USE OF MIXED-INTEGER FORMULATIONS IN PREDICTIVE CONTROL ALGORITHMS

Lincoln F. L. Moro[*a], Ignacio E. Grossmann[b]
[a]Petrobras S.A. – Sao Paulo, Brazil
[b]Carnegie Mellon University - Pittsburgh, USA

Abstract

Most industrial model predictive controllers (MPC) use the traditional two-layer structure developed in the early 1980's, where the upper layer defines optimal steady-state targets for inputs and outputs, while the lower layer calculates the control moves that drive the system towards these steady-state targets. As a rule, both layers use continuous quadratic programming (QP) formulations to derive the optimal solutions. On the other hand, the advances in mixed-integer programming (MIP) algorithms and their successful utilization to solve large scheduling problems in reasonable time show that MIP formulations have the potential of being advantageously applied to the multivariable model predictive control problem. In this paper we present a mixed-integer quadratic programming (MIQP) formulation for the steady state targets calculation layer, and show that several difficulties faced in the MPC practical implementation can be overcome with this approach. 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 was applied to a simulated industrial system and the results compared with those achieved by a traditional continuous MPC.

Keywords

Mixed integer programming, predictive control, hybrid control.

Introduction

Most industrial model predictive controllers currently in use are based on the algorithms developed in the early 1980's (Qin and 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 towards these steady-state targets.

The upper layer solves an optimization problem aiming at minimizing a linear combination of the projected steady–sate values of the inputs, and simultaneously minimizes 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 will always be found.

The lower layer involves an optimization problem that includes constraints only on the inputs, which guarantees that a feasible solution can always be found.

We propose to replace both optimization problems by a mixed-integer (MIP) formulation, thus building a hybrid MPC. Several advantages may result from such a formulation; 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

* To whom all correspondence should be addressed

are to be moved to adjust each output. The formulation also makes it possible to set a minimum limit for control moves, which is adequate for valves subject to hysteresis.

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 already been developed, and successfully used in industrial applications as described for instance by Bemporad and Morari (1999), Morari and Barić (2006), and Zabiri and Samyudia (2006). Nevertheless, most of these contributions address the control of hybrid systems, while we are focusing on the development of a mixed-integer algorithm based on the traditional MPC that can be advantageously applied even to continuous systems.

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 adjustment 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 difficulty, also related to the multivariable nature of the controller, is the change in independent variables that have only a small influence on an output, especially when this variable hits a constraint. This is the case, for example, of the feed flowrate, which is 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.

Another opportunity for improvement concerns the relaxation algorithm used in the steady state target calculation, which basically involves transferring some constraints into the objective function through terms that minimize the violation of such constraints. This relaxation frequently results in violations of the limits of variables that are currently within these limits, which is a puzzling 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.

Continuous MPC formulation

According to Sotomayor et al. (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:

$$\underset{\Delta u^*, \delta_y}{Min} \quad \varphi^{ss} = \frac{1}{2}.\Delta u^* W_0 \Delta u^* + W_1^T \Delta u^* + W_2^T \delta_y \qquad (1)$$

subject to:

$$\begin{aligned} &\Delta u^* = u^* - u \\ &y^* = G_0 \Delta u^* + \hat{y}_{k+n|k} \\ &u^L \le u^* \le u^U \\ &y^L \le y^* + \delta_y \le y^U \end{aligned} \qquad (2)$$

where:
u = inputs implemented at time $k-1$,
u^* = vector of steady-state targets of the inputs,
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,
n = settling time of the process in open loop,
W_0, W_1, W_2 = weight vectors,
u^L, u^U = bounds of the manipulated inputs,
y^L, y^U = bounds of the controlled outputs.

In the 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.

The solution of the problem defined by equations (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 et.al (2008). This version solves the following optimization problem:

$$\begin{aligned} &\min_{\Delta \hat{u}_k} \varphi^{qdmc} = \left(\bar{y}_k - y_{sp}\right)^T Q \left(\bar{y}_k - y_{sp}\right) + \\ &+ \Delta \bar{u}_k^T \Lambda \Delta \bar{u}_k + \left(\bar{u}_k - u^*\right)^T R \left(\bar{u}_k - u^*\right) \end{aligned} \qquad (3)$$

Subject to:

$$
\begin{aligned}
&-\Delta u^U \le \Delta \bar{u}_k \le \Delta u^U \\
&u^L \le \bar{u}_k \le u^U \\
&\bar{y}_k = A\Delta \bar{u}_k + \hat{y}_k
\end{aligned} \quad (4)
$$

where:

$\bar{y}_k = \left[\bar{y}_{k+1|k}, \bar{y}_{k+2|k}, \cdots, \bar{y}_{k+p|k}\right]^T$

y^{sp} = set-point to the system output. This set-point is usually made equal to y^*.

$\bar{u}_k = \left[\bar{u}_{k|k}, \bar{u}_{k+1|k}, \cdots, \bar{u}_{k+m-1|k}\right]^T$

$\Delta\bar{u}_k = \left[\Delta\bar{u}_{k|k}, \Delta\bar{u}_{k+1|k}, \cdots, \Delta\bar{u}_{k+m-1|k}\right]^T$

$\hat{y}_k = \left[\hat{y}_{k+1|k}, \hat{y}_{k+2|k}, ..., \hat{y}_{k+n|k}\right]^T$

Δu^U = upper limit to the control moves,
m = control horizon,
p = prediction horizon,
Q, Λ and R are weighting matrices.

In the equations above, $\bar{y}_{k+\ell|k}$ represents the predicted output at time step $k+\ell$, based on information available at time step *k*, including the planned control moves. This prediction also includes the effect of measured disturbances that were not shown for the sake of simplicity. The dynamic matrix A relates future input changes $\Delta\bar{u}_k$ to predicted outputs $\bar{y}_k$.

This formulation is similar to the structure of several MPC packages widely applied in refining and petrochemical processes.

Steady-State Optimization using MIQP Approach

We propose to replace the steady state target calculation described by equations (1) and (2) by a mixed-integer quadratic problem (MIQP) as follows:

Objective function:

$$
\min_{\Delta u^*} \varphi^{miss} = \sum_{j=1}^{Nm}\left(\frac{1}{2}\Delta u_j^* \mu_j \Delta u_j^* - \lambda_j^u \Delta u_j^* + \pi_j^u z_j^u\right) + \\
+\sum_{i=1}^{Nc}\left(\frac{1}{2} y_i^v \varpi_i y_i^v - \sum_{i=1}^{Nc} \pi_i^y z_i^y\right) \quad (5)
$$

where:

μ_j = minimum movement tuning parameter for input $j,\ j = 1, \cdots, Nm$

λ_j^u = profit tuning parameter for input $j,\ j = 1, \cdots, Nm$

π_j^u = priority parameter for input $j,\ j = 1, \cdots, Nm$

π_i^y = priority parameter for output $i, i = 1, \cdots, Nc$

ϖ_i = weight parameter for output violation, $i = 1, \cdots, Nc$

z_i^y = decision to enforce the limits of output *i* (binary variable – if equal to 0 then the limits are relaxed), $i = 1, \cdots, Nc$

z_j^u = decision to move input *j* (binary variable), $j = 1, \cdots, Nm$

Nm = number of inputs ($u_j,\ j = 1, \cdots, Nm$)

Nc = number of outputs ($y_i, i = 1, \cdots, Nc$)

In order to allow the inclusion of constraints for the minimum movement of the inputs, we introduce the variables $\Delta u_j^{*+}, \Delta u_j^{*-} \ge 0$, such that:

$$
\Delta u_j^* = \Delta u_j^{*+} - \Delta u_j^{*-} \qquad \forall j = 1, ..., Nm \quad (6)
$$

Equality constraints

Equations defining the amount of upper and lower limit violations for each CV:

$$
y_i^v + \sum_{j=1}^{Nm} G_{i,j}\left(\Delta u_j^{*+} - \Delta u_j^{*-}\right) + \hat{y}_i - y_i^L \ge 0, \forall i = 1, ..., Nc \quad (7)
$$

$$
y_i^v - \sum_{j=1}^{Nm} G_{i,j}\left(\Delta u_j^{*+} - \Delta u_j^{*-}\right) + y_i^U - \hat{y}_i \ge 0, \forall i = 1, ..., Nc \quad (8)
$$

where:
y_i^L = lower operation limit for output *i*
y_i^U = upper operation limit for output *i*
Additionally,

$$
y_i^v \ge 0 \qquad \forall i = 1, ..., Nc \quad (9)
$$

Equations defining the decision to satisfy the limits of each output.

Lower limit:

$$
y_i^* \ge y_i^L + M\left(z_i^y - 1\right) \quad \forall i = 1, ..., Nc \quad (10)
$$

Upper limit:

$$y_i^U - y_i^* \geq M\left(z_i^y - 1\right) \quad \forall i = 1,...,Nc \tag{11}$$

y_i^U = upper limit for output *i*
y_i^L = lower limit for output i
M = big-M constant.

Equations defining the decision to move each input

$$\Delta u_j^{*+} \leq M z_i^u \qquad \forall j = 1,...,Nm \tag{12}$$

$$\Delta u_j^{*-} \leq M z_i^u \qquad \forall j = 1,...,Nm \tag{13}$$

Minimum movement to be applied to an input if the decision to move it is taken.

$$\Delta u_j^{*+} + \Delta u_j^{*-} \geq \Delta u_j^L z_i^u \qquad \forall j = 1,..., Nm \tag{14}$$

where:
Δu_j^L = minimum change to be applied to input j, once the decision to move it is taken.

The formulation described above applies only to the steady-state targets calculation layer. The dynamic layer used in this study is a traditional MPC solved by a QP algorithm.

Mixed-integer quadratic programming solver

The MIQP problem described in the previous section was solved by the Outer-Approximation method (Duran and Grossmann, 1986), consisting of a series of QP subproblems and MILP master problems. In order to describe this algorithm we consider the optimization problem shown by eq. (15):

$$\text{P} \begin{cases} \min\limits_{x} \varphi = \frac{1}{2} x^T C x + D^T x \\ s.t. \\ Ax + B \geq 0 \\ x^L \leq x \leq x^U \end{cases} \tag{15}$$

where x is the vector of free variables x_i ($i = 1, \cdots, n$), which includes both continuous and discrete variables (in this paper we consider that the discrete variables are binary) and is divided into the I^c and I^b subsets, comprising the continuous and the binary variables, respectively. Vectors x^L and x^U contain the upper and lower bounds for x, which can also be described as:

$$x_i \in \left[x_i^L, x_i^U\right] \qquad \forall i \in I^c \tag{16}$$

$$x_i \in \{0,1\} \qquad \forall i \in I^b \tag{17}$$

C is an $n \times n$ positive definite matrix, *D* is an $n-$dimensional vector, *A* is an $m \times n$ matrix, and *B* is an $m-$dimensional vector, where m is the number of constraints.

The algorithm works according to the following sequence:

1 – Solve problem P as a relaxed QP, i.e., set $x_i \in [0,1]$, $\forall i \in I^b$ and let $x^{NLP,k}$, with $k = 0$, be the solution vector. If the solution is integral, which means that every $x_i^{NLP,0}, \forall i \in I^b$ has a value of 0 or 1, then this solution is optimal for P. Otherwise, proceed with the algorithm.

2 –Linearize the objective function around $x^{NLP,k}$, set *k = k + 1* and solve the following MILP:

$$\begin{aligned} & \min_{x} \varphi_{milp} = \alpha \\ & s.t. \\ & \alpha \geq \left(x^{NLP,j^T} C + D^T\right)x + \\ & -\frac{1}{2} x^{NLP,j^T} C x^{NLP,j}, \; j = 1, \cdots k \\ & Ax + B \geq 0 \\ & x^L \leq x \leq x^U \\ & \alpha \in R^1 \end{aligned} \tag{18}$$

which will result in a new optimal solution $x^{MILP,k}$.

3 - Fix the binary variables $x_i^k = x_i^{MILP,k}$, $\forall i \in I^b$ and solve P with only the $x_i, \forall i \in I^c$ as free variables, thus obtaining $x^{NLP,k}$. If the NLP objective function value $\phi^{NLP,k}$ is equal to $\phi^{MILP,k}$ within a given tolerance, the algorithm converged and $x_i^{NLP,k}$ is the optimum solution. Otherwise, proceed to step 2.

The QP subproblem was solved using the QL algorithm, written in FORTRAN by Schittkowski (2005), while the MILP master problem was solved by lp_solve, which is a freely available LP/MILP solver written by M. Berkelaar at Eindhoven University of Technology.

Process simulation

The proposed formulation was applied to a simulation of a Fluid Catalytic Cracking unit (FCC), as described by Moro and Odloak (1995).

The FCC is one of the most important refining processes, and transforms intermediate oil fractions into light and more valuable hydrocarbon products. The FCC converter, which is the main equipment of such units, consists of three major sections: the separator vessel, the regenerator and the riser. The riser is a tubular reactor where at the bottom the preheated liquid feed is injected, and mixed with hot fluidized catalyst flowing from the regenerator. This hot catalyst provides the energy for feed vaporization and for the endothermic cracking reactions. These reactions generate lighter hydrocarbons and also a high carbon-content, solid coke, which is deposited over the catalyst surface resulting in its deactivation. The catalyst is reactivated in the regenerator by burning the coke in a fluidized bed.

The MPC configuration used here was taken from the actual industrial implementation and includes 33 outputs and 11 manipulated inputs, and covers the plant subsection from the preheat train to the fractionator column. This configuration results in an MIQP with 55 continuous and 44 binary variables, as well as 165 constraints.

Although each one of the variables was kept active in the simulated test described in the next section, we will focus on the control of just one variable, the regenerator temperature, which is mainly affected by the air injection. The air is injected through 3 different pipes and adjusted by 3 flow controllers, FC01, FC01A and FC02, as depicted in Figure 1.

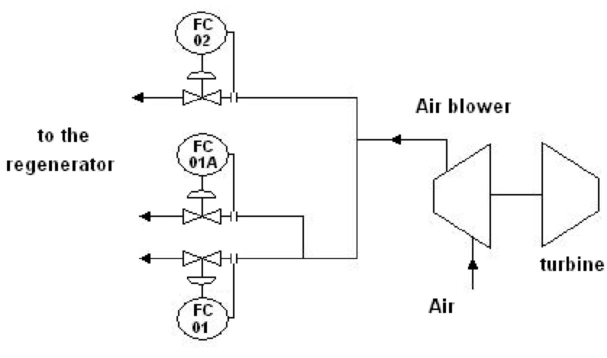

Figure 1. regenerator air subsystem

FC01 controls the flow in the main injection line and is responsible for about 60% of the total air. FC01A works as a complement to FC01 and is designed for frequent small adjustments. FC02 is responsible for about 15% of the total air flow and is connected to the regenerator second stage.

The best practice for this system consists in using the larger valve, i.e. FC01, only for aggressive control moves, while the smaller ones should be used to deal with the regular fluctuations. The application of frequent movements on the larger valve, besides being ineffective due to hysteresis, generates wear and tear that may lead to premature failure.

The usual approach adopted by control engineers to adjust the controller behavior in such cases, is to increase the move suppression term (Λ in eq. (3)) of the input responsible for the larger valve. This usually does not result in the desired behavior, and impairs the MPC ability to deal with situations when aggressive control actions are necessary.

In this simulated test, we show that the mixed-integer formulation is able to generate this behavior, i.e., to move the larger valve only for larger flow modifications, and still provide adequate regulation of the regenerator.

Simulated Testing

In this simulation, we evaluated the performance of the MIQP algorithm and compared it with the MPC currently used to control the plant. The system is allowed to reach steady state and then a change in the allowable range of the regenerator temperature – a controlled variable – is imposed. This change affects only the lower limit of the temperature, which is raised from 680^{o}C to 700^{o}C. The results are depicted in Figures 2 through 5, where the solid lines represent the behavior with the MIQP formulation, and the dotted lines the behavior with the traditional QP algorithm.

As it can be seen in Figure 2, the temperature profile is similar in both cases, with the MIQP algorithm being slightly faster but equally accurate.

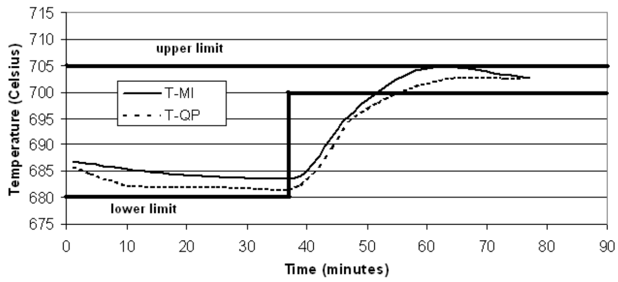

Figure 2. Regenerator temperature with the MIQP formulation (T-MI) and with the traditional QP (T-QP).

The behavior of the manipulated variables related to the air injection can be seen in the subsequent figures.

It can be noticed that with the MIQP formulation the manipulated variables stay more or less constant, while no setpoint changes are imposed on the controller. On the other hand, it is capable of vigorous action when such change happens. As previously described, this is exactly the kind of behavior that we were aiming for with this mixed-integer formulation

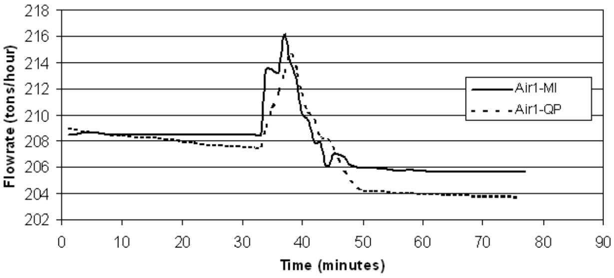

Figure 3. Main air flow to the Regenerator with the MIQP formulation (Air1-MI) and with the traditional QP (Air1-QP).

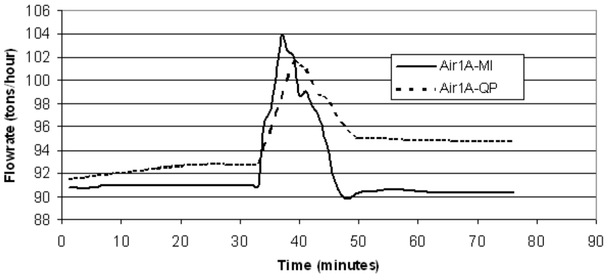

Figure 4. Secondary air flow to the Regenerator with the MIQP formulation (Air1A-MI) and with the traditional QP (Air1A-QP).

It is to be expected that better results will be obtained once the controller is retuned to utilize more freely the characteristics of the hybrid approach.

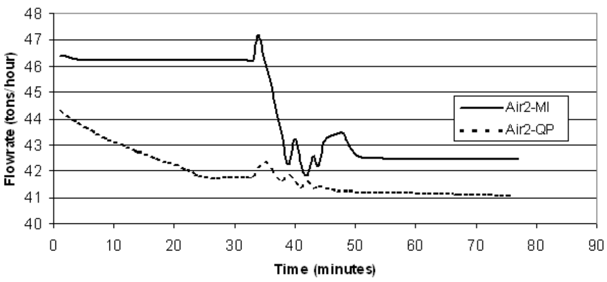

Figure 5. Air flow to the Regenerator second stage with the MIQP formulation (Air2A-MI) and with the traditional QP (Air2A-QP).

Conclusions

In this paper we have presented an MIQP formulation for the calculation of steady state targets that are generally used in industrial MPC algorithms. This formulation was applied to a simulated industrial case, and compared to the traditional continuous MPC. The results show that the desired behavior is obtained, even without any changes in the tuning parameters previously used.

As a follow-up to this work, we intend to develop an analogous MIQP formulation for the dynamic layer and integrate it to the steady state layer. The resulting algorithm will then be tested in a simulated process, and after validation, in an industrial refining unit.

Acknowledgments

The authors acknowledge the support of Petrobras S.A., without which this work would not be possible.

References

Bemporad, A., Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints, *Automatica* 35 (3) 407–427.

Duran, M.A., Grossmann, I.E. (1986). An outer-approximation algorithm for a class of mixed-integer nonlinear Programs. Math Programming, 36, 307-339.

García, C.E., Morshedi, A.M. (1986). Quadratic programming solution of dynamic matrix control (QDMC). *Chemical Engineering Communications*, 73-87.

Morari, M., Barić, M. (2006). Recent developments in the control of constrained hybrid systems. *Computers and Chemical Engineering*, 30, 1619-1631.

Moro, L. F. L., Odloak, D. (1995). Constrained multivariable control of fluid catalytic cracking converters. *Journal of Process Control*, 5(1), 29–39.

Qin, S. J., Badgwell, A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7), 733–764.

Schittkowski, K. (2005). QL: A Fortran Code for Convex Quadratic Programming - User's Guide, Version 2.11. *University of Bayreuth*. Bayreuth, Germany. http://www.ai7.uni-bayreuth.de/QL.pdf (accessed in Oct 27,2011)

Soliman, M., Swartz, C.L.E., Baker, R. (2008). A mixed-integer formulation for back-off under constrained predictive control. *Computers and Chemical Engineering*, 32, 2409-2419.

Sotomayor, O.A.Z., Odloak, D., Moro, L.F.L. Moro (2009). Closed-loop model re-identification of processes under MPC with zone control. *Control Engineering Practice.* 17, 551-563.

Zabiri, H., Samyudia, Y. (2006). A hybrid formulation and design of model predictive control for systems under actuator saturation and backlash. *Journal of Process Control.*, *16*, 693-709.