Journal of Process Control 54 (2017) 38–46

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

A feasible-side globally convergent modifier-adaptation scheme

Alejandro G. Marchetti [a,*,1], Timm Faulwasser [a,b], Dominique Bonvin [a]

[a] Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
[b] Institute for Applied Informatics, Karlsruhe Institute of Technology, D-76131 Karlsruhe, Germany

ARTICLE INFO

Article history:
Received 25 April 2016
Received in revised form
23 December 2016
Accepted 22 February 2017
Available online 12 April 2017

Keywords:
Real-time optimization
Static optimization
Optimization under uncertainty
Global convergence
Feasible-side convergence

ABSTRACT

In the context of static real-time optimization (RTO) of uncertain plants, the standard modifier-adaptation scheme consists in adding first-order correction terms to the cost and constraint functions of a model-based optimization problem. If the algorithm converges, the limit is guaranteed to be a KKT point of the plant. This paper presents a general RTO formulation, wherein the cost and constraint functions belong to a certain class of convex upper-bounding functions. It is demonstrated that this RTO formulation enforces feasible-side global convergence to a KKT point of the plant. Based on this result, a novel modifier-adaptation scheme with guaranteed feasible-side global convergence is proposed. In addition to the first-order correction terms, quadratic terms are added in order to convexify and upper bound the cost and constraint functions. The applicability of the approach is demonstrated on a constrained variant of the Williams–Otto reactor for which standard modifier adaptation fails to converge in the presence of plant-model mismatch.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In many industrial processes, there are strong economic incentives for finding the operating conditions that optimize some performance criterion while satisfying operating constraints. In many cases, the optimization problem relies on first-principles models and can be formulated as a nonlinear program (NLP). As examples, one can mention the steady-state optimization of continuously operating processes [8,11,23] and the optimization of batch and semi-batch processes using parameterized input profiles [28]. In the presence of plant-model mismatch and time-varying disturbances, it is necessary to continuously guide the operation towards the optimum. For this purpose, several model-based real-time optimization (RTO) schemes have been proposed, which iteratively update the model-based NLP problem using some adaptation strategy based on measurements.

The standard strategy used in industry is the two-step approach of parameter estimation followed by re-optimization [9,11]. The parameters of a first-principles model are estimated based on measurements available at the current operating point, and the updated model is used in the optimization problem to compute the next operating point. However, it is well known that, in the presence of structural plant-model mismatch, the two-step approach does not in general converge to the plant optimum [4,12,17]. In response to this deficiency, a modified two-step approach known as *Integrated Systems Optimization and Parameter Estimation* (ISOPE) was proposed by Roberts and co-workers [23,24]. ISOPE incorporates plant-gradient information in a gradient-modification term that is added to the cost function of the optimization problem, such that the Karush–Kuhn–Tucker (KKT) optimality conditions *for the plant* are satisfied upon convergence. The ISOPE algorithm was simplified by Tatjewski [30] by eliminating the parameter estimation problem. Gao and Engell [16] extended the approach of Tatjewski [30] to problems with process-dependent constraints by including first-order correction terms to the constraints in the optimization problem. Finally, Marchetti et al. [17] used the same type of first-order correction terms in the cost *and* constraint functions, and labeled the approach *Modifier Adaptation*, providing a comprehensive analysis of many of the algorithm's properties, such as optimality upon convergence, model-adequacy conditions, and necessary conditions for local asymptotic convergence. Since then, many variants of modifier adaptation have been proposed, such as *dual* modifier adaptation [18,25,19], *directional* modifier adaptation [10], *nested* modifier adaptation [22], and *second-order* modifier adaptation [13].

Bunin et al. [7] made the crucial observation that basically none of the available RTO algorithms can provide practical or even conceptual guarantees for converging to the plant optimum with all

* Corresponding author.
E-mail addresses: marchetti@cifasis-conicet.gov.ar (A.G. Marchetti), timm.faulwasser@kit.edu (T. Faulwasser), dominique.bonvin@epfl.ch (D. Bonvin).

[1] Also with the French-Argentine International Center for Information and Systems Sciences (CIFASIS), CONICET-UNR, S2000EZP Rosario, Argentina.

http://dx.doi.org/10.1016/j.jprocont.2017.02.013
0959-1524/© 2017 Elsevier Ltd. All rights reserved.

the iterations being feasible points for the plant, and modifier adaptation is no exception to this assertion. A necessary condition for modifier adaptation to converge to the plant optimum is that the model be adequate, in the sense that it should predict the correct curvature of the cost function in the vicinity of the converged point [17]. François and Bonvin [15] showed that model adequacy is satisfied if the cost and constraint functions of the model used for optimization are selected as convex functions. A general abstract sufficient condition for the convergence of modifier adaptation is given in [13]. However, this condition is difficult to verify and there is no practical way to enforce it. Global convergence conditions have also been proposed in [6] by implementing modifier adaptation as a trust-region algorithm, and exploiting the global convergence results available in trust-region theory. However, the results in [13,6] cannot guarantee *feasible-side* convergence of modifier adaptation. Bunin et al. [7] proposed a set of *sufficient conditions for feasibility and optimality* (SCFO) that, when met by any RTO algorithm, would enforce feasible-side global convergence. The feasible solution computed via RTO is projected onto a cone of feasible descent directions for the cost and constraint functions, and then a line-search step is conducted to improve the cost and remain feasible.

In the mathematical programming literature, a number of sequential convex programming methods have been proposed for solving inequality-constrained nonconvex NLPs [2,21,29]. The idea therein is to replace the nonconvex cost and/or constraints in the optimization problem by convex inner approximations, which results in interior-side monotone convergence to a KKT point. In the present paper, using similar ideas, we present a feasible-side globally convergent RTO formulation, wherein the cost and constraint functions belong to a certain class of convex upper-bounding functions. We propose to construct the required upper-bounding functions by adding quadratic terms to the modified cost and constraint functions used in standard modifier adaptation. The main contribution of the present paper is a modifier-adaptation algorithm guaranteeing global feasible-side convergence to a KKT point of the plant in the presence of plant-model mismatch.

The rest of the paper is organized as follows. Section 2 recalls the RTO problem, presents the necessary conditions of optimality, and introduces the main definitions and assumptions used in this work. An RTO scheme based on using general convex upper-bounding functions is presented and analyzed in Section 3. In particular, a proof of feasible-side global convergence is provided. On the grounds of this result, a modifier-adaptation algorithm with convex upper-bounding functions is proposed in Section 4. The new modifier-adaptation algorithm is applied to an optimization problem that is defined for the Williams–Otto reactor in Section 5. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Real-time optimization

The purpose of static real-time optimization (RTO) is to optimize process operation by finding the solution to the following optimization problem

$$\begin{aligned} \min_{\mathbf{u}} \quad & \Phi_p(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \\ \text{s.t.} \quad & G_{p,i}(\mathbf{u}) := g_i(\mathbf{u}, \mathbf{y}_p(\mathbf{u})) \leq 0, \quad i = 1, \ldots, n_g, \\ & \mathbf{u} \in \mathcal{U}, \end{aligned} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the decision (or input) variables; $\mathbf{y}_p \in \mathbb{R}^{n_y}$ are the measured output variables; $\phi : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is the cost function to be minimized; $g_i : \mathbb{R}^{n_u} \times \mathbb{R}^{n_y} \to \mathbb{R}$, $i = 1, \ldots, n_g$, is the set of process-dependent inequality constraints; and $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^{n_u} : \mathbf{u}^{\mathrm{L}} \leq \mathbf{u} \leq \mathbf{u}^{\mathrm{U}}\}$. The notation $(\cdot)_p$ is used throughout for variables associated with the plant.

This formulation assumes that ϕ and g_i are known functions of $\mathbf{u}$ and $\mathbf{y}_p$, i.e., they can be directly measured or evaluated from the knowledge of $\mathbf{u}$ and the measurement of $\mathbf{y}_p$. However, the steady-state input-output mapping of the plant $\mathbf{y}_p(\mathbf{u})$ is typically unknown, and only an approximate nonlinear steady-state model is available:

$$\boldsymbol{F}(\mathbf{x}, \mathbf{u}) = \mathbf{0}, \tag{2a}$$

$$\mathbf{y} = \boldsymbol{H}(\mathbf{x}, \mathbf{u}), \tag{2b}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$ are the state variables, and $\mathbf{y} \in \mathbb{R}^{n_y}$ are the output variables predicted by the model. For $\mathbf{u}$ given, the solution to the system (2a) is given by:

$$\mathbf{x} = \boldsymbol{\xi}(\mathbf{u}), \tag{3}$$

where $\boldsymbol{\xi}$ is an operator expressing the steady-state mapping between $\mathbf{u}$ and $\mathbf{x}$. The steady-state input-output mapping predicted by the model can now be expressed as:

$$\mathbf{y}(\mathbf{u}) := \boldsymbol{H}(\boldsymbol{\xi}(\mathbf{u}), \mathbf{u}). \tag{4}$$

The model-based counterpart of Problem (1) is given by the following NLP:

$$\begin{aligned} \min_{\mathbf{u}} \quad & \Phi(\mathbf{u}) := \phi(\mathbf{u}, \mathbf{y}(\mathbf{u})) \\ \text{s.t.} \quad & G_i(\mathbf{u}) := g_i(\mathbf{u}, \mathbf{y}(\mathbf{u})) \leq 0, \quad i = 1, \ldots, n_g, \\ & \mathbf{u} \in \mathcal{U}, \end{aligned} \tag{5}$$

In the presence of plant-model mismatch, the solution to Problem (5) does not generally match the solution to Problem (1). In real-time optimization, the solution to Problem (1) is approached by iteratively re-evaluating the operating point applied to the plant. Let $\mathbf{u}_k$ denote the steady-state operating point applied to the plant at the kth RTO iteration. The next optimal RTO solution is obtained by solving the following model-based optimization problem:

$$\begin{aligned} \mathbf{u}_{k+1}^{\star} = \arg\min_{\mathbf{u}} \quad & \Phi_k(\mathbf{u}) \\ \text{s.t.} \quad & G_{i,k}(\mathbf{u}) \leq 0, \quad i = 1, \ldots, n_g, \\ & \mathbf{u} \in \mathcal{U}. \end{aligned} \tag{6}$$

In order to deal with plant-model mismatch, the models used for the cost function Φ_k and the constraint functions $G_{i,k}$, $i = 1, \ldots, n_g$, are typically updated at each RTO iteration k, based on collected measurements. Examples of updating strategies are the computation of new model parameters based on available plant data—i.e. two-step approaches [9,11]—and the computation of first-order correction terms, i.e. modifier-adaptation approaches [17].

For stability reasons, $\mathbf{u}_{k+1}^{\star}$ is usually filtered before it is applied to the plant:

$$\mathbf{u}_{k+1} = \mathbf{u}_k + K(\mathbf{u}_{k+1}^{\star} - \mathbf{u}_k), \tag{7}$$

where $K \in (0, 1]$ is the filter gain, with $K = 1$ meaning no filtering. The combination of (6) and (7) constitutes an RTO algorithm.

In general, the design of any RTO algorithm should enforce the following desirable properties:

(i) **Plant optimality**: Despite structural mismatch between (1) and (6), a KKT point of (1) is reached upon convergence of (6)–(7).
(ii) **Plant feasibility**: All RTO iterates $\mathbf{u}_k$ satisfy the constraints of (1).
(iii) **Monotonic cost improvement:** The performance is required to improve between consecutive RTO iterates.

Besides these important basic properties, one would like to have sufficiently fast convergence and sufficient robustness with respect

Download English Version:

https://daneshyari.com/en/article/4998466

Download Persian Version:

https://daneshyari.com/article/4998466

Daneshyari.com