Journal of Process Control 30 (2015) 104–116

Contents lists available at ScienceDirect

Journal of Process Control

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

Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization

L.T. Biegler [a,*], X. Yang [a], G.A.G. Fischer [b]

[a] Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[b] Petroleo Brasileiro S.A., Refineria Henrique Lage, REVAP, Sao Jose dos Campos, SP, Brazil

ARTICLE INFO

Article history:
Received 17 April 2014
Received in revised form 27 January 2015
Accepted 1 February 2015
Available online 5 March 2015

Keywords:
Nonlinear programming
Nonlinear model predictive control
Dynamic real-time optimization

ABSTRACT

Recent results in the development of efficient large-scale nonlinear programming (NLP) algorithms have led to fast, on-line realizations of optimization-based methods for nonlinear model predictive control (NMPC) and dynamic real-time optimization (D-RTO), with predictive nonlinear dynamic (e.g., first principle) models. For NMPC, optimization-based controllers are developed that lead to well-understood stability and robustness properties, even for large, complex plant models. The realization of NMPC requires the application of a fast NLP solver for time-critical, on-line optimization, as well as efficient NLP sensitivity tools that require 2–3 orders of magnitude less computation than the NLP solution. This leads to advanced step NMPC (asNMPC), which essentially eliminates computational delay. We also extend these capabilities to dynamic real-time optimization (D-RTO) with more general stage costs that are economically based. This overview also extends input to state stability (ISS) properties for asNMPC to handle active set changes, and also for D-RTO through convex regularizations. Two large scale distillation case studies, based on nonlinear first principle models, are presented that demonstrate the effectiveness of these approaches.

1. Introduction

For over three decades, real-time optimization (RTO) and model predictive control (MPC) have emerged as essential technologies for optimal process operation in the chemical and refining industry. More recently, MPC has been extended to nonlinear model predictive control (NMPC) in order to realize high-performance control of highly nonlinear processes. Moreover, for many applications there is a need for RTO to evolve from steady-state optimization models to dynamic models, especially for systems, such as batch and cyclic processes, that are never in steady state. Both NMPC and dynamic real-time optimization (D-RTO) allow the incorporation of first principle process models, which lead to on-line optimization strategies consistent with higher-level tasks, including scheduling and planning. The realization of both of these tasks requires fast optimization algorithms. A major concern is that computational times needed to solve these large-scale optimizations lead to feedback delays in implementation that can degrade performance and possibly destabilize the process. In addition, state estimation of the process must be accomplished in a similar, efficient manner. Nonlinear model predictive control for tracking and so-called "economic" stage costs, as well as associated state estimation tasks, are reviewed, formulated and analyzed in considerable detail in [46,36]. Due to advances described in [8,35], fundamental stability and robustness properties of NMPC are well-known, and many of the key issues related to the applicability and relevance of NMPC are understood. Moreover, the availability of detailed dynamic process models for off-line process analysis and optimization allows NMPC to be realized on challenging process applications. Nevertheless, an important hurdle is the cost and reliability of on-line computation; lengthy and unreliable optimization calculations lead to unsuccessful controller performance.

Several advances to NMPC address the important problem of computational delay. Newton-type strategies for constrained nonlinear processes were originally proposed in [32]; here the nonlinear dynamic model is linearized around a nominal trajectory, and a quadratic program (QP) is solved at every sampling time. More recently, a real-time iteration NMPC was proposed in [11] where only one Newton or QP iteration of the NLP is executed on-line at every sampling time, instead of solving the NLP completely. More generally, NMPC strategies have been developed that separate the optimization problem into an off-line NLP based on predicted states, and fast on-line calculation for the actual state [11,57,60,54,53,39,34]. In all of these cases, the dynamic model is

* Corresponding author. Tel.: +1 412 268 2232; fax: +1 412 268 7139.
E-mail address: biegler@cmu.edu (L.T. Biegler).

http://dx.doi.org/10.1016/j.jprocont.2015.02.001
0959-1524/

optimized over a moving horizon and sensitivity-based updates are made on-line.

The ability to perform nonlinear model-based control extends naturally to dynamic real-time optimization (D-RTO). Current practice in process applications decomposes economic optimization into two layers. First, real-time optimization (RTO) optimizes an economic objective with steady state models, leading to a setpoint handled by the lower-level control layer. The advanced control layer (using, e.g., NMPC) then tracks the setpoint to achieve a new steady state. However, this two-layer approach assumes that model disturbances and transients are neglected in the RTO layer [13]. Moreover, model inconsistency between layers and unresolved transient behavior may lead to unreachable setpoints [48]. Also, since the control layer has no information on dynamic economic performance, it may generate trajectories that simply track suboptimal setpoints to steady state [47,25].

Recent studies on dynamic real-time optimization (D-RTO) have reported significant performance improvements with economically oriented NMPC formulations [59,47,13,2]. In addition, stability theory supporting economically oriented NMPC requires development beyond the mature results for setpoint tracking based on a discrete Lyapunov analysis. This problem formulation and stability analysis must be modified to ensure a stable and robust D-RTO implementation, especially if optimum steady state operation is expected.

Finally, knowledge of the plant state is essential for realization of NMPC and D-RTO. In practice state information can only be inferred through a set of noisy measurements, in combination with the dynamic process model. For linear systems this is done via Kalman filters (KF). Based on linearizations of the nonlinear plant model, extended Kalman filters (EKF) have typically been applied [26,7]. Moreover, when integrating EKF and NMPC, a local separation principle has been established between the estimation and control problems [23,24]. On the other hand, EKF may have poor performance for highly nonlinear systems [9,42], thus spawning related estimation methods that include the unscented Kalman filter [28], the ensemble Kalman filter [14], and the particle filter [4]. While none of these methods deal with bounds on the states, recent extensions [51,50,43,42], have been developed to address these features.

On the other hand, the state estimation problem can also be formulated directly as an NLP through moving horizon estimation (MHE), which uses a moving window of past measurements to find the optimal state estimates with an objective function based on maximum likelihood concepts. MHE has very desirable asymptotic stability properties [45] with bounds on plant states handled directly by the NLP solver. Efficient algorithms for MHE are presented in [49,60,58,1,33,37], which also address computational delay.

A comprehensive survey of sensitivity-based concepts for MHE, NMPC and D-RTO was presented in [5]. There we addressed recent results for NMPC, MHE and D-RTO that are based on *advanced step* concepts that particularly focus on efficient NLP algorithms for background solutions, along with on-line updates based on NLP sensitivity. This leads to asNMPC and asMHE, respectively. Moreover, recent MHE and asMHE extensions to efficient updating of arrival costs and outlier detection were developed and demonstrated on a distillation case study.

This study builds on the survey in [5] and also focuses on new results related to sensitivity-based NMPC and D-RTO. In the next section we present the basic NMPC problem formulation and review an optimization framework based on interior-point NLP solvers and sensitivity concepts. Section 3 presents advanced step NMPC (asNMPC) strategies and related stability properties. New results for ISS stability that incorporate controller clipping to deal with active set changes are also presented. Section 4 then describes a multi-step extension of asNMPC that allows very large process models to be solved in background over multiple time steps. This approach is illustrated on a large-scale distillation example. Section 5 then discusses recent updates for Economic NMPC properties and demonstrates their impact with a D-RTO distillation case study. Finally, Section 6 summarizes the paper along with directions for future work.

2. NLP strategies for NMPC

We begin with the following discrete-time nonlinear dynamic model of the plant with uncertainties:

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), u(k), w(k)) \\ &= f(x(k), u(k)) + g(x(k), u(k), w(k)) \end{aligned} \tag{1}$$

where $x(k) \in \Re^{n_x}$, $u(k) \in \Re^{n_u}$ and $w(k) \in \Re^{n_w}$ are the plant states, controls and disturbance signals, respectively, defined at time steps t_k with integers $k > 0$. The mapping $f : \Re^{n_x+n_u} \mapsto \Re^{n_x}$ with $f(0, 0) = 0$ represents the nominal model, while the term $g : \Re^{n_x+n_u+n_w} \mapsto \Re^{n_x}$ is used to describe modeling errors, estimation errors and disturbances. $f(\cdot, \cdot)$ and $g(\cdot, \cdot, \cdot)$ are continuous. We assume that the noise $w(k)$ is drawn from a bounded set $\mathcal{W}$.

With this model description, we compute an estimate of the current state $x(k)$ that can be used for our model-based controller, defined by the following NMPC problem:

$$J_N(\eta_0) := \min_{z_l, v_l} \quad \Psi(z_N) + \sum_{l=0}^{N-1} \psi(z_l, v_l) \tag{2a}$$

$$\text{s.t.} \quad z_{l+1} = f(z_l, v_l) \quad l = 0, \ldots N-1 \tag{2b}$$

$$z_0 = \eta_0 \tag{2c}$$

$$z_l \in \mathbb{X}, \quad v_l \in \mathbb{U}, \quad z_N \in \mathbb{X}_f. \tag{2d}$$

Here $\eta_0 = x(k)$ and we assume that the states and controls are restricted to the domains $\mathbb{X}$ and $\mathbb{U}$, respectively. $\mathbb{X}_f$ is the terminal set and $\mathbb{X}_f \subset \mathbb{X}$. We assume that N is sufficiently long such that $z_N \in \mathbb{X}_f$ is always true for the solution of (2). The set $\mathbb{U}$ is compact and contains the origin; the sets $\mathbb{X}$ and $\mathbb{X}_f$ are closed and contain the origin in their interiors.

The stage cost is given by $\psi(\cdot, \cdot) : \Re^{n_x+n_u} \to \Re$, while the terminal cost is denoted by $\Psi(\cdot) : \Re^{n_x} \to \Re$. For tracking problems, we can assume that the states and control variables can be defined with respect to setpoint and reference values, and that the nominal model has the property, $f(0, 0) = 0$.

After solution of (2) the control action is extracted from the optimal trajectory $\{z_0^*, \ldots, z_N^*, v_0^*, \ldots, v_{N-1}^*\}$ as $u(k) = v_0^*$. At the next time step, the plant evolves as in (1) and we shift the time sequence one step forward, $k = k+1$, obtain the next state estimate $x(k+1)$ and solve the next NMPC problem (2). This recursive strategy gives rise to the feedback law,

$$u(k) = \kappa(x(k)) \tag{3}$$

with $\kappa(\cdot) : \Re^{n_x} \to \Re^{n_u}$. With the feedback law (3) system (1) becomes

$$\begin{aligned} x(k+1) &= \hat{f}(x(k), \kappa(x(k)), w(k)) \\ &= f(x(k), \kappa(x(k))) + g(x(k), \kappa(x(k)), w(k)) \\ &= f(x(k), \kappa(x(k))) + g(x(k), w(k)) \end{aligned} \tag{4}$$

Hence, we replace $g(x(k), u(k), w(k))$ with $g(x(k), w(k))$ since $u(k) = \kappa(x(k))$. We refer to the above strategy as *ideal NMPC* (iNMPC), where the on-line calculation time is neglected. Correspondingly we denote the feedback law of iNMPC as $u^{id}(k) = \kappa^{id}(x(k))$. iNMPC has well-known stability properties (see [35,58]) that will be discussed in the next section. On the other hand, the computation