Computers and Chemical Engineering 55 (2013) 1–18

Contents lists available at SciVerse ScienceDirect

Computers and Chemical Engineering

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

A mixed-integer model predictive control formulation for linear systems

Lincoln F.L. Moro[a,*], Ignacio E. Grossmann[b]

[a] Petrobras S.A., Sao Paulo, Brazil
[b] Carnegie Mellon University, Pittsburgh, PA, USA

ARTICLE INFO

Article history:
Received 1 August 2012
Received in revised form 26 March 2013
Accepted 1 April 2013
Available online 17 April 2013

Keywords:
Mixed integer programming
Predictive control
Hybrid control

ABSTRACT

Since their inception in the early 1980s industrial model predictive controllers (MPC) rely on continuous quadratic programming (QP) formulations to derive their optimal solutions. More recent advances in mixed-integer programming (MIP) algorithms show that MIP formulations have the potential of being advantageously applied to the MPC problem. In this paper, we present an MIP formulation that can overcome difficulties faced in the practical implementation of MPCs. In particular, it is possible to set explicit priorities for inputs and outputs, define minimum moves to overcome hysteresis, and deal with digital or integer inputs. The proposed formulation is applied to simulated process systems and the results compared with those achieved by a traditional continuous MPC. The solutions of the resulting mixed-integer quadratic programming (MIQP) problems are derived by a computer implementation of the Outer Approximation method (OA) also developed as part of this work.

1. Introduction

Most industrial model predictive controllers currently in use are based on the algorithms developed in the early 1980s (Qin & Badgwell, 2003). These algorithms have two main functions, i.e., to reduce the process variability through better dynamic control and to move the operating point closer to the constraints, which in general results in significant economic benefits. In order to perform these functions, the usual practice is to adopt a hierarchical structure with two layers where the upper layer deals with the steady-state problem of defining optimal targets for inputs and outputs, while the lower layer, responsible for the dynamic problem, calculates the control moves that drive the system toward these steady-state targets.

The upper layer solves an optimization problem aiming at minimizing a linear combination of the projected steady-state values of the inputs, while simultaneously minimizing the square of the moves to be imposed on these inputs. Linear relations among inputs and outputs, and constraints limiting the allowable range of both kinds of variables are also imposed. As a result of these constraints the problem may be infeasible, and this fact demands the implementation of a relaxation strategy in order to guarantee that some kind of solution is always found. The lower layer involves an optimization problem that is always feasible because it includes constraints only on the manipulated inputs.

We propose to replace both optimization problems by mixed-integer (MIP) formulations, thus building a hybrid MPC. Several advantages may result from such an approach; for instance, the possibility of assigning explicit priorities for the outputs, i.e., the definition of a preferential order of constraint relaxation in case the initial steady-state problem proves infeasible. The inputs can also receive explicit priorities to select the order in which they are to be moved to adjust each output. The formulation also makes it possible to set a minimum limit for the control moves, i.e., any movement must be greater than a limit that is defined large enough to overcome the hysteresis of valves significantly affected by this problem.

The MIP formulation also allows the controller to deal with discrete inputs, either manipulated variables or disturbances, i.e., variables that can assume only a set of discrete values like for instance, 0 or 1 (on or off).

Hybrid formulations for MPC have been developed and successfully used in industrial applications as described for instance by Bemporad and Morari (1999), Morari and Barić (2006), Zabiri and Samyudia (2006), and Oldenburg and Marquardt (2008). Nevertheless, most of these contributions address the control of hybrid systems, while the present work focus on the development of a mixed-integer algorithm based on the traditional MPC that can be advantageously applied to continuous systems. Such a capability

* Corresponding author. Tel.: +55 11 3523 9813.
E-mail addresses: lmoro@petrobras.com.br, lmoro209@gmail.com (L.F.L. Moro).

0098-1354/$ – see front matter
http://dx.doi.org/10.1016/j.compchemeng.2013.04.011

has so far received little attention from the academic community, probably due to the inevitable increase in computational complexity in relation to the continuous formulations.

One instance of such a possible advantage can be identified in systems where two or more inputs present similar influence on the outputs. Due to the intrinsic multivariable characteristic of the process and the controller, the inputs will be moved at the same time. But frequently a better approach would be to use one of them for smaller moves and the other for larger ones. This is the case when valves of different dimensions are set in parallel lines with precisely the intention of allowing better manipulation of the inputs. The larger valve should only be used for larger flowrate changes, since smaller ones may not be actually implemented due to valve hysteresis.

Another characteristic, also related to the multivariable nature of the controller, is the manipulation of independent variables that have only a small influence on an output, especially when this last variable hits a constraint. This is the case, for example, of the feed flowrate, an input that affects almost every output in the plant. The controller, as a rule, aims at maximizing the feed but this may be prevented by almost any output hitting a constraint. To cope with this situation, a frequent practice is the outright elimination of the response model relating the feed and several less-important outputs. The undesired side-effect of this practice is that the controller will be unable to move the feedrate when this is the only solution to avoid constraint violation on such outputs, thus compromising the overall performance.

The relaxation algorithm used in the steady-state target calculation represents another opportunity for improvement. The usual algorithm basically dualizes some constraints, i.e., it transfers them to the objective function as terms minimizing the violation of the original constraints. This relaxation strategy frequently results in violations of the limits of variables that are currently within these limits, which is an undesirable change in the controller behavior. This happens because there is no straightforward way to determine which and how many limits should be relaxed. Additionally, when violations are unavoidable, some inputs are no longer minimized (or maximized) without any obvious reason for the plant operators.

Hence the algorithm proposed in this paper includes binary variables to represent the decisions to move the manipulated inputs during the control horizon, and these decisions can be penalized in the objective function or subjected to a priority sequence. This ensures that the available spans of the less important inputs are exhausted before the algorithm moves the more important ones. Binary variables are also included to represent the decisions to allow violations of the upper or lower limits of the controlled outputs. In an analogous way, these decisions can be penalized or prioritized, meaning that a specific sequence of permissions to violate the limits can be defined.

The inclusion of these binary decisions provides the control engineer additional freedom to define the expected behavior of the algorithm, but on the other hand this behavior can be negatively affected by an inadequate definition of priorities or weights, and it is important to state that this paper does not attempt to address the stability and robustness properties in any other way than through examples, and we recognize that this is an important issue that deserves future theoretical work. Additionally, this paper does not analyze the application of the proposed algorithm to nonlinear or hybrid systems, and this may also constitute a worthy subject for future investigation.

It is interesting to add that including the cited binary variables eases a future integration of the two layers that constitute the traditional linear controllers, since the main reason for keeping them as separate formulations is the fact that the original problem solved by the upper layer may be infeasible, and this is not the case in the proposed algorithm. The integration will probably result in a formulation whose theoretical properties can be more easily analyzed.

The outline of the paper is as follows. In Section 2, the usual continuous MPC formulation, comprising the static and dynamic layers, is presented. Section 3 describes the proposed mixed-integer formulation for the static layer, responsible for deriving the steady-state targets for inputs and outputs. Section 4 deals with the formulation of the mixed-integer dynamic layer, which calculates the control actions. The mixed-integer quadratic programming solver developed to derive the solutions of both layers is described in Section 5. Section 6 covers the application of the proposed formulation on the 4-tank benchmark system, including a comparison of the results with the ones obtained with the traditional controller. Section 7 also deals with the application on a benchmark system, the Shell control problem. In Section 8, the proposed formulation is applied on a simulated industrial system and a comparison with the continuous controller is provided. Finally, Section 9 concludes this paper by briefly summarizing the results.

2. Continuous MPC formulation

According to Sotomayor, Odloak, and Moro (2009), the MPC target calculation layer, also called steady-state linear optimizer, solves at each sampling instant a QP problem where the objective is to force one or more inputs to their bounds, while keeping the outputs inside the bounds. This problem may be defined as follows:

$$\min_{\Delta\tilde{u},\delta^y} \varphi^{ss} = \frac{1}{2}\cdot\Delta\tilde{u}^T W_0\,\Delta\tilde{u} + W_1^T\,\Delta\tilde{u} + \delta^{y^T} W_2 \delta^y \tag{1}$$

subject to

$$\begin{aligned} &\Delta\tilde{u} = \tilde{u} - u \\ &\tilde{y} = G_0\,\Delta\tilde{u} + \hat{y}_{k+n|k} \\ &u^{LB} \le \tilde{u} \le u^{UB} \\ &y^{LB} \le \tilde{y} + \delta^y \le y^{UB} \end{aligned} \tag{2}$$

where u is the vector of the current values of the inputs (implemented at time $k-1$); $\tilde{u}$, vector of steady-state targets of the inputs; $\tilde{y}$, vector of steady-state targets of the outputs; δ^y, vector of slack variables for the controlled outputs; G_0, steady-state gain matrix of the process; k, the present time step; n, settling time of the process in open loop; W_0, W_1, W_2, weight matrices; u^{LB}, u^{UB}, vector of bounds of the manipulated inputs; y^{LB}, vector of lower operation limits for the outputs; and y^{UB}, vector of upper operation limits for the outputs.

In the equations above, $\hat{y}_{k+n|k}$ represents the contributions of the past inputs to the predicted output at time step $k+n$, i.e., at the end of the time horizon and is obtained from the MPC dynamic layer as the predicted output value at the end of the time horizon, i.e., after system stabilization.

The solution of the problem defined by Eqs. (1) and (2) generates the input targets that are transferred to the MPC dynamic layer. The version of MPC we consider in this work is a modification of the quadratic dynamic matrix control (QDMC) as described in García and Morshedi (1986) and Soliman, Swartz, and Baker (2008). This version solves the following optimization problem:

$$\min_{\Delta\bar{U}_k} \varphi^{qdmc} = (\bar{Y}_k - \tilde{Y})^T Q(\bar{Y}_k - \tilde{Y}) + \Delta\bar{U}_k^T \Lambda\,\Delta\bar{U}_k \tag{3}$$

subject to:

$$\begin{aligned} &-\Delta u^{UB} \le \Delta\bar{u}_{k+\ell-1} \le \Delta u^{UB} && \forall \ell = 1,\ldots,m \\ &u^{LB} \le \bar{u}_{k+\ell-1} \le u^{UB} && \forall \ell = 1,\ldots,m \\ &\bar{Y}_k = A\Delta\bar{U}_k + \hat{Y}_k \end{aligned} \tag{4}$$