On the Equivalence Between the Modifier-Adaptation and Trust-Region Frameworks

Gene A. Bunin

Abstract

In this short note, the recently popular modifier-adaptation framework for real-time optimization is discussed in tandem with the well-developed trust-region framework of numerical optimization, and it is shown that the basic version of the former is simply a special case of the latter. This relation is then exploited to propose a globally convergent modifier-adaptation algorithm using already developed trust-region theory. Cases when the two are not equivalent are also discussed.

Keywords: modifier adaptation, trust-region methods, real-time optimization

1. The Real-Time Optimization Problem

In the process systems engineering community, the basic idea of most real-time optimization (RTO) schemes consists in finding a set of optimal operating conditions – often steady-state setpoints in a multilayer hierarchical scheme – that minimize (resp., maximize) the steady-state cost (resp., profit) of some given plant (Brdys & Tatjewski, 2005). While models of the process being optimized are often available, it is generally the case that they are either inaccurate and/or incomplete, which motivates the data-driven "real-time" element of RTO, thereby forcing the optimization algorithm to use the measurements obtained from the process as feedback to modify the provided setpoints so as to ultimately reject the model uncertainty and converge to the optimal conditions of the plant. For simplicity, let us focus on the unconstrained problem, which may be formulated mathematically as:

$$\underset{\mathbf{u}}{\text{minimize}} \quad \phi_p(\mathbf{u}) \quad , \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the decision variables (or the "inputs") of the problem, while the function $\phi_p : \mathbb{R}^{n_u} \to \mathbb{R}$ denotes the cost. The subscript p (for "plant") is used to indicate that the function corresponds to an experimental relationship that is not perfectly known and may only be approximated by a model, which we will mark with the subscript $\hat{p}$ (i.e., $\phi_{\hat{p}}$). In the simplest terms, the goal of

Email address: gene.a.bunin@ccapprox.info (Gene A. Bunin)

an RTO algorithm is to solve Problem (1) by iterative experimentation – i.e., by generating a sequence of steady-state $\mathbf{u}$ values that converges to the plant optimum.

2. Review of the Modifier-Adaptation Framework

An approach to solving (1) that has recently gained popularity in the research community is that of *modifier adaptation*, which originally dates back to the work of Roberts (1978) and owes its numerous refinements and fundamental ideas to the *ISOPE* ("iterative setpoint optimization and parameter estimation") framework (Brdys & Tatjewski, 2005). Recent works by Gao & Engell (2005) and Marchetti et al. (2009) have given the approach its current "modern" form (by accounting for plant-model mismatch in both the cost and constraints), with the name "modifier-adaption" due to the latter. The method has also been picked up by a number of researchers in the last few years, with a number of works considering different fundamental aspects of the framework, the majority of them focusing on mathematical reformulations that may ease or better accomodate particular problem types (Costello et al., 2013; François & Bonvin, 2013; Serralunga et al., 2013). Other works have looked at important implementation aspects (Bunin et al., 2012; Marchetti et al., 2010; Rodger, 2010), while a few have attempted to tackle major theoretical issues like feasibility (Bunin et al., 2011; Navia et al., 2012) and global convergence (Chachuat et al., 2008).

The basic philosophy of modifier adaptation lies in applying local corrections to an inherently incorrect model at each RTO iteration k, and solving this corrected version to obtain the following iterate at $k+1$:

$$\mathbf{u}_{k+1} := \arg \underset{\mathbf{u}}{\text{minimize}} \quad \phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k) \ , \tag{2}$$

with the *modifiers* ε_k and $\boldsymbol{\lambda}_k$ defined as:

$$\varepsilon_k := \phi_p(\mathbf{u}_k) - \phi_{\hat{p}}(\mathbf{u}_k), \tag{3}$$

$$\boldsymbol{\lambda}_k := \nabla\phi_p(\mathbf{u}_k) - \nabla\phi_{\hat{p}}(\mathbf{u}_k). \tag{4}$$

Placing this into algorithmic form yields the following basic implementation.

Algorithm 1 (Basic Modifier-Adaptation Algorithm)

1. (Initialization) The initial point, $\mathbf{u}_0$, is provided. Set $k := 0$.
2. (Modifier Computation) Compute the modifiers ε_k and $\boldsymbol{\lambda}_k$ according to (3) and (4).
3. (New Input Calculation) Obtain $\mathbf{u}_{k+1}$ by solving the "modified" problem (2) and apply this to the plant.
4. (Iterate) Set $k := k+1$ and return to Step 2.

The key oft-stated motivation for applying this algorithm is the following upon-convergence guarantee.

Theorem 1 (First-Order Critical Point Upon Convergence). *Assume that the minimization of (2) always yields a first-order critical point of the modified objective function* $\phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k)$ *and that Algorithm 1 has converged to a fixed point* $\mathbf{u}_\infty$*. It follows that* $\mathbf{u}_\infty$ *is a first-order critical point of* ϕ_p*.*

PROOF. The result follows immediately from the fact that a first-order critical point is defined entirely by the values of the function and its derivatives at that point. As these must match for the modified model and the plant at any iterate (including $\mathbf{u}_\infty$), it follows that finding a first-order critical point for the modified function is identical to finding one for the plant. See, for example, Marchetti et al. (2009) for a slightly more detailed proof. □

3. The Basic Trust-Region Algorithm

A somewhat well-established approach for iteratively minimizing a nonlinear function in the mathematical optimization context is that of trust-region methods. In this section, we will consider what attempting to solve Problem (1) within this framework would entail.

Let us start by stating the basic trust-region algorithm for solving (1). This is essentially the algorithm provided in the well-known monograph on trust-region methods (Conn et al., 2000, Ch. 6) but with a few additional simplifications and some notational changes.

Algorithm 2 (Basic Trust-Region Algorithm)

1. (Initialization) The initial point, $\mathbf{u}_0$, and initial trust-region radius, $\Delta_0 > 0$, are provided, together with the constants η_1, η_2, γ_1, and γ_2 satisfying

$$0 < \eta_1 \leq \eta_2 < 1, \tag{5}$$

$$0 < \gamma_1 \leq \gamma_2 < 1. \tag{6}$$

   Set $k := 0$ and $\mathbf{u}_0^* := \mathbf{u}_0$.

2. (Model Construction) Construct the model m_k, which is an approximation of ϕ_p over the trust region $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ (i.e., a Euclidean ball of radius Δ_k centered at $\mathbf{u}_k^*$).

3. (New Input Candidate Calculation) Compute a candidate point $\mathbf{u}_{k+1} \in \mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ that "sufficiently reduces the model" m_k.

4. (Acceptance of the Candidate Point) Apply $\mathbf{u}_{k+1}$ to the plant and evaluate $\phi_p(\mathbf{u}_{k+1})$. Define:

$$\rho_k := \frac{\phi_p(\mathbf{u}_k^*) - \phi_p(\mathbf{u}_{k+1})}{m_k(\mathbf{u}_k^*) - m_k(\mathbf{u}_{k+1})}. \tag{7}$$

If $\rho_k \geq \eta_1$, then set $\mathbf{u}_{k+1}^* := \mathbf{u}_{k+1}$. Otherwise, set $\mathbf{u}_{k+1}^* := \mathbf{u}_k^*$.

5. (Trust-Region Radius Update) Set Δ_{k+1} such that

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \geq \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \tag{8}$$

6. (Iterate) Set $k := k + 1$ and return to Step 2.

Let us now review the assumptions necessary to prove the global convergence of Algorithm 2 to a first-order critical point (Conn et al., 2000). The following are assumed about the nature of the plant:

Assumption 1. *ϕ_p is twice continuously differentiable on $\mathbb{R}^{n_u}$.*

Assumption 2. *ϕ_p is lower-bounded on $\mathbb{R}^{n_u}$.*

Assumption 3. *The Hessian of ϕ_p is upper-bounded on $\mathbb{R}^{n_u}$.*

The following assumptions are made on the model:

Assumption 4. *For all k, m_k is twice differentiable over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$.*

Assumption 5. *For all k, the Hessian of m_k over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ is upper-bounded.*

Assumption 6. *m_k matches ϕ_p locally to first order at every k, i.e.:*

$$m_k(\mathbf{u}_k^*) = \phi_p(\mathbf{u}_k^*), \tag{9}$$

$$\nabla m_k(\mathbf{u}_k^*) = \nabla \phi_p(\mathbf{u}_k^*). \tag{10}$$

Finally, one requires the following assumption on the algorithm regarding its ability to achieve "sufficient reduction" in the model:

Assumption 7. *There exists a constant $\kappa \in (0, 1)$ such that for all k:*

$$m_k(\mathbf{u}_k^*) - m_k(\mathbf{u}_{k+1}) \geq \kappa \|\nabla m_k(\mathbf{u}_k^*)\| \min\left[\frac{\|\nabla m_k(\mathbf{u}_k^*)\|}{\beta_k}, \Delta_k\right], \tag{11}$$

with $\beta_k > 1$ a finite constant.

One may then state the following.

Theorem 2 (Global Convergence to a First-Order Critical Point). *If Assumptions 1-7 are satisfied, it then follows that the iterates generated by Algorithm 2 converge asymptotically to a first-order critical point, i.e.:*

$$\lim_{k\to\infty} \|\nabla\phi_p(\mathbf{u}_k^*)\| = 0. \tag{12}$$

PROOF. The reader is referred to Theorem 6.4.6 in Conn et al. (2000). □

4. Equivalence and a Globally Convergent Modifier-Adaptation Scheme

It is not difficult to see that Algorithm 2 is a generalization of Algorithm 1, as setting $m_k(\mathbf{u}) := \phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k)$, simplifying Step 4 to $\mathbf{u}_{k+1}^* := \mathbf{u}_{k+1}$, and letting $\Delta_0 \to \infty$ and $\gamma_1 \to 1$ (i.e., removing the trust-region constraint) essentially establishes equivalence. Another blatant similarity is that the local matching required by Assumption 6 (and usually enforced in any standard trust-region algorithm) is satisfied by construction in modifier adaptation – for more on this point, see also the recent work of Biegler et al. (2014) and the references therein.

Apart from building a bridge between two frameworks that have traditionally been isolated from one another[1], a major benefit of establishing this equivalence is that we may easily use the theory developed for trust-region methods to propose a globally convergent modifier-adaptation algorithm, which is something that has been largely missing – Theorem 4.1 in Brdys & Tatjewski (2005), the unpublished (and unfinished) report by Chachuat et al. (2008), and the meta-conditions (that are valid for any RTO algorithm) of Bunin et al. (2013) being the only exceptions. Note that only three additions to Algorithm 1 are needed to make it globally convergent:

- a trust region to regulate the locality of the model,
- the testing of candidate points, so as to always iterate with respect to the best iterate $\mathbf{u}_k^*$,
- a few technical assumptions on the model $\phi_{\hat{p}}$.

The first two points are what is essentially "missing" in Algorithm 1, in that nothing is done to control the locality of the model and in that the latest tested point is always used as the reference $\mathbf{u}_k^*$, with the candidate being blindly accepted regardless of whether it is actually better or not. With respect to the necessary assumptions, one retains Assumptions 1-3 and simply enforces the following modified versions of Assumptions 4-5:

Assumption 8. *For all k, $\phi_{\hat{p}}$ is twice differentiable over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$.*

[1] In the author's opinion, this is very likely due to modifier-adaptation and trust-region methods being developed independently in the engineering and mathematical communities, respectively.

Assumption 9. *For all k, the Hessian of $\phi_{\hat{p}}$ over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ is upper-bounded.*

It is sufficient to make these assumptions on $\phi_{\hat{p}}$ as the addition of the affine modifier correction will not compromise the validity of these assumptions for the modified cost.

Note again that Assumption 6 is satisfied by construction in Algorithm 1 for $m_k(\mathbf{u}) := \phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k^*)$ and the refined definitions of the modifiers:

$$\varepsilon_k := \phi_p(\mathbf{u}_k^*) - \phi_{\hat{p}}(\mathbf{u}_k^*), \tag{13}$$

$$\boldsymbol{\lambda}_k := \nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi_{\hat{p}}(\mathbf{u}_k^*). \tag{14}$$

The remaining Assumption 7 should always hold if the minimization of (2) is functional (i.e., if minimization to a local minimum takes place). In the case when algorithmic/numerical difficulties prevent this, the sufficient decrease for the modified model may be guaranteed by essentially doing a line search in the steepest descent direction to obtain the Cauchy decrease (Conn et al., 2000).

We may now state a globally convergent version of Algorithm 1.

Algorithm 3 (Globally Convergent Modifier-Adaptation Algorithm)

1. (Initialization) Identical to Step 1 of Algorithm 2.
2. (Modifier Computation) Compute the modifiers ε_k and $\boldsymbol{\lambda}_k$ according to (13) and (14).
3. (Model Correction) Construct the model $m_k(\mathbf{u}) := \phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k^*)$.
4. (New Input Candidate Calculation) Compute a candidate point $\mathbf{u}_{k+1}$ as

$$\mathbf{u}_{k+1} := \arg \underset{\mathbf{u} \in \mathcal{B}(\mathbf{u}_k^*, \Delta_k)}{\text{minimize}} \quad m_k(\mathbf{u}) \ . \tag{15}$$

5. (Acceptance of the Candidate Point) Identical to Step 4 of Algorithm 2.
6. (Trust-Region Radius Update) Identical to Step 5 of Algorithm 2.
7. (Iterate) Set $k := k + 1$ and return to Step 2.

The following key result follows.

Theorem 3. (Global Convergence to a First-Order Critical Point for Modifier Adaptation) *If Assumptions 1-3 and 7-9 are satisfied, it then follows that the iterates generated by Algorithm 3 converge asymptotically to a first-order critical point, i.e.:*

$$\lim_{k \to \infty} \|\nabla\phi_p(\mathbf{u}_k^*)\| = 0. \tag{16}$$

Proof. Algorithm 3 is special case of Algorithm 2, with all of the assumptions needed for Theorem 2 satisfied. □

5. Nonequivalent Cases and Some Practical Considerations

Needless to say, not every modifier-adaptation algorithm will be a special case of the basic trust-region algorithm. For example, a technique that is often employed in modifier-adaptation schemes is to filter the modifiers (Marchetti et al., 2009; Serralunga et al., 2013), and to define them in the following manner:

$$\varepsilon_k := \alpha \left[\phi_p(\mathbf{u}_k^*) - \phi_{\hat{p}}(\mathbf{u}_k^*)\right] + (1-\alpha)\varepsilon_{k-1}, \tag{17}$$

$$\boldsymbol{\lambda}_k := \alpha \left[\nabla\phi_p(\mathbf{u}_k^*) - \nabla\phi_{\hat{p}}(\mathbf{u}_k^*)\right] + (1-\alpha)\boldsymbol{\lambda}_{k-1}, \tag{18}$$

starting from some initial values $\varepsilon_{-1}, \boldsymbol{\lambda}_{-1}$, with $\alpha \in (0, 1]$ a filter gain. For $\alpha < 1$, Assumption 6 is generally not satisfied, and one thus cannot apply the same global convergence analysis to such algorithms. However, one might very well ask if this sort of filtering is really necessary, as that which it is meant to promote (i.e., convergence) appears to be easier obtained by adding a trust region.

Another major difference that comes up is the handling of constraints. The general approach in modifier adaptation is to modify the constraints as one would modify the cost function (Gao & Engell, 2005; Marchetti et al., 2009), and to define the next iterate as:

$$\begin{aligned} \mathbf{u}_{k+1} := \arg \underset{\mathbf{u}}{\text{minimize}} \quad & \phi_{\hat{p}}(\mathbf{u}) + \varepsilon_k + \boldsymbol{\lambda}_k^T(\mathbf{u} - \mathbf{u}_k^*) \\ \text{subject to} \quad & g_{\hat{p},j}(\mathbf{u}) + \varepsilon_{g_j,k} + \boldsymbol{\lambda}_{g_j,k}^T(\mathbf{u} - \mathbf{u}_k^*) \leq 0, \ , \\ & j = 1, ..., n_g \end{aligned} \tag{19}$$

with g denoting the n_g constraint functions and $\varepsilon_{g_j,k}, \boldsymbol{\lambda}_{g_j,k}$ denoting the corresponding modifiers. As the trust region in trust-region methods is generally regulated with respect to the cost function (and does not directly accomodate constraints), the most common approach is to use an augmented cost function that penalizes constraint violations and the solution of which can be shown to be equivalent to the solution of the costrained problem for a sufficiently high penalty coefficient (Conn et al., 2000). In this case, one should be able to derive a globally convergent modifier-adaptation scheme by using the augmented cost function in place of the original in Algorithm 3 – in fact, the recent work of Biegler et al. (2014) essentially does this without stating so explicitly.

Finally, this story of equivalences truly becomes messy when one leaves the ideal setting – by "ideal", we refer to the fact that one can neither obtain the function values nor its derivatives exactly in experimental real-time optimization. Working with the estimates of $\phi_p(\mathbf{u}_k)$ and $\nabla\phi_p(\mathbf{u}_k)$ leads to estimated modifiers for which global-convergence guarantees are more difficult to derive. Unfortunately, as there is little work with regard to such problems in the mathematical literature (the reader is nevertheless referred to Carter (1991) for a look at trust-region methods that use *inexact* gradient information), there is little that one could export to benefit modifier adaptation for such scenarios. However, it is the author's belief that there is much to be gained by considering

the derivative-free trust-region framework (Conn et al., 2009) and some of its more recent robust versions (Larson, 2012).

References

Biegler, L. T., Lang, Y., & Lin, W. (2014). Multi-scale optimization for process systems engineering. *Comput. Chem. Eng.*, *60*, 17–30.

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

Bunin, G., François, G., & Bonvin, D. (2012). Exploiting local quasiconvexity for gradient estimation in modifier-adaptation schemes. In *2012 American Control Conference (Montréal)* (pp. 2806–2811).

Bunin, G. A., François, G., & Bonvin, D. (2013). Sufficient conditions for feasibility and optimality of real-time optimization schemes - I. Theoretical foundations. *arXiv:1308.2620v2 [math.OC]*, (pp. 1–57).

Bunin, G. A., François, G., Srinivasan, B., & Bonvin, D. (2011). Input filter design for feasibility in constraint-adaptation schemes. In *18th World Congress of the International Federation of Automatic Control (IFAC) (Milan)* (pp. 5585–5590).

Carter, R. (1991). On the global convergence of trust region algorithms using inexact gradient information. *SIAM J. Numer. Anal.*, *28*, 251–265.

Chachuat, B., Marchetti, A., & Bonvin, D. (2008). *Convergence of Modifier-Adaptation Schemes for Real-Time Optimization*. Technical Report EPFL.

Conn, A., Scheinberg, K., & Vicente, L. (2009). *Introduction to Derivative-Free Optimization*. Cambridge University Press.

Conn, A. R., Gould, N. I. M., & Toint, P. L. (2000). *Trust-Region Methods*. SIAM.

Costello, S., François, G., Bonvin, D., & Marchetti, A. (2013). Real-time optimization when the plant and the model have different inputs. In *Dynamics and Control of Process Systems (DYCOPS) (Mumbai)*.

François, G., & Bonvin, D. (2013). Use of convex model approximations for real-time optimization via modifier adaptation. *Ind. Eng. Chem. Res.*, *52*, 11614–11625.

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

Larson, J. M. (2012). *Derivative-free Optimization of Noisy Functions*. Ph.D. thesis University of Colorado.

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

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

Navia, D., Marti, R., Sarabia, D., Gutirrez, G., & de Prada, C. (2012). Handling infeasibilities in dual-modifier methodology for real-time optimization. In *8th IFAC symposium on advanced control of chemical processes* (pp. 537–542).

Roberts, P. (1978). Algorithms for integrated system optimisation and parameter estimation. *Electron. Lett.*, *14*, 196–197.

Rodger, E. (2010). *Dual Modifier Adaptation Methodology For the On-line Optimization of Uncertain Processes*. Master's thesis McMaster University.

Serralunga, F. J., Mussati, M. C., & Aguirre, P. A. (2013). Model adaptation for real-time optimization in energy systems. *Ind. Eng. Chem. Res.*, *52*, 16795–16810.