ELSEVIER
IFAC
PUBLICATIONS

ITERATIVE NONLINEAR MODEL PREDICTIVE CONTROL OF A PH REACTOR. A COMPARATIVE ANALYSIS

J.R. Cueli and C. Bordons [1]

Escuela Superior de Ingenieros. University of Seville.
Camino de los Descubrimientos, s/n. 41092, Seville, Spain
cueli@cartuja.us.es,bordons@esi.us.es

Abstract: This paper describes the control of a batch pH reactor by a nonlinear predictive controller that improves performance by using data of past batches. The control strategy combines the feedback features of a nonlinear predictive controller with the learning capabilities of run-to-run control.
The inclusion of real-time data collected during the on-going batch run in addition to those from the past runs make the control strategy capable not only of eliminating repeated errors but also of responding to new disturbances that occur during the run. The paper uses these ideas to devise an integrated controller that increases the capabilities of Nonlinear Model Predictive Control (NMPC) with batch-wise learning. This controller tries to improve existing strategies by the use of a nonlinear controller devised along the last-run trajectory as well as by the inclusion of filters.
A comparison with a similar controller based upon a linear model is performed. Simulation results are presented in order to illustrate performance improvements that can be achieved by the new method over the conventional iterative controllers. Although the controller is designed for discrete-time systems, it can be applied to stable continuous plants after discretization. *Copyright© 2005 IFAC.*

Keywords: Learning algorithms, pH control, predictive control.

1. INTRODUCTION

Nonlinear chemical plants have been many often controlled by means of nonlinear feedback controllers, even for batch mode of operation. Given the nonlinear and time-varying characteristics of batch plants, an acceptable tracking of set point profiles can be only achieved with advanced nonlinear controllers because the process is never in steady-state regime. In this point, iterative controllers have several advantages over classical ones, mainly perfect tracking capabilities, as well as trajectory improvement at every run of the process.

A good example of strongly nonlinear process suitable to be controlled by iterative controllers is a pH batch fermentation process. pH control has been extensively studied in literature using non linear feedback controllers. Wright and Kravaris (2001) propose an adaptive strategy for waste-water treatment plants that uses a simplified dynamic model. A self-tuning controller is described in Babuska *et al.* (2002). Adaptive nonlinear control strategies (Henson and Seborg, 1994), model-

---

based (Sing and Postlethwaite, 1997) and fuzzy logic controllers (Garrido *et al.*, 1997) have also been used.

However, feedback controllers have only limited effectiveness in handling periodic reference trajectories and disturbances. This is due to the fact that there is no mechanism for passing feedback error information from one run to another, and henceforth, *learning* based on previous runs cannot take place. In batch mode of operation, batch-to-batch variations can be significant and are of primary concern. In most industrial cases, the batch-to-batch variations are strongly autocorrelated, providing the possibility of using previous batch results to adjust the recipe of a subsequent batch. The error that cannot be removed by on-line feedback control can be eliminated or reduced by the so called *batch-to-batch* or *run-to-run* control.

This can be done by means of Iterative Learning Control (ILC) and Repetitive Control (RC), which refer to a body of methodologies that attempt to improve the control performance of a repeated run based on the results from previous runs. The two differ in that ILC deals with systems that reset the state at the start of each new run, while RC addresses those with continuous state transition from one run to the next. The original objectives behind the development of ILC and RC were different. RC was developed as a way to cancel periodic disturbances and/or tracking periodic reference trajectories in a continuous system, whereas ILC emerged as means to achieve run-to-run improvements in robotics and servomechanical systems. In the development of the RC methods, the internal model principle has played a major role (Hara *et al.*, 1988), while ILC designs have mostly followed the direction of successive model inversion (Moore, 1998). In fact, both control concepts share the same basic principles and underlying issues (Longman, 1997).

Given the advantages of the batch-to-batch and feedback control strategies, it is natural to explore the possibility of combining them. Because feedback control can respond to disturbances immediately and batch-to-batch control can correct any bias left uncorrected by the feedback controller, which may be due to unmodelled disturbances, parameter errors, and dynamics, the combined scheme can potentially complement each other to render the benefits of both. The idea of combining batch-to-batch control with feedback control has appeared in Lee *et al.* (1999).

Since the objective is to eliminate persisting errors from previous runs as well as to reject new disturbances as they occur during the current run, the combination of these two techniques can be a good solution. Notice that the problem is not easy to solve since the feedback control of the on-going run is a difficult problem itself, because it involves a nonlinear controller. The inclusion of real-time data collected during the on-going batch run (in addition to those from the past runs) makes the feedback control strategy capable of responding to new disturbances that occur during the run.

The paper uses these ideas to develop an integrated controller that increases the capabilities of NMPC with batch-wise learning. This controller tries to improve existing strategies by the use of a nonlinear controller devised along the last-run trajectory as well as by the inclusion of filters. Tracking the setpoint profile is tackled by a nonlinear controller based on EPSAC (De Keyser, 1997) while its iterative nature improves the performance at each batch.

The paper is organized as follows. In section 2 a description of the pH control batch process is presented. Section 3 describes the proposed algorithm, showing the development of the control strategy for the plant model. This strategy is tested on a nonlinear simulator, comparing it with other iterative controllers, and the results are shown in section 4. Finally the major conclusions are drawn in section 5.

2. PROCESS DESCRIPTION

The control of pH (McMillan, 1984) is common in chemical and biotechnological industries. Examples of this kind of plants can be found in wastewater treatment plants, the production of pharmaceutical products and fermentation processes. Controlling the pH value of these processes is difficult due to the highly nonlinear response of the pH to the addition of acid or base and strong disturbances appearing in the process. These processes can exhibit severe static nonlinear behavior because the gain can vary several orders of magnitude for a slight range of pH values.

This example corresponds to a laboratory fermentation process taken from the literature. The pH value ranks as one of the most important factors that influence a fermentation process. A pH value out of its optimum often inhibits the growth of the essential micro-organisms, alters the bacterial population and inhibits the desirable enzymatic activities. The result is a delay in the fermentation process or even the death of the micro-organisms. The controller must achieve the prescribed accuracy (in some cases around 0.05 pH unit) despite the severe process nonlinearities.

The process is shown in Figure 1 and consists of a tank where three streams are mixed:

- an acid ($H\,Cl$) stream (q_1),

Fig. 1. pH process

Fig. 2. Titration curve

- a buffer ($K H_2 P O_4$ and $Na_2 H P O_4$) stream (q_2), and
- a base ($Na O H$) stream (q_3).

The process output is the pH value of the solution which can be controlled acting on the valves.

A continuous model can be obtained by first principles, assuming perfect mixing, constant density, and complete solubility of the ions. This model can be described by the following state-space equations (see Babuska *et al.* (2002) for details and parameter values):

$$Ah\frac{dW_{a_4}}{dt} = q_1 (W_{a_1} - W_{a_4}) + q_2 (W_{a_2} - W_{a_4})$$
$$+ q_3 (W_{a_3} - W_{a_4}) \qquad (1)$$

$$Ah\frac{dW_{b_4}}{dt} = q_1 (W_{b_1} - W_{b_4}) + q_2 (W_{b_2} - W_{b_4}) +$$
$$+ q_3 (W_{b_3} - W_{b_4}) \qquad (2)$$

$$\frac{dh}{dt} = \frac{q_1 + q_2 + q_3 - q_4}{A} \qquad (3)$$

The output function is given implicitly by

$$W_{b_4}\frac{K_{a_4}x + 2K_{a_4}K_{a_5}x^2 + 3K_{a_4}K_{a_5}K_{a_6}x^3}{1 + K_{a_4}x + K_{a_4}K_{a_5}x^2 + K_{a_4}K_{a_5}K_{a_6}x^3} +$$
$$+ W_{a_4} + K_w x - x^{-1} = 0 \qquad (4)$$

and

$$pH = -\log\left[H_3O^+\right] = \log x \qquad (5)$$

The static Equations (4)-(5) make the process highly nonlinear and can be used to obtain the theoretical pH curve of the buffer solution for a changing volume of acid or base (titration curve). This curve is shown in Figure 2 and is obtained by integrating Equations (1)-(5) with zero initial conditions (initial states are obtained assuming $q_3 = u = 0$). The figure shows the static nonlinear pH characteristic of a 1.25 l phosphoric acid buffering solution to the addition of base and acid. It is clearly shown how the slope of the pH curve presents high variations along the curve.

3. CONTROLLER SYNTHESIS

The idea of the controller is to combine iterative and model predictive control. Mainly, this is achieved by using last batch trajectory in the construction of an approximated linear time-varying model. Also, batch deviation variables are used. If the batch index is denoted by the superindex k, they can be specified in the following form:

$$\tilde{x}^k(t) = x^k(t) - x^{k-1}(t) \qquad (6)$$

Finite duration linear time-varying batch process can be described completely in the following simplified matrix form:

$$\tilde{y} = G\tilde{u} \qquad (7)$$

where $\tilde{y}$ is the process output and $\tilde{u}$ the input. The use of batch deviation variables eliminates the bias term that should appear in equation (7). This means that modelling and control tasks are also simplified because the bias term, which includes batch-correlated disturbances, does not need to be identified or compensated. Deviation variables will tend to zero when the system converges to the reference trajectory.

Fig. 3. Linearized system around a given base trajectory at time t

Therefore, the first task to build the controller is to construct the linear time-varying (LTV) model using deviation variables. The linearization should be made around a base trajectory (for details, see De Keyser (1997)). For batch processes, it is equal to the one followed by the system in last batch. Later, the system is discretized using an adequate method, denoting $x_n = x(T_s n)$, being

T_s the sampling time. A LTV system is obtained in the form of equation (7).

In this case there exists a well-known first principles model (equations 1-5). In other cases, an empirical identification of the process should be realized. Hammerstein models can also be used to model the process (Camacho and Bordons, 2004).

It is implicitly assumed that the deviation variables are small, so the linearized system is accurate. This is assured making the control variations slow. Second order Volterra models could be used for control (Bordons and Dorado, 2002), and they should improve speed of convergence (in number of batches) because model approximation to the real one is better.

3.1 pH discrete-time model

The model given by equations (1)-(5) is linearized and discretized using sampling time equal to $T_s = 1s$. Resulting equations are non observable, and the states (W_{a_4} and W_{b_4}) are not measurable. It makes the control task more difficult, because the states have to be known in order to get the linearized model. Henson and Seborg (1994) propose an open loop estimation technique for these chemical invariants (W_{a_4} and W_{b_4}).

The state-space linearized model has the form:

$$x_{k+1} = A_k x_k + B_k u_k \qquad (8)$$
$$y_k = C_k x_k \qquad (9)$$

and the following expressions for matrices A_k and B_k are obtained:

$$A_k = \begin{pmatrix} a & 0 & a_{13} \\ 0 & a & a_{23} \\ 0 & 0 & 1 \end{pmatrix} \qquad (10)$$

with

$$a = 1 - T \frac{\sum_{i=1}^{3} q_i W_{a_i}}{Ah} \qquad (11)$$

$$a_{13} = T \frac{q_4 W_{a_4} - \sum_{i=1}^{3} q_i W_{a_i}}{Ah^2} \qquad (12)$$

$$a_{23} = T \frac{q_4 W_{b_4} - \sum_{i=1}^{3} q_i W_{a_i}}{Ah^2} \qquad (13)$$

and

$$B_k = \left(T \frac{W_{a_3} - W_{a_4}}{Ah}, \ T \frac{W_{b_3} - W_{b_4}}{Ah}, \ T \frac{1}{A} \right)^t \qquad (14)$$

As the output is given as an implicit function, the implicit function theorem must be used to compute matrix C_k (see Henson and Seborg (1994)).

Differentiating the function $F(W_{a_4}, W_{b_4}, x) = 0$ (equation (4)) and expression (5), we have:

$$dF = \frac{\partial F}{\partial W_{a_4}} dW_{a_4} + \frac{\partial F}{\partial W_{b_4}} dW_{b_4} + \frac{\partial F}{\partial x} dx = 0 \quad (15)$$

$$\frac{d(pH(x))}{dx} = \frac{d}{dx} \log_{10} x = \frac{1}{x \ln 10} \qquad (16)$$

If $F_x \neq 0$ the implicit function theorem applies here and the elements of the matrix C_k are obtained. These elements are the partial derivatives of the output with respect to the states W_{a_4} and W_{b_4}. Note that $C_{1,3} = 0$ because the pH does not depend of the volume of the solution. Then:

$$\frac{\partial pH(W_{a_4}, W_{b_4})}{\partial W_{i_4}} = \frac{\partial pH}{\partial x} \frac{\partial x}{\partial W_{i_4}} = -\frac{1}{x \ln 10} \frac{F_{W_{i_4}}}{F_x} \qquad (17)$$

where i must be substituted with a or b and

$$F_{W_{i_4}} = \left. \frac{\partial}{\partial W_{i_4}} \right|_{\left(W_{a_4}^*, W_{b_4}^*, x^*\right)} F(W_{a_4}, W_{b_4}, x) \quad (18)$$

Matrix C_k is given by

$$C_k = \left(-\frac{1}{x \ln 10} \frac{1}{F_x}, \ -\frac{1}{x \ln 10} \frac{F_{W_{b_4}}}{F_x}, \ 0 \right) \quad (19)$$

with

$$F_{W_{b_4}} = \frac{k_{a_4} x + 2 k_{a_4} k_{a_5} x^2 + 3 k_{a_4} k_{a_5} k_{a_6} x^3}{1 + k_{a_4} x + k_{a_4} k_{a_5} x^2 + k_{a_4} k_{a_5} k_{a_6} x^3} \qquad (20)$$

$$F_x = k_w + \frac{1}{x^2}$$
$$- \frac{W_{b_4}}{x} \left(\frac{k_{a_4} x + 2 k_{a_4} k_{a_5} x^2 + 3 k_{a_4} k_{a_5} k_{a_6} x^3}{1 + k_{a_4} x + k_{a_4} k_{a_5} x^2 + k_{a_4} k_{a_5} k_{a_6} x^3} \right)^2$$
$$+ W_{b_4} \frac{k_{a_4} + 4 k_{a_4} k_{a_5} x + 9 k_{a_4} k_{a_5} k_{a_6} x^2}{1 + k_{a_4} x + k_{a_4} k_{a_5} x^2 + k_{a_4} k_{a_5} k_{a_6} x^3} \quad (21)$$

3.2 Dealing with uncertainties

Filtering is used in order to enhance closed loop performance in the presence of noise. Strong nonlinearities and large levels of measurement noise are many often present in pH plants, making the use of robustness filters desirable. In batch processes, two kind of filters are applicable. Both are described in the following lines.

The first one is the well-known median filter. The controller uses the pair $\left(u^{k-1}(t), y^{k-1}(t)\right)$ as base trajectory, so the variance of the system variables could tend to increase as the batch number grows, even more when noise and batch disturbances are present. Sometimes this problem occurs in controllers that use past batches information. Performance can be improved by using a median filter in order to keep the variables smooth enough, as it is pointed out by Mezghani *et al.* (2002). The filter is defined by

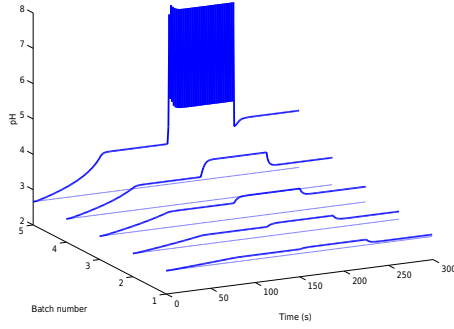$$y(t) = \frac{1}{2l + 1 - r - s} \sum_{j=-l+r}^{l-s} u(t+j) \qquad (22)$$

Fig. 4. Several batch simulation with ILC

where u is a vector with $2l + 1$ sorted elements of the filtered variable v around time t. The $r + s$ extreme values are discarded and the mean is computed.

Other possibility of filtering is the Exponential Weighting Moving Average (EWMA) filter. This filtering, used in industrial batch processes (Moyne *et al.*, 2001), can estimate parameters, reduce system noise or compensate drift terms. The result is that the noise variance in the closed loop system is lower. This filter can be viewed as a simple signal estimator for batch processes. The expression for the filtered output variable is

$$y^k(t) = \lambda_f u^k(t) + (1 - \lambda_f) y^{k-1}(t) \qquad (23)$$

where λ_f is a parameter.

4. RESULTS

In this section a comparison between nonlinear INMPC and linear iterative controllers (BMPC and ILC) is performed. Also an analysis of the controlled system in the presence of disturbances (a 10% step in inlet acid flow at time $t = 225s$) and noise is presented. The results are obtained in simulation, and the controller has been tested with the continuous time nonlinear plant model.

The first iterative controller is ILC. Its control law is analogous to the proportional controller, and it is determined by a constant gain K and the plant delay d:

$$u_t^{k+1} = u_t^k + K e_{t+d}^k \qquad (24)$$

The main difference between ILC and INMPC is that the last one uses a batch-varying LTV model. Therefore, it is more adequate for controlling this strongly non-linear pH plant. A simulation with an ILC controller ($K = 6 \cdot 10^{-4}$) is performed in Fig. 4. Although the gain is conservative, closed loop system becomes unstable after 5 iterations. It is clear that control gain at $t \leq 150$ and at $t \geq 150$ should be different (*time-varying*) and, moreover, it should be modified when the batch index is increased (*batch-varying*).
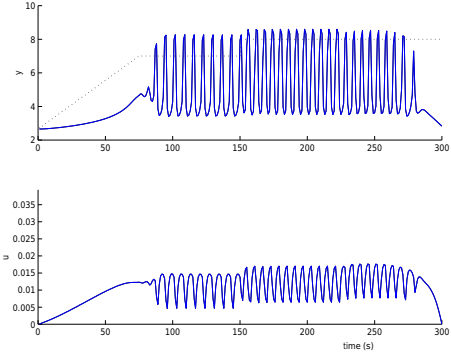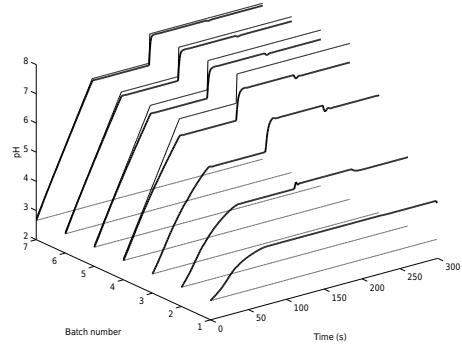

Fig. 5. Linear model-based BMPC

Fig. 6. Several batch simulation, convergence to the set point

BMPC is an iterative-model based predictive controller. Originally, it has been devised for dealing with linear batch processes, specially chemical processes (Lee *et al.*, 1999), and it belongs to the class of model-based iterative controllers. Other example of model-based iterative controller is the QILC, presented by Amann *et al.* (1996). In this work, we have tested the BMPC controller on the simulated pH plant.

Simulation studies have shown that this pH plant cannot be directly controlled with a linear model based controller, as can be seen in figure 5. The model was identified performing two open loop simulations with constant input ($u = 0.010$ and $u = 0.012$). Unstable trajectories were obtained, due to the strong nonlinearity of the process. This behavior can be very harmful in controlled systems. Nevertheless, it should be pointed out that a linearization technique could allow the use of BMPC in this plant.

In spite of the process difficulties, INMPC has shown good capabilities to control this plant. Fig. 6 shows the simulation of several sequential batches controlled with INMPC. The set point is only shown in the last 4 simulations to keep the figure clear. Noise variance is equal to 0.01. The controller is able to reach the set point in a few batches.

However, the behavior deteriorates in a noisy environment or in the presence of greater distur-
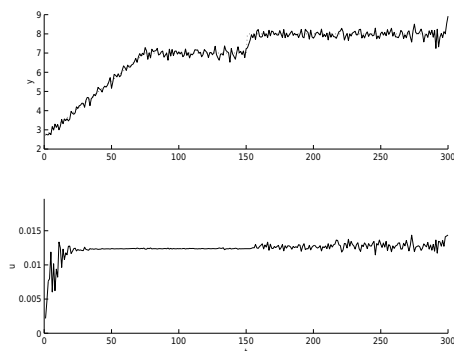
Fig. 7. Controlled system with uncertainties

bances. Indeed, if the step in acid flow is increased to 100% or the noise levels are increased to 0.5, system could even become oscillatory. This unstable behavior is caused by large prediction errors due to the effect of uncertainties together with the strong nonlinearity. Filters described in section 3.2 are useful in this situation. A median filter with parameters $l = 4$, $r = 2$ and $s = 2$ and an EWMA filter with $\lambda = 0.2$ are tested. These filters permits the controller to deal with higher level of disturbances, as shown in Fig. 7.

5. CONCLUSIONS

This paper has presented an strategy that has been designed to control a batch pH reactor, combining nonlinear predictive control with the learning capabilities of batch-wise control.

The control law enhances the nonlinear controller with the ability to make continuous batch-to-batch trajectory refinement. Simulation results are presented to illustrate performance improvements that can be achieved by the new method over the conventional MPC and *learning* methods.

The inclusion of real-time data collected during the on-going batch run in addition to those from the past runs make the control strategy capable not only of eliminating repeated errors but also of responding to new disturbances that occur during the run. This strategy can improve existing strategies by the use of a nonlinear controller devised along the last-run trajectory as well as by the inclusion of filters.

A comparison between INMPC and conventional iterative controllers has been made. The time and batch varying nature of the control gain permits INMPC to improve the transient response, decreasing the number of iterations needed to algorithm convergence.

REFERENCES

Amann, N., D. H. Owens and E. Rogers (1996). Iterative learning control for discrete-time sys-

tems with exponential rate of convergence. *IEE Proc. on Control Theory and Applications* **143**(2), 217–224.

Babuska, R., J. Oosterhoff, A. Oudshoorn and P.M. Bruijn (2002). Fuzzy self-tuning PI control of pH in fermentation. *Engineering Applications of Artificial Intelligence* **15**, 3–15.

Bordons, C. and F. Dorado (2002). Non-linear models for an gypsum kiln. a comparative analysis. In: *Preprints 15 th IFAC World Congress*. Barcelona.

Camacho, E. F. and C. Bordons (2004). *Model Predictive Control* 2nd *edition*. Springer-Verlag. London.

De Keyser, R. (1997). A gentle introduction to model based predictive control. In: *Proc. International Conference on Control Engineering and Signal Processing*. Lima, Peru.

Garrido, R., M. Adroer and M. Poch (1997). Wastewater neutralization control based in fuzzy logic: simulation results. *Ind. Eng. Chem. Res.* **36**, 1665.

Hara, S., Y. Yamamoto, T. Omata and N. Nakano (1988). Repetitive control system: a new type servo system for periodic exogeneous signals. *IEEE Trans. Automat. Contr.* **33**, 659–668.

Henson, M.A. and D.E. Seborg (1994). Adaptive nonlinear control of a pH neutralization process. *IEEE Trans. Contr. Syst. Technol.* **2**(3), 169–182.

Lee, K.S., I-S. Chin, H.J. Lee and J.H. Lee (1999). Model predictive control technique combined with iterative learning for batch processes. *AIChE J.* **45**(10), 2175–2186.

Longman, R.W. (1997). Design methodologies in learning and repetitive control. In: *Proc. of the 2nd ASCC*. Seoul, Korea.

McMillan, G.K. (1984). *pH Control*. Instrument Society of America.

Mezghani, M., G. Roux, M. Cabassud, M.V. Le Lann, B. Dahhou and G. Casamatta (2002). Application of iterative learning control to an exothermic semibatch chemical reactor. *IEEE Trans. Contr. Syst. Technol.* **10**(6), 822–834.

Moore, K.L. (1998). Iterative learning control: an expository overview. *Applied and Computational Controls, Signal Processing and Circuits* pp. 425–488.

Moyne, J., E. del Castillo and A. Hurwitz (2001). *Run-to-run control in semiconductor manufacturing*. CRC Press LLC. Boca Raton, Florida.

Sing, C.H. and B. Postlethwaite (1997). pH control: handling nonlinearity and deadtime with fuzzy relational model-based control. *IEE Proc Control Theory Appl* **144**(3), 263–268.

Wright, R.A. and C. Kravaris (2001). On-line identification and nonlinear control of an industrial pH process. *Journal of Process Control* **11**(36), 361–374.