European Symposium on Computer Aided Process Engineering – 15
L. Puigjaner and A. Espuña (Editors)

First Principles Model Based Control

Manuel Rodríguez[a*], David Pérez[a]
[a]Universidad Politécnica de Madrid
José Gutiérrez Abascal, 2. Madrid 28006, Spain

Abstract
Model Based Control is an important and widely used (mainly MPC) technique. This paper provides a new control architecture based on the use of physical models. In this preliminary work the architecture is applied for unconstrained multivariable control. It has been applied on several standard process units obtaining encouraging results. It has some advantages over MPC as it can use non linear rigorous models and it doesn't need any identification step.

Keywords: Model Based Control, Model Predictive Control, Internal Model Control.

1. Introduction

The interest in Model Predictive Control (MPC) started to increase after the presentation of IDCOM (Identification and Command) (Richalet, 1978) and DMC (Dynamic Matrix Control) (Cutler, 1979). After 25 years MPC has become a widely used technology in process control. Nowadays, a new crude distillation unit in a refinery is not conceived with other control scheme but MPC (and the same happens in many other processes). The technology applied is usually based on a previous identification step (which is of the most importance) to get a linear model of the unit and, then, on an implementation step (usually more simple and less time consuming). Although a lot of research has been done regarding Nonlinear MPC using different approaches, differential equations, neural nets (Temeng, 1995), Hammerstein models (Fruzzetti, 1997), Volterra equations (Maner, 1996), fuzzy models (Sousa, 1997) … it is still an open area where many problems arise.

The purpose of this work is the use of first principles models for model predictive control. To achieve this goal, a new architecture has been developed and tested on some simulated process units.

The remaining paper is organised as follows: Section two describes in detail the new architecture, how it works and its components, and explains the software implementation. Section three shows the results obtained when it is applied to control some operation units. Finally, section four presents the conclusions and future steps of this work.

* Author to whom correspondence should be addressed: mrod@diquima.upm.es

2. Physical Model Based Control Architecture.

The PMBC architecture is composed of two modules: the Physical Model Based (PMB) Controller Module and the PMB Model Module. Figure 1 shows the proposed architecture:

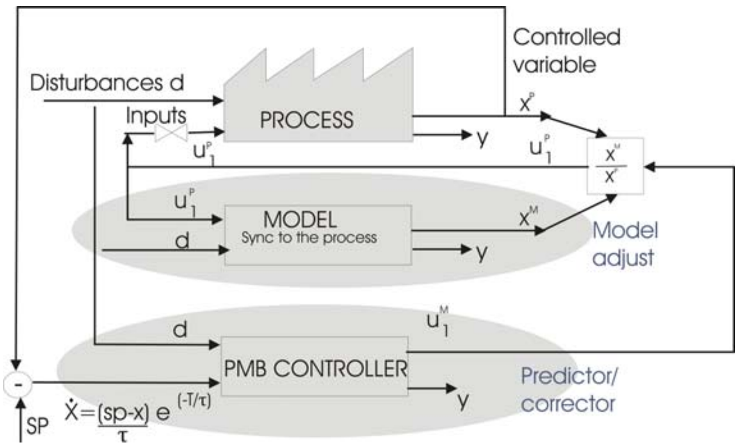

Figure 1. PMBC Architecture

The models used in the components of the architecture have been developed with some degree of error with respect to the process, as it is impossible to get a perfect model of a real process. To this purpose, some parameters have been slightly changed.

2.1 PMB Controller Module

This module is the core of the architecture, it predicts the values of the manipulated variables that provide the desired performance (set point change or disturbance rejection).

This module has a model of the unit to control. It has a first principles model comprised of a set of DAEs. This module has the following variables as inputs:

- The error of the controlled variables, it can be referred to the controlled variable directly or to its derivative. It is the desired response or the target curve.
- Measured disturbances.

The outputs of this module are the set of manipulated variables (which are commonly inputs in a model of the process).

If the controlled variable is at its set point, the input of the module will be the derivative set to zero or the value set to that constant (so it remains in that value). If a disturbance happens, the controller will produce an output to compensate that disturbance and keep the controlled variable at the desired set point value. If a set point change is desired, the input will be an exponential (softened step) in the controlled variable. If the controlled variable acts like an integrator, such as controlling the level with an output stream, a ramp has to be used instead of a step. The time used for the step or ramp is an adjustable parameter.

The time scale of this module is the simulation execution time of the model in the selected platform. It is an adjustable parameter, it can be slowed down to avoid excessive control actions that sometimes will provide a worse system performance.

2.2 PMB Corrector Module

This module has a model of the unit to be controlled. The model is exactly the same as above except for which are the input variables to it. This model is used in the standard way, so the inputs to it are the disturbances and the inputs to the real process. The outputs of the model are the controlled and output variables of the process.

The purpose of this module is to correct, in some way, the prediction made by the Controller Module. As no model is perfect, the controller output will lead to a control action that doesn't set the controlled variable to the desired value. This means that some feedback corrections are necessary. In order to accelerate this procedure, this module compares its output with the actual process output and sets a correction factor to be applied to the control action. So, this module has two components: the model component and the comparison component. The final output of this module, with the correction factor, is the control action to be applied to the real process.

The time scale of this module has to be that of the real process, so it has to be synchronized to it as accurate as possible, in order to be able to compare the same variables.

2.3. Software implementation

The implementation has three components, the Controller Module, the Corrector Module and the Process Module. All the components have been implemented in the same machine, a PC running under Linux OS. The model used in the Controller and Corrector Modules is not perfect, some changes have been made in several parameters of them in order to check the proposed architecture.

The models have been developed using the ABACUSS II simulator (Barton,2003). This software allows to embed the simulation code in other application. Using the C++ programming language, different executables have been generated for every component of the architecture. The information flow has been implemented through shared memory procedures.

3. Applications

In this section the results of the application of this architecture are presented. Set point and load changes are applied to all the tested units and the performance of the controller is evaluated.

3.1. Stirred tank heater

This unit is a perfectly stirred tank heater with a jacket. It has two input streams and one output stream. There are level and temperature control loops. The manipulated variables are the tank input flow and the jacket input flow. These two variables are calculated in the Controller Module. This module has as inputs the heater temperature and the tank level. To achieve a good control on the tank, the following control equations are used:

$$\frac{dH}{dt} = \left(SPH - PH\right)\times\frac{e^{-\frac{t}{t}}}{t} \qquad (1)$$

SPH: Level set - point, H: level of the model, PH: level of the actual process.

$$\frac{dT_O}{dt} = K \times (SPT - PT) \times \frac{e^{-\frac{t}{t}}}{t} \quad (2)$$

SPT: Temperature set - point, To: jacket temperature of the model, Pt: temperature of the actual process.

The jacket temperature is used as the forcing equation of control instead of the tank temperature because the use of the latter poses a 2 - index problem. Due to this problem, the gain between the two variables needs to be added to the control equation. In this model some parameters are changed about 5% with respect to the actual process.
Figure 2 shows the control of the unit in the presence of a flow disturbance and a level set point change.

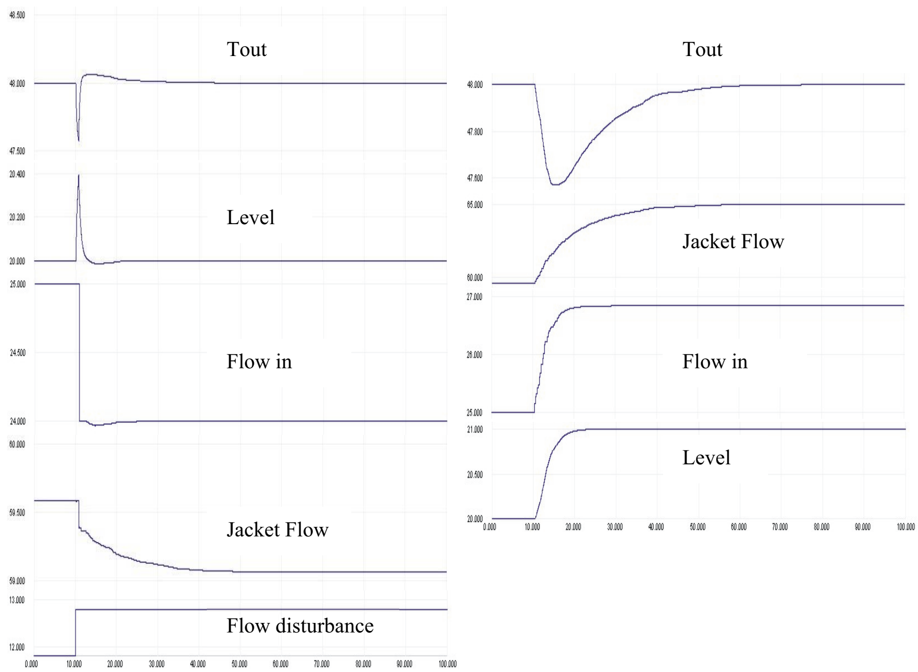

Figure 2. Heater response after a flow disturbance on the left. Heater response after a level set point change on the right.

In these tests both loops are affected as the temperature changes with a change in any of the stream flows.

3.2. Continuous Stirred Tank Reactor (CSTR)

This unit has the same equations as the heater but a first order reaction, following the Arrhenius expression, is added. In this case, the same two loops (level and temperature) are controlled. But, in this case, the level is controlled with the output stream, which means that the control action has to be changed. An exponential (step) cannot be used and a ramp is used instead. The following equation shows the level control action implemented:

$$\frac{dH}{dt} = \frac{SPH - PH}{t} \qquad (3)$$

The temperature control equation is the same as in the heater application. Figure 3 shows the performance of the system in the presence of concentration and temperature disturbances. In this model, the parameter that is changed with respect to the actual process is the Arrhenius pre-exponential constant in 5%.

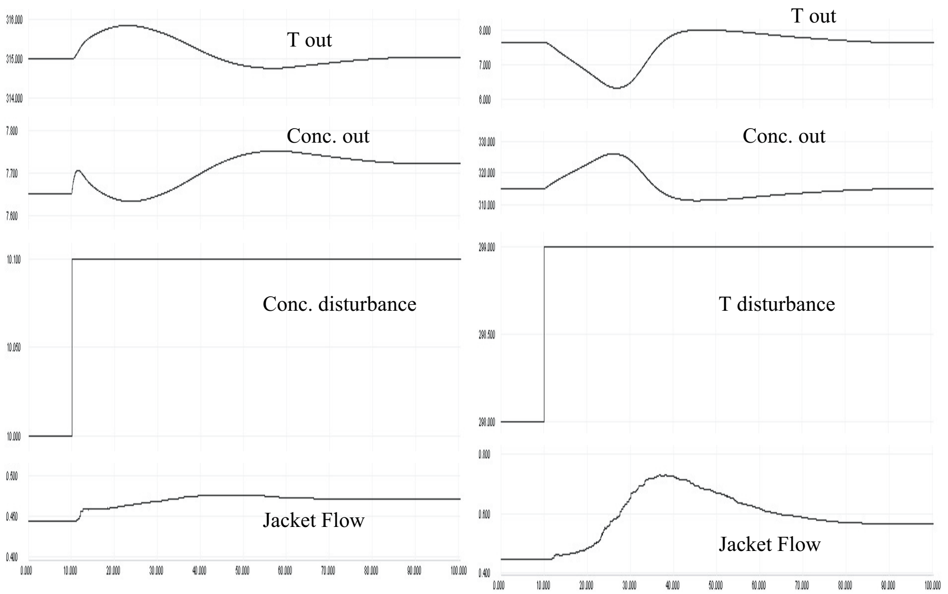

Figure 3. CSTR response after a concentration disturbance on the left. CSTR response after a temperature disturbance on the right.

3.3. Distillation Column

This unit is a binary distillation unit. It assumes equimolal overflow, 20 theoretical trays, simple liquid tray hydraulics and constant relative volatility (ideal mixtures) for the vapour - liquid equilibrium in every stage. Feed enters above 10th tray (feed stage). Total overhead condenser and partial reboiler.

The implementation of this unit for a continuous test is not finished at the time of this paper. The only implementation available is just the first control action to be applied to the process. The model in this initial implementation is assumed perfect (this means that just a single control action gets the controlled variable to the desired value). The control equations used are similar to the ones presented in the previous applications. Next figure shows that this control architecture can be applied to the distillation column as well.

4. Conclusions

This paper has presented a new architecture for model based control. Regarding the obtained results, this physical model based control seems to be a usable technology for

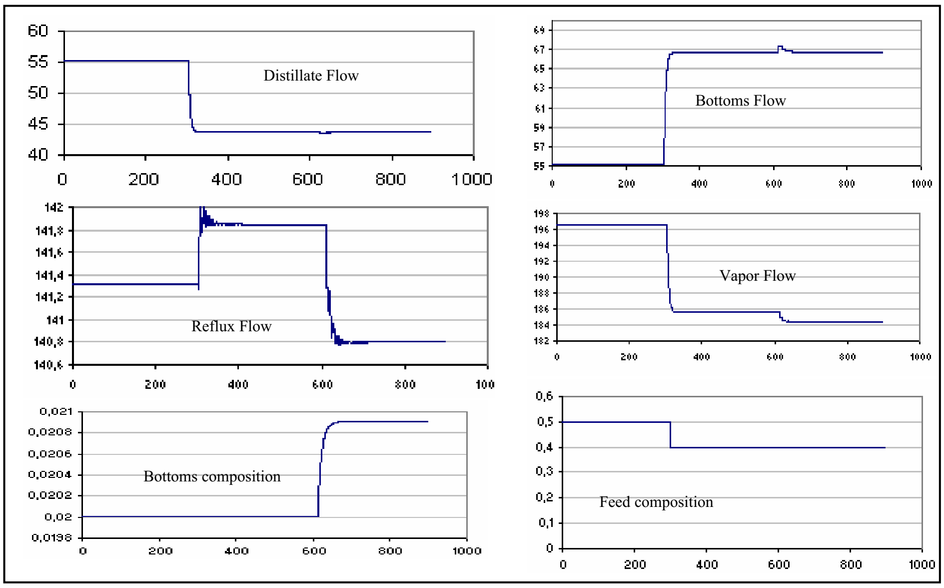

Figure 4. Distillation column performance with a composition disturbance and a bottoms composition set point change.

any process unit. This new approach has some clear advantages over the ones used currently in the industry. It uses a first-principles model that is suited to any operating region, so, it is a non-linear MPC. It doesn't need any identification step which is very resource consuming for any process unit.

Although the results presented are promising, a lot of work still has to be done to establish the availability of this technology. First of all, the initial approach taken is for unconstrained predictive control, additional steps taking into account constraints (like valve range,…) have to be made. New issues have to be studied as dead-time, stability,…The next step in this work will be to test this architecture with a real system.

References

Barton, P.I, 2003 http://yoric.mit.edu/abacuss

Cutler, C. R. and Ramaker B. L. ,1979. Dynamic matrix control—a computer control algorithm. AIChE 86th National Meeting. Houston,TX.

Fruzzetti et al., 1997. Nonlinear control using Hammerstein models. J. Proc. Control, 7, n1, 31-4

Maner et al, 1996. Nonlinear model predictive control of a simulated multivariable polymerisation reactor using second order Volterra models.

Richalet, J., Rault, A., Testud, J. L., and Papon, J. ,1978. Model predictive heuristic control: applications to industrial processes. Automatica, 14(5), 413–428.

Sousa et al, 1997. Fuzzy predictive control applied to an air conditioning system. Control Engineering Practice, 5. n10, 1395-1406.

Temeg et al, 1995. Model predictive control of an industrial packed bed reactor using neural networks. J. Proc. Control, 5, n1, 19-27

Acknowledgements

The project is sponsored by Repsol-YPF foundation.