

# Mixed-integer MPC for the temperature control of the batch reactor

Jakub Novák<sup>a</sup> and Petr Chalupa

*Faculty of Applied Informatics, Tomas Bata University in Zlin, 76001 Zlin, Czech Republic*

**Abstract.** Discrete Model Predictive Control (MPC) became one of the most widespread modern control principles. Process controls with a finite number of admissible values are common in a large number of relevant applications. For this type of optimization problems, the computational complexity is exponential in the number of binary optimization variables. The solver is based on a standard branch-and-bound method and interior point method is used for solution of the relaxed problem. The simulation experiment involved controlling the temperature of a batch reactor by using two on/off input valves and a discrete-position mixing valve.

## 1 Introduction

Model Predictive Control (MPC) has become of the most widespread modern control strategies which have been successfully applied in many industrial applications. The idea of MPC is to determine an optimal control at the current time instant by solving an optimal control problem on a prediction horizon. The main reason for the wide-scale adoption of MPC is its ability to handle constraints on inputs and states that arise in most applications. MPC requires process model that allows it to react to future changes of the reference signal. MPC also naturally handles the multidimensional systems. Process controls with a finite number of admissible values are common in a large number of relevant applications. For example, chemical plants are equipped with valves that can be either open or closed. Hybrid systems represent a unified framework for modelling such processes that combine continuous and discrete dynamic with logical rules and appear in various applications like robotic systems and automotives. Bemporad and Morari [1] introduced a special class of hybrid systems called Mixed Logical Dynamical (MLD) systems. They can effectively model a variety of systems: hybrid automata, nonlinear systems with the nonlinearity represented by the piece-wise affine functions, linear systems with constraints, etc. The interest in hybrid systems has grown considerably during the last decade because of its potential impact on the industrial applications of control. Hybrid systems arise naturally in areas such as automotive systems [2], chemical systems [3] or power systems [4]. In all of these applications, there are several operation modes in the plant that justify the implementation of a hybrid MPC controller.

MPC is a general approach for control of hybrid systems. However, the optimization problem is no longer quadratic programming (QP) problem but a Mixed-

Integer Quadratic Programming (MIQP) problem. The inclusion of integer variables turns the easily solved QP problem, into an NP-hard problem [2]. It has been demonstrated in the recent research that convex QP problem can be solved at high sampling rates on embedded computing platforms [5]. The main methods are tailored optimization methods for particular embedded platform. Several application of embedded MIQP solvers have been proposed in [6], [7]. Many methods have been developed for solution of the MIQP problem, however branch-and-bound method have proved superior to other methods such as decomposition method or logic-based method [8]. Authors in Ref [9] provide a review of computational methods of solving MIQP problems. CPLEX and GUROBI represent a commercial optimization software that is able to solve the MIQP problems and also several toolboxes for MATLAB exists (OPTI Toolbox which uses SCIP solver [10], Hybrid toolbox [11]).

In this paper, MLD approach is used for modelling and predictive control of a batch reactor as a switched linear system.

## 2 Mixed logical dynamical systems

The MLD modelling framework is based on the idea of translating logic relations, discrete/logic dynamics, A/D (analogue to digital (logic)), D/A conversion and logic constraints into mixed integer linear inequalities. These inequalities are combined with the continuous dynamical part, which is described by linear difference equations. The resulting MLD system is described by the following relations:

<sup>a</sup> Corresponding author: jnovak@fai.utb.cz

$$\begin{aligned}
 x(k+1) &= A x(k) + B_1 u(k) + B_2 \delta(k) \\
 &\quad + B_3 z(k) + B_0 \\
 y(k) &= Cx(k) + D_1 u(k) + D_2 \delta(k) \\
 &\quad + D_3 z(k) + D_0 \\
 E_2 \delta(k) + E_3 z(k) &\leq A x(k) + E_1 u(k) \\
 &\quad + E_4 x(k) + E_5
 \end{aligned} \tag{1}$$

where

$x = \begin{bmatrix} x_c \\ x_b \end{bmatrix}$ ,  $x_c \in R^{n_c}$ ,  $x_b = \{0,1\}^{n_b}$ ,  $n_x = n_c + n_b$   $x$  are the system states,

$y = \begin{bmatrix} y_c \\ y_b \end{bmatrix}$ ,  $y_c \in R^{p_c}$ ,  $y_b = \{0,1\}^{p_b}$ ,  $n_y = p_c + p_b$  are the system outputs combining continuous and binary outputs and

$u = \begin{bmatrix} u_c \\ u_b \end{bmatrix}$ ,  $u_c \in R^{m_c}$ ,  $u_b = \{0,1\}^{m_b}$ ,  $n_u = m_c + m_b$  are the command inputs and may also contain continuous and binary commands,  $\delta$  and  $z$  are respectively auxiliary logical and continuous variables. The auxiliary variables are introduced when translating propositional logic into linear inequalities. In order to be well posed, the MLD system (1) must be such that for any given  $x(k)$  and  $u(k)$  the values of  $\delta(k)$  and  $z(k)$  are defined uniquely.

### 3 Predictive control of MLD systems

In [12], Bemporad and Morari introduced a model predictive control of hybrid systems using mixed logical dynamical (MLD) system description and a mixed integer linear program solver. Assuming a quadratic cost function from the MLD model the optimization has the following form:

$$\min J(u, z, \delta) = \sum_{k=0}^{N-1} \|u(k) - u_e\|_{Q_1}^2 + \|z(k) - z_e\|_{Q_2}^2 + \|\delta(k) - \delta_e\|_{Q_3}^2 \tag{2}$$

Subjected to equations which define the MLD system for  $N$  steps ahead predictions. The matrices  $Q_1, Q_2, Q_3, Q_4$  are given weight matrices. This problem can be rewritten to a standard MIQP programming form:

$$\begin{aligned}
 \min 0.5 \begin{bmatrix} x_c^T & x_i^T \end{bmatrix} H \begin{bmatrix} x_c^T \\ x_i^T \end{bmatrix} + f^T \begin{bmatrix} x_c^T \\ x_i^T \end{bmatrix} \\
 x_c \in R^{n_c}, x_i \in N^{n_i} : a_j^T \begin{bmatrix} x_c^T \\ x_i^T \end{bmatrix} + b_j = 0, \quad j = 1, \dots, m_{ec} \tag{3} \\
 c_j^T \begin{bmatrix} x_c^T \\ x_i^T \end{bmatrix} + d_j \geq 0, \quad j = 1, \dots, m_{ic}
 \end{aligned}$$

where  $n_c$  and  $n_i$  define the numbers of continuous and integer variables,  $H$  is a positive definite matrix,  $f$  is the

$n$ -dimensional vector. The  $n$ -dimensional vectors  $a_j$  and  $c_j$  and vectors  $b$  and  $d$  are used to set up the constraints. The numbers of equality and inequality constraints are specified with  $m_{ec}$  and  $m_{ic}$ , respectively. The equality and inequality constraints define a feasible region in which the solution to the problem must be located in order for the constraints to be satisfied. The optimization procedure of (3) leads to problems with the following optimization vector :

$$x = [u(k), \dots, u(k+N-1), \delta(k), \dots, \delta(k+N-1), \dots, z(k), \dots, z(k+N-1)] \tag{4}$$

Mixed Integer Quadratic Program (MIQP) represents a non-convex optimization problem with quadratic objective function and linear constraints. The non-convexity is caused by the fact that the optimized variables  $x_i$  belong to the binary set. The only difference between the convex QP and MIQP is the presence of binary variables  $x_i$  which makes system non-convex. Fortunately, for relaxed or fixed binary variables the problem becomes convex and can be solved using methods for convex optimization. In practice a constrained QP is usually solved either using an interior point method or an active set method.

Active set method searches for optimum solution by scanning the boundary of the problem's feasible region. It moves from a point on the boundary to another point that has better value of the objective function. This process is repeated until the optimal solution is found. With the interior point method the solution that is obtained at each iteration never lies on the boundary of the feasible region. Interior point methods solve the general inequality constrained problem by converting it to an equality constrained problem. Solution is reached by solving KKT (Karush-Kuhn-Tucker) conditions of this problem. There are two famous types of interior point methods; Barrier method and Primal-Dual interior point method. Due to higher efficiency, the primal-dual interior method is used in the paper for the solution of the following problem:

$$\begin{aligned}
 \min 0.5 x^T H x + f^T x \\
 x \in R^n : A_c x + b_c = 0 \\
 C_c x + d_c \geq 0
 \end{aligned} \tag{5}$$

where  $A_c$  is a  $m_{ec} \times n$  matrix describing the equality constraints and  $C_c$  is a  $m_{ic} \times n$  matrix describing the inequality constraints.  $b_c$  and  $d_c$  are  $m_{ec} \times n$  and  $m_{ic} \times n$  vectors respectively.

The primal-dual method uses some variations to Newton methods. Newton's method is used to solve the KKT conditions of the constrained QP problem. The concept of Newton's method is to calculate a direction that points toward improving the value of the objective function. The strategy is improved with methods like central path and path-following. These modifications improve the speed of convergence by allowing the algorithm to take large steps in the calculated search direction.

Using a brute force approach to find the optimal solution all of the QPs (LPs) that are associated to all the

feasible combinations of the discrete decision variables would have to be solved. The solution of MIQP would be the minimum of the solutions of all these problems. For  $n_i$  Boolean decision (input) variables, the number of possible QP (LP) problems is  $2^{n_i}$ .

Fortunately, efficient methods can be used for solution of this type of problem. Branch-and-bound has been the most used tool for solving large scale NP-hard combinatorial optimization problems since the branch-and-bound method is an order of magnitude faster than any of the other methods such as Generalized Benders Decomposition or Outer Approximation. The method is so fast due to the fact that the QP subproblems are easy to solve. For MATLAB, free software like YALMIP [13] can be used. At any time in the solution process, the status of the solution is given by a pool of yet unexplored subset of the solution space and by the best solution that has been found so far. The algorithm dynamically generates a search tree, which initially contains only one node called the root. At each iteration of a classical branch-and-bound algorithm only one node that represents unexplored subspaces is processed. The iteration consists of two main components: selection of the next node to process and strategy for branching. The nodes that created via branching strategy are stored together with the bound of the currently processed node. The search stops when the pool of unexplored subset is empty and then the optimal solution is the one recorded as "current best".

There are two basic node selection strategies for next node to proceed. The first one is best-first-search, where the next node is always the one with the lowest dual bound. However, this method has high storage requirements. The second type of node selection strategies is the depth-first-search where warm-starting can be successfully applied since the subproblems are similar and also number of unexplored nodes is low, so this method does not require a lot memory for nodes storage. Branching on a variable involves choosing the branching variable of the current optimal solution of the relaxed problem and then adding a constraint to it. The maximum fractional branching strategy is used in the solver and this method uses the variable with the highest fractional part for branching. The scheme of branch-and-bound method is depicted in Figure 1.

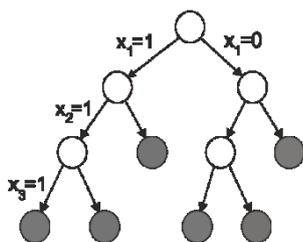


Figure 1. Branch-and-bound algorithm.

## 4 Relaxation of MIQP

In order to reduce the complexity of MIQP the constraints of binary variables can be relaxed, i.e. binary variables are allowed to span over the continuous interval  $[0,1]$  and the problem can be solved as a QP problem. In the receding horizon scheme only the first control action is applied so the later stages can be relaxed as result of trade-off of computation time for performance. For given prediction horizon  $H_p$ , after  $H_r$  ( $H_r < H_p$ ) steps the binary variables are relaxed to the continuous interval  $[0,1]$ .

## 5 Simulation results

The developed predictive controller was tested on a simulation example of a real batch reactor with the parameters from the paper [14]. The goal is to control the temperature of the ingredients stirred in the reactor core so that they synthesize into the final product. The main task is to control the temperature in the reactor as accurately as possible. A scheme of the batch reactor is shown in Figure 2. The reactor's core (temperature  $T$ ) is heated or cooled through the reactor's water jacket (temperature  $T_w$ ) in which there is a mixture of fresh input water, which enters the reactor through an on/off valve and reflux water. The dynamics of the system depend on the physical properties of the ingredients in the batch reactor's core. The temperature of the fresh input water  $T_{in}$  depends on two inputs: the positions of the on/off valves  $k_H$  and  $k_C$ . Only one of them can be open at a time. The ratio of fresh input water to reflux water is controlled by the mixing valve  $k_M$ . There are six possible ratios that can be set by the mixing valve. The share of fresh input water can be either 0, 0.01, 0.02, 0.05, 0.1 or 1. Therefore the system has two binary inputs, one input with six admissible values and two measurable outputs ( $T$  and  $T_w$ ). Due to the dynamics of the process the sampling period was set to 10s.

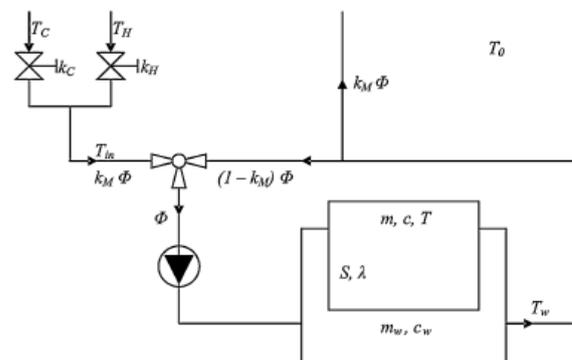


Figure 2. Scheme of batch reactor.

Hybrid linear model of the second order was developed in [15] by identification from the input-output data using the least-squares identification method. The model has the following form:

$$T_w(k+1) = \begin{cases} \Theta_1 [T_w(k) \ T(k) \ k_M(k) \ 1]^T & \text{if } k_C = 0 \\ \Theta_2 [T_w(k) \ T(k) \ k_M(k) \ 1]^T & \text{if } k_C = 1 \end{cases}$$

$$T(k+1) = 0.9967T(k) + 0.0033T_w(k) \quad (6)$$

$$\Theta_1 = [0.9429 \ 0.0395 \ 10.5969 \ 0.5273]$$

$$\Theta_2 = [0.9322 \ 0.0466 \ -7.8013 \ 0.5650]$$

The set of possible input variables  $[k_M \ k_H \ k_C]^T$  is defined in

$$M = \left\{ \begin{matrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.01 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.02 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0.05 \\ 0 \\ 1 \end{bmatrix}, \dots \\ \begin{bmatrix} 0.1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \end{matrix} \right\} \quad (7)$$

Using the HYSDEL Toolbox [16], problem (3) for the piece-wise affine system (6) is translated into a mixed integer quadratic program (MIQP), i.e., into the minimization of a quadratic cost function subject to linear constraints, where some of the variables are constrained to be binary or integer. The optimization problem is solved in the extended  $(x; u; z; \delta)$  space subjected to constraints, which increase the size of the problem dramatically.

The weighting matrices were defined as follows:

$$Q_1 = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$$

$$Q_2 = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 100 \end{bmatrix}, \quad (8)$$

$$Q_3 = \text{diag}([0 \ 100 \ 0 \ 100 \ 0 \ 100 \ 0 \ 100])$$

The weighting matrix  $Q_2$  weights the event of changing fresh input water from hot to cool. The value was set higher than  $Q_1$  to prevent changes of valves  $k_H$  and  $k_C$  but low enough to allow the on/off valves to change when a reference step occurs.  $Q_3$  weights the error between the predicted temperatures and reference signals. The highest values of weights are set on the reactors core temperature  $T$ .

The prediction of  $N = 4$  was chosen as control and prediction horizon for batch reactor. The resulting optimization problem has 4 discrete variables (positions of the valve  $k_M$ ), 4 binary variables (position of the valve  $k_H$ ) and 8 continuous variables (auxiliary variables  $z$ ) and 76 constraints. Either  $k_C$  or  $k_H$  can be open so the value of the  $k_C$  is computed as the negation of  $k_H$ . If all possible

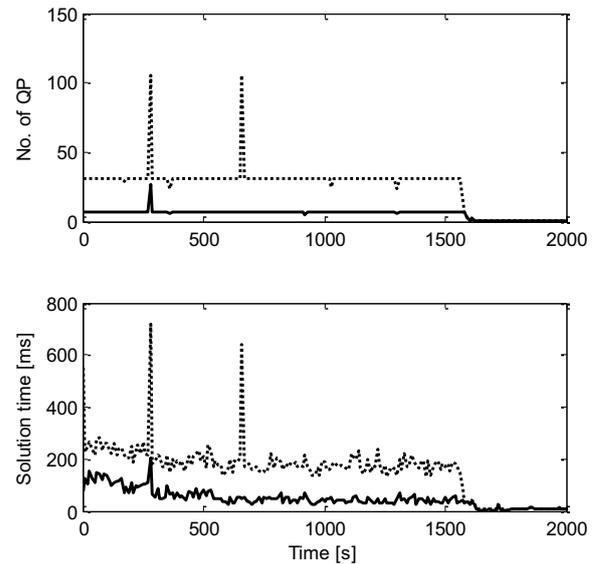
combinations of input signal were computed the solver would have to solve 20736 problems.

$$M = 6^4 2^4 = 20736 \quad (9)$$

from which only

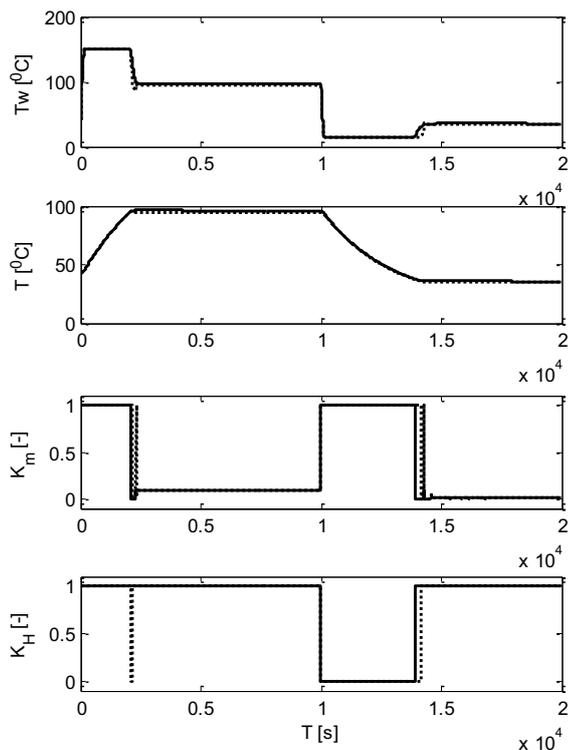
$$M = 7^4 = 2401 \quad (10)$$

are valid because only  $k_M = 1$  is allowed for cold water  $k_C=1$  as given in the value set (7). Increasing the horizons may increase the performance of the controller, but it increases the complexity of the mixed-integer programming procedure. The choice of prediction horizon offers reasonable values in terms of performance and speed. The optimization was performed in MATLAB using the solver we developed in [17] and extended from binary variables to integer variables. The optimization is computed in 200 ms to 700 ms using a computer with 3 GHz PENTIUM-4 processor and 8GB of RAM. In order to reduce the complexity of the problem only inputs related to the prediction horizon 2 steps are considered integer and binary and inputs related to the horizon steps 3 and 4 are considered continuous and in the range  $<0,1>$ . Thus the problem is reduced to an optimization problem with 2 discrete variables, 2 binary variables and 12 continuous variables is computed in less than 200 ms. The computation time are shown in Figure 3. For clarity only the first 2000s of the experiment are presented.



**Figure 3.** Optimization time and number of solved QP problems (original problem – solid line, reduced problem with relaxed variables – dotted line).

The initial value of the reference trajectory is set to 95 °C, so that the ingredients react at the optimal temperature. The control courses are presented on Figure 4. The similar control performance is obtained for control problem with horizon  $H_p=4$  and for control problem with horizon  $H_p=4$  but relaxed input variables from horizon  $H_r=3$ .



**Figure 4.** Control courses and input signals (relaxed problem – dotted line).

## Conclusion

The predictive control of a batch-reactor system using MLD approach was evaluated. The temperature can be optimally controlled using MLD approach within the sampling period; however the complexity of mixed-integer optimization that appears in optimal control of MLD systems limits the application to short prediction horizon. With the relaxed constraints the suboptimal solution in general can be found, but as can be seen in the case of the batch reactor, the performance degradation may be minimal while the necessary computation times are reduced significantly.

## Acknowledgment

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic within the National Sustainability Programme project No. LO1303 (MSMT-7778/2014).

## References

1. A. Bemporad, M. Morari, *Automatica* **35**, 407-427 (1999)
2. F. Borrelli, A. Bemporad, M. Fodor, D. Hrovat, *IEEE Transactions on Control Systems Technology* **14**, 541-552 (2006)

3. J. Causaa, G. Karer, A. Núñez, D. Sáez, I. Škrjanc, B. Zupančič, *Computers & Chemical Engineering* **32**, 3254-3263 (2008)
4. J. Hlava, R. Horalek, *Annals of DAAAM for 2009 & Proceedings of 20th DAAAM International Symposium Intelligent Manufacturing & Automation: Theory, Practice & Education* (2009)
5. M. Kögel, R. Findeisen, *Proceeding of the 50th IEEE Conference on Decision and Control and European Control Conference*, 501-506 (2011)
6. D. Frick, A. Domahidi, M. Morari, *Computers & Chemical Engineering* **72**, 21-33 (2015)
7. J. Novak, P. Chalupa, *Procedia Engineering* **100**, 1649-1656 (2015)
8. O. V. Volkovich, V. A. Roshchin, and I. V. Sergienko. *Cybernetics* **23**, 289 – 305 (1987)
9. R.T. Meyer, M. Žefran and R. A. DeCarlo, *IEEE Transactions on Control Systems Technology* **22**, 1784-1800 (2014)
10. T. Achterberg, *Mathematical Programming Computation* **1**, 1-41 (2009)
11. A. Bemporad, *Hybrid Toolbox - User's Guide*, 2004. <http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox>.
12. A. Bemporad, M. Morari, *Automatica* **35**, 407-427 (1999)
13. J. Lofberg, *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design*, 284-289 (2004)
14. G. Karer, I. Škrjanc, B. Zupancic, *Chemical Engineering and Processing* **47**, 2379-2385 (2008)
15. G. Karer, G. Music, I. Škrjanc, B. Zupancic, *Journal of Intelligent Robotic System* **20**, 297-319 (2007)
16. F.D. Torrisi, A. Bemporad, *IEEE Transactions on Control Systems Technology* **12**, 235-249 (2004)
17. J. Novak, P. Chalupa, *International Journal of Circuits, Systems and Signal Processing* **10**, 200-207 (2016)