_6^2 - 2 \\ g_4(\mathbf{u}) &= -1.2u_3 + u_4 + 2.5u_6^2 - 4 \end{aligned} \quad . \tag{24}$$

The following matrix of Lipschitz constants, where each row contains the Lipschitz constants of the corresponding constraint, is used:

$$\mathrm{K} = \begin{bmatrix} 8 & \mu & 2e^2 & \mu & 2 & 1 \\ \mu & 2 & 2 & 2 & 1 & \mu \\ 1 & 1 & \mu & \mu & 0.5e & 2 \\ \mu & \mu & 1 & 2 & \mu & 12 \end{bmatrix}, \qquad (25)$$

where μ is a small value (here, it is set to 0.001) that is meant to replace 0 so as not to make the polytopes degenerate. This basically means that that particular input is known to have no effect on a particular constraint value. An input filter with gain 0.5 is used, and the initial point $\mathbf{u} = [0\ 0\ 0\ 0\ 0\ 0.8]^T$ is chosen.

The algorithm works well for $\sigma = 0$ and $\sigma = 0.05$, and only the results for $\sigma = 0.10$ are given here. Fig. 6 shows that, despite undergoing an initial "roughness" (likely attributable to aggressive steps by the optimization algorithm), the algorithm finds the right region after about 50 iterations and then suffers a small bump before settling down for good despite the continued injection of noise.

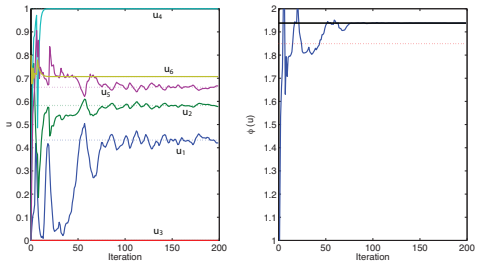

Fig. 6. Solution of Problem (23) for $\sigma = 0.10$. In the left plot, the dotted lines represent the optimal inputs for the process.

VI. CONCLUSIONS AND FUTURE WORK

A new methodology for estimating the local gradients in modifier-adaptation schemes has been presented, making the assumption of local quasiconvexity and showing how this facilitates what is otherwise a rather difficult problem. Through two numerical examples, it has been demonstrated that the stated methodology is effective at finding the optimal region and, with the additional hyperplane constraints on the gradients, keeping the system there even in the presence of significant noise. Particularly promising is the extension to the six-dimensional case, as most current gradient estimation schemes do not generally extend well to higher dimensions.

The potential weakness in the current work is, of course, the assumption that the real process is locally quasiconvex (as mentioned, one could also assume quasiconcavity, or even quasilinearity). Two difficulties arise here – the choice of assumption and the locality of neighborhood. In reality, nothing prevents the user from trying different assumptions and using the one that, over a number of iterations, appears to give the best estimates. In considering the locality of any of the assumptions, it is important to note that, save for pathological cases like the saddle point, every function must be locally quasiconvex or quasiconcave (or both, if monotonic). Just how small the neighborhood would be in practice is, naturally, unknown, but it may be argued that reasonably-sized neighborhoods would exist for real processes, and that the locality of the assumption would not be so hindering as to limit its applicability. Future work will, however, focus on ways to verify these assumptions on-line and to attempt to estimate the size of the neighborhood adaptively.

It should also be noted that all of the noise handling in this work was done in a crude, worst-case manner. This could probably be done stochastically, thereby allowing points that would normally be excluded from the set of constraints to still play a productive role in their construction, thereby allowing tighter estimates even with high noise levels. Other elements, such as the choice of weights in the gradient-estimation scheme, or ensuring that the process constraints are satisfied at every iteration [11], could also be improved upon.

Finally, it should be noted that, while this estimation scheme was applied here in the context of modifier adaptation, the theory is by no means limited to this context and could perhaps find use in other fields where gradient estimation is important [13], [14], [15].

REFERENCES

[1] C. Y. Chen and B. Joseph, On-line optimization using a two-phase approach: an application study, *Ind. Eng. Chem. Res.*, vol. 26, 1987, pp 1924-1930.

[2] W. Gao and S. Engell, Iterative set-point optimization of batch chromatography, *Computers and Chemical Engineering*, vol. 29, 2005, pp 1401-1409.

[3] B. Chachuat, A. Marchetti, and D. Bonvin, Process optimization via constraints adaptation, *J. Process Contr.*, vol. 18, 2008, pp 244-257.

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

[5] P. D. Roberts, Algorithms for integrated system pptimisation and parameter estimation, *Electronics Letters*, vol. 14(6), 1978, pp 196-197.

[6] A. R. Conn, K. Scheinberg, and L. N. Vicente, *Introduction to Derivative-Free Optimization*, Cambridge University Press, Cambridge, UK; 2009.

[7] A. Marchetti, B. Chachuat, and D. Bonvin, A dual modifier-adaptation approach for real-time optimization, *J. Process Contr.*, vol. 20(9), 2010, pp 1027-1037.

[8] W. Tadej and P. Tatjewski, Analysis of an ISOPE-type dual algorithm for optimizing control and nonlinear optimization, *Int. J. Appl. Math. Comput. Sci.*, vol. 11(2), 2001, pp 429-457.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK; 2008.

[10] G. A. Bunin, Z. Wuillemin, G. François, A. Nakajo, L. Tsikonis, and D. Bonvin, Experimental real-time optimization of a solid oxide fuel cell stack via constraint adaptation, *Energy*, vol. 39, 2012, pp 54-62.

[11] G. A. Bunin, G. François, B. Srinivasan, and D. Bonvin, Input filter design for feasibility in constraint-adaptation schemes, *in 18th IFAC World Congress*, Milan, Italy, 2011, pp. 5585-5590.

[12] K. J. Arrow and A. C. Enthoven, Quasi-concave programming, *Econometrica*, vol. 29(4), 1961, pp 779-800.

[13] C. D. Correa, R. Hero, and K.-L. Ma, A comparison of gradient estimation techniques for volume rendering of unstructured meshes, *IEEE Transactions on Visualization and Computer Graphics*, vol. 17(3), 2011, pp 305-319.

[14] T. H. Meyer, M. Eriksson, and R.C. Maggio, Gradient estimation from irregularly spaced data sets, *Mathematical Geology*, vol. 33(6), 2001, pp 693-717.

[15] Y. L. Yeow, J. Isac, F. A. Khalid, Y.-K. Leong, and A.S. Lubansky, A method for computing the partial derivatives of experimental data, *AIChE Journal*, vol. 56(12), 2010, pp 3212-3224.