Model predictive control of a pilot-scale distillation column using a programmable automation controller

Bart Huyck, Jos De Brabanter, Bart De Moor, Jan Van Impe and Filip Logist

***Abstract*— Following the evolution of growing computing power of industrial embedded devices, this paper investigates the use of an online Model Predictive Control (MPC) algorithm on an embedded Programmable Automation Controller (PAC). The controller is tested on a pilot-scale binary distillation column to track reference temperatures. A major task in each model predictive control algorithm is the solution of a quadratic problem. The Hildreth quadratic programming algorithm is used for this purpose. It turns out that this algorithm is able to control the set-up running on the PAC hardware.**

I. INTRODUCTION

In a world where economic and environmental issues become more and more important, efficient control systems have become indispensable. One of these control systems is Model Predictive Control (MPC). This control strategy, used since the 1970s in the chemical industry, has also become popular in other industry branches in the recent decade, e.g. the automotive industry. The increasing computing power of computers and embedded devices as well as algorithmic developments to speed up the solution to the Quadratic Problem (QP), the heart of a model predictive controller, encourage this evolution.

To quickly solve the QP, two main directions have been followed. One way to solve the QP problem fast is to solve the problem offline and employ a look-up table to find the solution online [2]. To encourage the use of this method, toolboxes and software were developed, e.g. [1]. This approach is often used on embedded devices which lack computing power, often in combination with the need for very fast MPC with update rates in the order of milliseconds. A different way is the use of fast online QP solving routines. For embedded devices, this approach is more recent. Again, to encourage the use of these algorithms they are integrated in dedicated software, code generators and toolboxes such as [4], [10].

As a consequence of the ever growing computing power of embedded devices such as Programmable Logic Controllers (PLCs) and Programmable Automation Controllers (PACs), manufacturers have included all kinds of additional features. Web- and FTP servers, Human-Machine Interaction (HMI), (web)panel control and data logging are examples. The increasing computing power also adds the possibility to run online QP solvers for small systems [7].

Bart Huyck, Filip Logist and Jan Van Impe are with BioTeC, Department of Chemical Engineering, Katholieke Universiteit Leuven, W. de Croylaan 46, B-3001 Leuven, Belgium. jan.vanimpe@cit.kuleuven.be

Bart Huyck, Jos De Brabanter and Bart De Moor are with ESAT-SCD, Department of Electrical Engineering, Katholieke Universiteit Leuven, kasteelpark Arenberg, B-3001 Leuven, Belgium

This means the need for additional computers to perform, e.g. data logging and supervision is no longer required. This evolution makes it possible to leave the often-used path to implement MPC controllers as it is done for large (chemical) installations, where the MPC controller and the underlying PID-controllers run on different (dedicated) machines [12]. For small and moderate installations, all controller levels can now be run on the same hardware. Hence, this paper investigates the use of a PAC system (National Instruments CompactRIO) to control a pilot-scale binary distillation column. Both the PI-controllers and the supervising online MPC controller run on the same device. Hence, before a model based controller can be used on a PAC, an accurate (but simple) process model has to be constructed. To this end, a black-box Multiple Input Multiple Output (MIMO) transfer function model is derived. The use of a linear transfer model is justified as they are commonly used in industrial MPC applications [12].

This paper is organized as follows. Section II describes the employed MPC formulation and the selected online QP solver. Section III discussed the implementation of the MPC controller on the PAC system. Next, in section IV, the pilot-scale binary distillation column is described together with (i) the model identification procedure for identifying the controller model and (ii) some additional model analysis. The results are written in Section V. Finally, Section VI summarizes the main conclusions.

II. MODEL PREDICTIVE CONTROL

Linear MPC is well known in literature [9], [3], [15] and the reader is invited to read these works for a detailed description. The basic formulation used to control the pilot-scale distillation column is briefly described below.

A. Model predictive control formulation

A linear, time invariant discrete-time system is described by:

$$\begin{aligned} \mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k), \end{aligned} \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$. Here m, n and p are the number of inputs, states and outputs, respectively. The objective of the controller is to find the optimal input for this system by means of minimizing a cost function:

$$J = \sum_{i=1}^{H_p} \|\hat{\mathbf{y}}(k+i|k) - \mathbf{y}_{\text{ref}}(k+i|k)\|^2_{W_y} + \sum_{j=0}^{H_c-1} \|\Delta\mathbf{u}(k+j|k)\|^2_{W_u} . \quad (2)$$

$\mathbf{y}_{ref}$ is the output reference and $\hat{\mathbf{y}}$ is the predicted output. The formulation $\mathbf{y}(k+i|k)$ represents the vector $\mathbf{y}$ on sample time $k+i$ at calculation time k. The change of the input is $\Delta\mathbf{u}(k+j|k) = \mathbf{u}(k+j|k) - \mathbf{u}(k+j-1|k)$. H_p and H_c (with $H_c \leq H_p$) are respectively the prediction and control horizon of the controller. $W_y \in \mathbb{R}^{p\times p}$ and $W_u \in \mathbb{R}^{m\times m}$ are positive definite weight matrices.

One of the key elements of MPC is the possibility to handle constraints. For this paper only input constraints are taken into account:

$$\mathbf{u}(j) \leq \mathbf{u}_{\text{Max}} \quad (3)$$
$$\mathbf{u}(j) \geq \mathbf{u}_{\text{Min}}.$$

Output constraints are omitted in the current study, but can easily be introduced. The optimization problem can be formulated as the minimization of Eq. (2), subject to Eq. (1) and (3). In order to solve this problem, the optimization problem is reformulated by elimination of the states in the form of a QP.

$$\min_{\theta} J = \frac{1}{2}\theta^T H\theta + g^T\theta \quad (4)$$
$$\text{subject to :} \quad P\theta \leq \alpha \quad (5)$$

with Eq. (4) the quadratic objective function, Eq. (5) the linear inequality constraints and θ the decision variables. To reformulate the problem, the following steps are taken. First, the state-space model Eq. (1) is rewritten in terms of the augmented state:

$$\xi = \begin{bmatrix} \Delta\mathbf{x} \\ \mathbf{y} \end{bmatrix} \quad (6)$$

with $\Delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{x}(k-1)$ and $\mathbf{y}$ the currently measured output [15]. The prediction over the prediction horizon is written in matrix formulation and is formulated as:

$$\hat{\mathbf{Y}} = F\xi + Q\Delta\mathbf{U}, \quad (7)$$

with $\hat{\mathbf{Y}}$ and $\Delta\mathbf{U}$ column matrices of predicted outputs and delta inputs, respectively. E.g. $\Delta\mathbf{U} \in \mathbb{R}^{nH_c\times 1}$ composed of $\Delta\mathbf{u}(k|k)$ to $\Delta\mathbf{u}(k+H_c-1|k)$. The matrices F and Q can be found in many works on MPC [9], [15]. The matrix Q is post-processed by including the weight matrices resulting in the hessian of the objective function in Eq. (4):

$$H = Q^T W_{y_{bd}} Q + W_{u_{bd}} \quad (8)$$

with $W_{y_{bd}}$ and $W_{u_{bd}}$ block diagonal matrices of W_y and W_u, respectively. To minimize the online calculation work, matrices that can be computed in advance are calculated offline. The hessian matrix H is one of the precomputed matrices that does not change during runtime. The gradient vector g from Eq. (4) has to be calculated online. It contains three parts depending on the current state and the reference of the in- and outputs. Thus g is calculated by:

$$g = G_1\xi - G_2\mathbf{Y}_{\text{ref}} - G_3\mathbf{U}_{\mathbf{ref}} \quad (9)$$

where G_1, G_2 and G_3 are gradient matrices. $\mathbf{Y}_{\text{ref}}$ is an $\mathbb{R}^{pH_c\times 1}$ matrix of the references $\mathbf{y}_{\text{ref}}(k|k)$ to $\mathbf{y}_{\text{ref}}(k+H_p|k)$. The matrices G_1, G_2 and G_3 are constant and are computed offline:

$$G_1 = H^T W_{y_{bd}}^T F \quad (10)$$
$$G_2 = H^T W_{y_{bd}}^T \quad (11)$$
$$G3 = W_{u_{bd}}^T \quad (12)$$

The constraints, i.e., the minimum and maximum admissible values for $\Delta\mathbf{U}$, are calculated online. $\Delta\mathbf{U}_{\text{Max}}$ is a column matrix of H_c times $\Delta\mathbf{u}_{\text{Max}} = \mathbf{u}_{\text{Max}} - \mathbf{u}(k-1)$. $\Delta\mathbf{U}_{\text{Min}}$ is a column matrix of H_c times $\Delta\mathbf{u}_{\text{Min}} = \mathbf{u}_{\text{Min}} - \mathbf{u}(k-1)$. Finally, with a lower triangular matrix of the appropriate identity matrices $C1$ to convert $\mathbf{u}$ to $\Delta\mathbf{u}$, the QP problem to be solved is:

$$\min_{\Delta\mathbf{U}} \quad \tfrac{1}{2}\Delta\mathbf{U}^T H\Delta\mathbf{U} + g\Delta\mathbf{U} \quad (13)$$
$$\text{subject to :} \quad \Delta\mathbf{U}_{\text{Min}} \leq C1\Delta\mathbf{U} \leq \Delta\mathbf{U}_{\text{Max}} \quad (14)$$

To solve this QP problem, the Hildreth quadratic algorithm [5] has been used.

B. The Hildreth QP algorithm

This algorithm has been selected for its easy implementation. The implementation as presented in [15] has been employed. One the advantages of this approach is that the solution of the QP is based on an element-by-element search, which can be easily implemented if no extended mathematical support is available on a device. As no matrix inversions are involved, the computations will not be interrupted, which is an advantage in real-time applications. In case of conflicting constraints, the algorithm will give a compromised, near-optimal solution. The main steps in this algorithm are presented in Algorithm 1. The maximum number of allowed iterations is 500 and the stop criterium is set to 10^{-8}. In contrast to explicit MPC strategies on Programmable Logic Controllers (PLCs) [8], [14], the use

```
Calculate the unconstrained solution
if unconstrained solution violate constraints then
    while maximum iterations not reached and solution
    not found do
        Solve one iteration of the QP
    end
    if maximum numbers of iterations is reached then
        Use calculated solution, but limited to the
        corresponding constraints
    end
end
```

Algorithm 1: Steps in the QP algorithm.

of online QP solvers on these devices can be considered [7]. Hence, an easy to implement algorithm is indispensable as the code has to be translated to an appropriate language following IEC 61131-3 which is used for PLCs. The Hildreth algorithm is a good candidate. More recent QP solvers [4], [10] are optimized extremely such that only code generators or off-the-shelf software is the best practice to implement them. Moreover, these recent solvers contain a lot of code and are often written in C/C++. Hence, a translation to a different language is required when these solvers are used on a PLC, which is a fault-sensitive and time consuming step. The experiments in this paper are set up to investigate the applicability of the classical Hildreth QP algorithm hosted by a PLC. As a step towards this goal, the algorithm is hosted on a PAC.

III. Controller implementation

The experimental set-up employed in this paper is a pilot-scale distillation column. The actuators and sensors are connected to a Compact Fieldpoint (National Instruments, Austin) with a cFP-2020 controller interface and I/O modules cFP-RTD-124, cFP-AIO-610, cFP-AIO-600 and cFP-AI-110. Due to the lack of computing power, this Compact Fieldpoint is connected to a powerful CompactRIO cRIO-9024 CPU (800 MHz, 512 MB) to run the controller programs. Two simultaneously running LabVIEW programs (National Instruments, Austin) have been written.

The first program, the *HMI program*, contains the PI-controllers, the human-machine interface, the data logging and the visualization. This program runs at a rate of 10 Hz. The second program, the *MPC program*, estimates the current state of the system by means of a Kalman filter. This estimated current state is passed, together with the current value of the top and reboiler temperature, the former estimated state and the former applied input to a library written in C++ calculating the gradient g and the upper and lower bounds for the QP. Next, the routine to solve the QP written in the same software library is called. The solution is returned to LabVIEW. After scaling, the inputs are passed to the HMI program where they are used as new set-points for the PI-controllers. The software library written in C++ is called via the *call library function* which is part of LabVIEW. The operating system of the cRIO is VxWorks 6.3. The compilation of the C++ code is done by gcc 3.4.4.

IV. Distillation Column Set-Up

A. Description

The experimental set-up involves a packed distillation column (see Fig. 1 and 2). The column is about 6 m high and has an internal diameter of 6 cm. The column works under atmospheric conditions and contains three sections of about 1.5 m with Sulzer CY packing (Sulzer, Winterthur) responsible for the separation. This packing has a contact surface of 700 m^2/m^3 and each meter packing is equivalent to 3 theoretical trays. The feed stream containing a mixture of methanol and isopropanol is fed into the column between packed sections 2 and 3. The temperature of the feed can be adjusted by an electric heater which can deliver heat up to a maximum of 250 W. At the bottom of the column a reboiler is present containing two electric heaters, each of maximum 3000 W. In the reboiler, a part of the liquid is vaporized while the rest is extracted as bottom stream. At the top of the column, a total condenser allows the condensation of the entire overhead vapor stream, which is then collected in a reflux drum. A part of the condensed liquid is fed back to the column as reflux, while the remainder leaves the column as the distillate stream. In this set-up the following

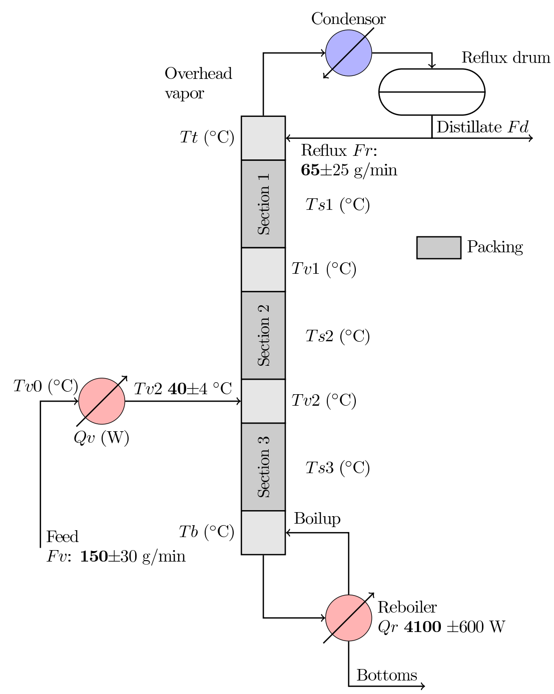


Fig. 1. Diagram of the pilot-scale distillation column. Nominal set-points are printed in bold and are followed by the maximum admissible deviations.

four variables can be manipulated: the reboiler duty Qr (W), the feed rate Fv (g/min), the duty of the feed heater Qv (W) and the reflux flow rate Fr (g/min). The distillate flow Fd (g/min) is adjusted to maintain a constant reflux drum level. Measurements are available for the reflux flow rate Fr, the distillate flow rate Fd, the feed flow rate Fv and nine temperatures: the temperature at the top of the column Tt, the temperatures in the center of every packing section (i.e. $Ts1$, $Ts2$ and $Ts3$, respectively), the temperature $Tv1$ between section 1 and 2, the temperature $Tv2$ between section 2 and 3, the temperature Tb in the reboiler of the column, and the temperatures of the feed before and after heating (i.e., $Tv0$ and $Tv2$, respectively). All temperatures are measured in degrees Celsius. There is no online measurement of the concentrations in the distillate and bottom stream. These can be measured off-line, e.g., based on a their refractive index using a refractometer. However, since the concentrations for the system under study can easily be inferred from the temperatures when temperature and pressure are known, only a controller for the temperatures has to be implemented.

$$\begin{bmatrix} Tt \\ Tb \end{bmatrix} = \begin{bmatrix} \frac{-2.26}{(1+2565s)(1+135s)} & \frac{0.53}{1+735s} & \frac{3.74}{(1+1803s)(1+78s)} & \frac{-3.45}{(1+2698s)(1+72s)}e^{-53.3s} \\ \frac{-2.13}{(1+1098s)}e^{-77.9s} & \frac{0.89}{1+1551s} & \frac{5.37}{2980s+1}e^{-37.1s} & \frac{-2.42}{(1270s+1)(545s+1)} \end{bmatrix} \begin{bmatrix} \mathrm{Fv} \\ \mathrm{Qv} \\ \mathrm{Qr} \\ \mathrm{Fr} \end{bmatrix} \tag{15}$$

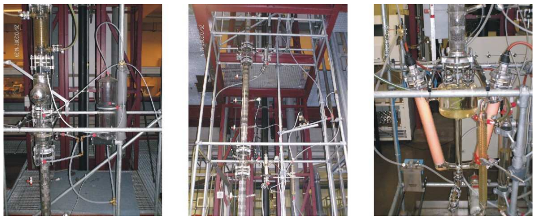


Fig. 2. Pictures of the pilot-scale distillation column: condenser (left), packed section and feed introduction (center), and reboiler (right).

B. Model Identification

For linear MPC, a linear process model is needed. To describe the behavior of the column, a model has to be created that links the 4 manipulated variables, i.e. the feed flow rate (Fv), feed duty ($Tv2$), the reboiler duty (Qr) and the reflux flow rate (Fr) with two temperatures, i.e. the top temperature (Tt) and the reboiler temperature (Tb). Although the concentration of the end product is actually a desired controller variable, this can easily be inferred from the temperatures when temperature and pressure are known. Hence, a *Multiple Input, Multiple Output* (MIMO) model that controls the reboiler and top temperature will be implemented (Fig. 3). More information on the excitation experiments that produced the data from which the model was derived, can be found in [6]. The linear black-box MIMO model was created based on a set of identified *Multiple Input, Single Output* (MISO) linear transfer models, identified with the Matlab System Identification Toolbox. The resulting MIMO model has been presented in Eq. (15). The time constants and delays are in seconds.

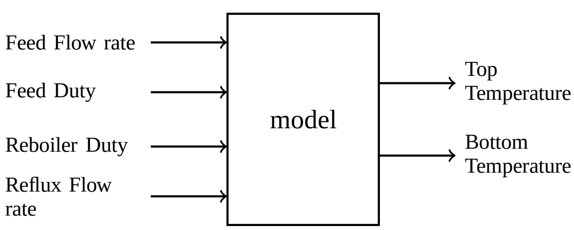


Fig. 3. Overview of the in- and outputs of the column model.

In order to be used in the model predictive controller, the model in Eq. (15) is converted to a discrete state-space model with a discretization interval of 1 minute. After model reduction, the resulting state-space model of order 13 is controllable, observable and stable.

The validation results presented in [6] have been demonstrated that the model for the top temperature is less accurate than the model for the reboiler temperature. Nevertheless, these models are believed to be of sufficient quality to be used in a model based controller.

C. Model analysis

To investigate the coupling between the input and output the Relative Gain Array (RGA) [11], [13] is used. This is a measure of the influence of a controlled variable, relative to other controlled variables. Elements close to one are the preferred control variables for that output. Elements close to zero are of less influence for the corresponding output. Input-output connections with negative elements indicate an inverse reaction and should be avoided if possible. The RGA From Eq. (15) is:

	Fv	Qv	Qr	Fr
Tt	0.3688	-0.0411	-1.0201	1.6923
Tb	-0.1792	0.0783	1.9547	-0.8537

This matrix indicates a strong influence of the reflux flow rate Fr on the top temperature Tt and an inverse gain for the reboiler power Qr. The reboiler temperature Tb is strongly determined by the reboiler power and in an inverse way by the reflux flow rate Fr. As stated above, negative elements must be avoided. Hence, when adapting the top temperature, only the reflux has to be manipulated and the reboiler power has to be kept constant. This is exactly the opposite for the adaptation of the reboiler temperature, which demonstrates that control is difficult. Manipulating one of the two temperatures will always disturb the other.

Both the feed flow rate and feed heating duty have a low impact as their corresponding values in the RGA are close to zero. The preferred inputs to manipulate both temperatures are Qr and Fr. Fv and Qv have a low impact on both the top and reboiler temperature and can only be used for small corrections.

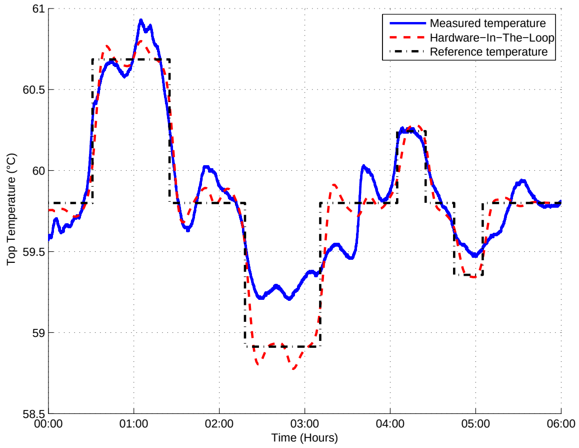
D. Model predictive control parameters

In the above formulation, the control horizon is set to 10 and the prediction horizon is 50. This leads to a hessian of size 40 in Eq. (13). The weight matrix W_y is a diagonal matrix with elements $W_{y_{11}} = 1$ and $W_{y_{22}} = 0.9$. This punishes each deviation from the top temperature reference slightly more than a deviation from the reboiler temperature. The weight matrix W_y has four elements on the diagonal $W_{u_{11}} = 0.8$, $W_{u_{22}} = 1$, $W_{u_{33}} = 0.8$ and $W_{u_{44}} = 1$ for the feed flow rate, the feed duty, reboiler duty and reflux flow rate, respectively. All non-diagonal elements are zero.

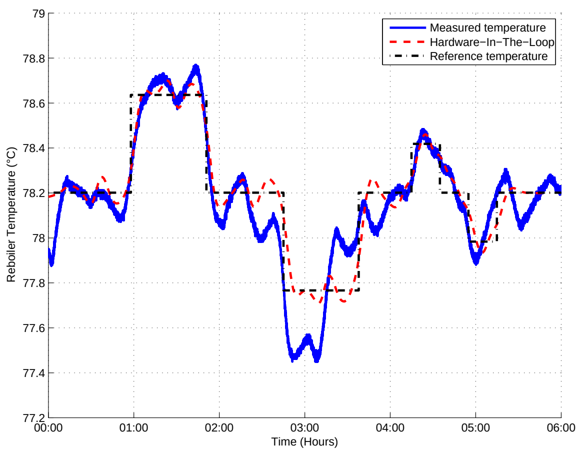
V. Results

A. Results on control

To make sure that the model is of sufficient quality and to have an idea of the expected measured temperature, Hardware-In-the-Loop (HIL) experiments are set up. Using the PAC device, the calculated controller inputs are supplied to the original non-reduced linear model of the column. The



(a) Top Temperature



(b) Reboiler Temperature

Fig. 4. Measured outputs for the top- and reboiler temperature for an experiment tracking a desired reference temperature profile with a MPC controller

result has been plotted on Fig. 4 and 5 for the controlled temperatures and inputs.

After successful HIL experiments, the experiment is performed on the column itself. Fig. 4 depicts the measured top and reboiler temperature during the experiment. Both controlled variables follow the same sequence of steps, but the ones for the reboiler temperature are delayed half an hour compared to the top temperature. The numerous set-point changes yield challenging reference paths to be tracked. Moreover, as the time constants of the different subsystems are longer than this half an hour, the system never reaches steady-state. These references are selected in order to combine the column's safety regulations with a sufficient number of different steps. As the aim of this experiment is to investigate the behavior of the QP solver, the actual quality of control is less important.

In the first two hours of the experiment, the temperatures are followed quite accurately. The shape of the steps up in the temperature references for both temperatures are clearly seen and the measured temperatures are close to the HIL experiments. The differences can be explained by the environmental conditions and difference between the actual

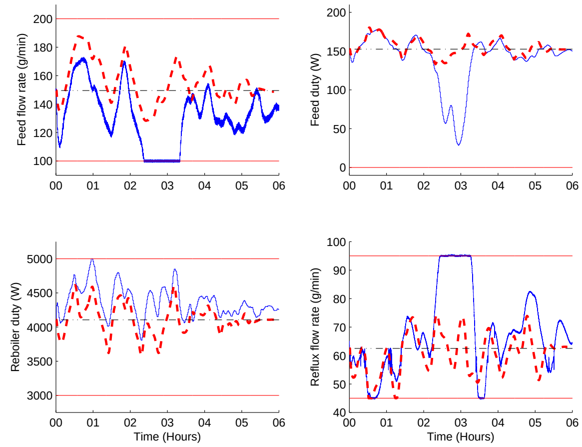


Fig. 5. Measured inputs for the experiment depicted in Fig 4.

distillation column and its linear model. The small deviations from the reference temperature halfway between each step are caused by the step change of the reference for the other temperature.

Between the second and fourth hour, both temperatures have to follow a step down, and return after one hour back to the starting temperature. At 2h15, the top temperature starts to decrease, but soon this decrease stops. The temperature stays constant until 2h45 and starts increasing slowly until 3h30. The measured temperature does not follow the reference nor resembles the HIL experiments. Similarly, the reboiler temperature decreases rapidly at the beginning of the step, but it reaches a temperature almost 0.3°C lower than its reference. As soon as the top temperature needs to step up at 3h10, the reboiler temperature evolves to a temperature nearly 0.2°C above the reference. Finally, at 4h, the reboiler temperature is back on the reference. Both references are not followed well, as both the constraints on the feed flow rate and reflux flow rate are reached (Fig. 5).

The last two hours of the experiment are a repetition of the first four hours, but slightly faster and with only half the step sizes. This time, the measured temperature resembles the HIL experiments better, as none of the constraints are reached. Fig. 5 depicts the four inputs of the distillation column. The two duties never reach their constraints. The feed duty hardly leaves its reference except around 3h00. Both flow rates touch the constraints.

It has to be noted that the difficult control of the temperatures, reflected by sometimes large deviations from the temperature reference, was predicted by the relative gain array. Moreover, as two of the four inputs hardly have any effect on the controlled temperatures and set-point changes succeed each other fast, the temperature control can not be perfect.

B. Results on implementation

More important for future implementations on PLC is the current behavior of the QP solver. The bottom plot in Fig. 6 depicts the number of iterations required to solve the QP problem for experiments on the set-up. The number of iterations is mostly one or zero iterations. No iterations

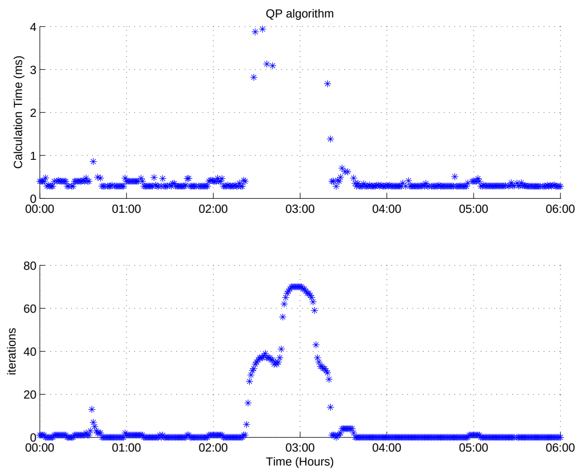


Fig. 6. Calculation time and number of iterations for the used QP algorithm

means that the solution is the unconstrained solution of the problem. The number of iterations increases to 13 at the first set-point change of the top temperature as the reboiler power reaches its constraint. Up to 70 iterations are needed between 2h30 and 3h30 when both flow rates reach their constraint. The maximum number of iterations is set to 500, thus the algorithm always presents the optimal solution during this experiment. In the upper plot of Fig. 6 the corresponding time is plotted. There is a relation between the time needed to solve the QP and the number of iterations. More iterations mean in general more required time, but there are a lot of exceptions. These exceptions are caused by the fact that the algorithm is paused when other tasks are performed on the PAC. Although the time needed to calculate new PI set-points is not identical for the same number of iterations, it is clear that a new solution is delivered within at most 3.5 ms for this experiment. The PAC has clearly sufficient computation power to run the MPC controller together with the PI-controllers, the data logging and the Human-Machine Interface.

For future use on PLCs, the memory consumption of the employed algorithm is interesting. The current MPC controller settings result in a hessian with size 40×40 elements. This matrix is need to be stored 8 times in memory for temporary storage to solve the QP. Together with some additional variables and without taking symmetry of the hessian into account, our implementation requires 13281 real variables to be allocated. This number of variables can easily be stored in memory as soon as the PLC is equipped with more than 64 kB of memory. Speed is not an issue on a PAC. However, it is impossible to predict the speed of the algorithm on a PLC. Additional research on a PLC is needed to investigate the actual speed of this algorithm on a PLC.

For the actual experiment, the required memory consumption was much higher. Together with the VxWorks operating system, the LabVIEW programs require almost 150 MB of the available working memory of the device. Based on the results shown in this paper, an MPC controller running in parallel with the PI-controllers and Human-Machine Interface for this pilot-scale distillation column is feasible.

VI. CONCLUSIONS

This paper has investigated the use of an online model predictive control algorithm on an embedded programmable automation controller for a pilot-scale binary distillation column to track reference temperatures. It has been demonstrated that it is possible to run an online MPC algorithm on standard industrial hardware which is not only able to run the supervisory algorithm, but is also responsible for data logging, Human-Machine Interfacing and low level control. This proves that a strategy where all procedures and computations needed to control and supervise processes that can be described by a moderate size process model (e.g., this pilot-scale distillation column) can be integrated in a single standard industrial embedded device.

ACKNOWLEDGMENT

Work supported in part by Katholieke Universiteit Leuven: OT/10/035, OPTEC Center-of-Excellence Optimization in Engineering (PFV/10/002), SCORES4CHEM (KP/09/005); by the Belgian Federal Science Policy Office: Belgian Program on Interuniversity Poles of Attraction; by the European Commission: Interreg IVa 2 Seas 07_022_BE_i-MOCCA. J.F. Van Impe holds the chair Safety Engineering sponsored by the Belgian chemistry and life sciences federation essenscia.

REFERENCES

[1] A. Bemporad. Hybrid Toolbox - User's Guide, 2004.
[2] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.
[3] E. F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2003.
[4] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.
[5] C. Hildreth. A quadratic programming procedure. *Naval Research Logistics Quarterly*, 4:79 – 85, 1957.
[6] B. Huyck, K. D. Brabanter, F. Logist, J. D. Brabanter, J. Van Impe, and B. De Moor. Identification of a pilot scale distillation column: A kernel based approach. In *18th World Congress of the International Federation of Automatic Control*, pages 471–476, Milano, Italy, August 28 - September 2 2011.
[7] B. Huyck, H. Ferreau, M. Diehl, J. De Brabanter, J. Van Impe, B. De Moor, and F. Logist. Towards online model predictive control on a programmable logic controller: practical considerations. *Mathematical Problems in Engineering*, vol. 2012, Article ID 912603, 20 pages, 2012. doi:10.1155/2012/912603
[8] M. Kvasnica, I. Rauova, and M. Fikar. Automatic code generation for real-time implementation of model predictive control. In *Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on*, pages 993–998, sept. 2010.
[9] J. Maciejowski. *Predictive Control With Constraints*. Pearson Education Limited, 2002.
[10] J. Mattingley and S. Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13:1–27, 2012.
[11] T. Mc Avoy, Y. Arkun, R. Chen, D. Robinson, and P. Schnelle. A new approach to defining a dynamic relative gain. *Con. Eng. Prac*, 11:907–914, 2002.
[12] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Con. Eng. Prac.*, 11:733–764, 2003.
[13] F. G. Shinskey. *Process Control Systems, Application, Design, and Tuning*. McGraw-Hill Publishing, New York, third edition, 1988.
[14] G. Valencia-Palomo and J. Rossiter. Efficient suboptimal parametric solutions to predictive control for PLC applications. *Control Engineering Practice*, 19:732–743, 2011.
[15] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer-Verlag London Limited, 2009.