Infinite Horizon MPC applied to batch processes. Part II

Alejandro H. González[1], Eduardo J. Adam[1], Darci Odloak[2] and Jacinto L. Marchetti[1]

[1] *Institute of Technological Development for the Chemical Industry (INTEC), CONICET - Universidad Nacional del Litoral. Santa Fe – Argentina. (alejgon,eadam,jlmarch)@santafe-conicet.gov.ar*
[2] *Department of Chemical Engineering, Polytechnic School, University of São Paulo São Paulo – SP, Brazil (e-mail: odloak@usp.br)*

Abstract: In the second stage of this work (part II), a new infinite horizon model predictive controller (IHMPC), with learning properties applied to batch processes is presented. When a batch process is attempted to be controlled two convergence analyses are necessary: the convergence into a given iteration or batch run (intra-run stability) and the convergence from run to run (inter-run stability, considering an infinite number of batch runs). As was shown in González et al., 2009, to account for the first one, the proposed strategy uses a virtual horizon that matches the traditional idea of infinite receding horizon of MPC with the finite duration of the run batch. To account for the second convergence analysis, a learning scheme based on the closed-loop paradigm of the IHMPC, is developed. To evaluate the proposed controller, a numerical example corresponding to batch reactor is shown, where the learning properties of the algorithm can be clearly seen.

1. INTRODUCTION

A batch process is one that continuously repeats a finite-duration procedure (run) along the time. This kind of systems can be found in several industrial fields (Lee and Lee, 2000; Bonvin 2006; Cueli and Bordons, 2008). Because of its characteristic, these repetitive processes have two counter indexes (some authors call them *two time scales*): one, of finite length, being the time within a run or trail, and the other, of infinite length, identifying the number of runs. As a consequence of this two different time scales, handling repetitive systems requires a control strategy that accounts for two different objectives: the first one is an on-line or within-batch control, which rejects disturbances occurring during a given run, and no necessarily remain unmodified for the next run. The other, is the run to run control, which reject disturbances that remain almost constant from one run to the next, and so, the controller can use information from previous operations. In this last case, a control scheme with learning properties is desired.

As it was said in the first stage (González et al., 2009), the IHMPC proposed in this work is formulated under a closed-loop paradigm (Rossiter, 2003). The basic idea of a closed-loop paradigm is to choose a stabilizing control law and assume that this law (underlying input sequence) is present throughout the predictions. The idea here is to consider an underlying control sequence as a manipulated input candidate (input reference) for the perfect tracking control, and to associate this input reference with the learning vector (i.e. the vector that is updated from one batch to the next, to improve the performance). If there is no additional information (first iteration), the input reference could be a constant value. Then, by means of a learning procedure (based on the time convergence for each batch), it is ensured that it converges iteratively to the perfect tracking control (run to run convergence). This is the way the proposed controller accounts for the typical characteristics of batch processes, i.e., finite time duration and events repetition.

The paper is organized as follows. In section 2 the basic definition and notation are presented. Then, in section 3, it is introduced the proposed MPC formulation and some related properties. The repetitive learning scheme (main result) is presented in section 4. Finally, a succinct illustrative example and the conclusion are presented in sections 5 and 6, respectively.

2. PRELIMINARIES

We assume here the same preliminaries definition considered in the part I of the paper, except for the batch index *i*, which will explicitly appear in the formulation in order to identify each batch run. So, the quantities $\boldsymbol{u}$, $\boldsymbol{y}$, $\boldsymbol{u}^r$ and $\boldsymbol{d}$ will be replaced by $\boldsymbol{u}^i$, $\boldsymbol{y}^i$, $\boldsymbol{u}^{i^r}$ and $\boldsymbol{d}^i$. The output disturbance, $\boldsymbol{d}^i$, is assumed to be known. (it is assumed to remain constant for several batch runs).

Here, the nominal model is the same as the one presented in the part I of this work (González et al., 2009).

2.1. Indexes

To clarify the notation, we define the following indexes:

"*i*" is the iteration or run index, where i=0 is the first batch run, when any learning procedure is applied.

"*k*" is the time into a given batch run. For a given iteration, it goes from 0 to T_f-1 (that is, T_f time instants).

"*j*" is the time for the MPC predictions. For a given batch run, and a given time instant into the batch run, it goes from 0 to T_f-1.

2.2. Convergence analysis

In the next sections, we will consider two convergence analyses:

Intra-run convergence: concerns the decreasing of a Lyapunov function (associated to the output error) along the run time k, that is, $V\left(y_{k+1}^i - y_{k+1}^r\right) \le V\left(y_k^i - y_k^r\right)$, for $k = 1,\cdots,T_f - 1$, in one specific batch. If the control algorithm execution goes beyond T_f, with $k \to \infty$, and the output reference remains constant at the final reference value ($y_k^r = y_{T_f}^r$ for $T_f \le k < \infty$), then the intra-run convergence concerns the convergence of the output to the final value of the output reference trajectory ($y_k^i \to y_{T_f}^r$ as $k \to \infty$). This convergence was proved in González et al. (2009).

Inter-run convergence: concerns the convergence of the output trajectory to the complete reference trajectory from one batch to the next one, that is, considering the output of a given run as a vector of T_f components ($\boldsymbol{y}^i \to \boldsymbol{y}^r$ as $i \to \infty$).

3. BASIC FORMULATION

For the first proposed MPC formulation we will assume that an appropriate input reference is available, and the disturbance sequence, d_k^i, is known. The MPC optimization problem associated to batch run i is as follows:

Problem P1)

$$\min_{\left\{\overline{u}_{k|k}^i,\cdots,\overline{u}_{k+T_f-1|k}^i\right\}} V_k^i = \sum_{j=0}^{T_f-1} \ell\left(e_{k+j|k}^i, \overline{u}_{k+j|k}^i\right) + F\left(x_{k+T_f|k}^i\right)$$

subject to:

$$e_{k+j|k}^i = C x_{k+j|k}^i + d_{k+j}^i - y_{k+j}^r, \quad j = 0,\cdots,T_f, \tag{1}$$

$$x_{k+j+1|k}^i = A x_{k+j|k}^i + B u_{k+j|k}^i, \quad j = 0,\cdots,T_f, \tag{2}$$

$$u_{k+j|k}^i \in U, \quad j = 0,1,\cdots,N-1, \tag{3}$$

$$u_{k+j|k}^i = u_{k+j}^{i^r} + \overline{u}_{k+j|k}^i, \quad j = 0,1,\cdots,T_f - 1, \tag{4}$$

$$\overline{u}_{k+j|k}^i = 0, \quad j = N, N+1,\cdots,T_f - 1, \tag{5}$$

where

$$u_k^{i^r} = u_{T_f-1}^{i^r} = G^{-1}\left(y_{T_f}^r - d_{T_f}^i\right), \quad k = T_f - 1, T_f,\cdots,2T_f - 2,$$

$$y_k^r = y_{T_f}^r, \quad k = T_f, T_f + 1,\cdots,2T_f - 1.$$

The importance of the $\overline{u}_{k+j|k}^i$ in the MPC algorithm is described in González et al. (2009), in the remark 0.

Remark 1: This problem is the one presented by González et al. (2009) in the section 3 (Problem P1), except that now it is associated to a particular batch run i. As a result, all the properties are the same for both formulations, and they are omitted here for brevity. Particularly, the convergence of the MPC cost (virtual horizon convergence) can be expressed as:

$$V_k^{i^{opt}} - V_{k-1}^{i^{opt}} + \ell\left(e_{k-1|k-1}^{i^{opt}}, \overline{u}_{k-1|k-1}^{i^{opt}}\right) \le 0 \tag{6}$$

4. IHMPC WITH LEARNING PROPERTIES

In the last section we studied the within-run control problem. We assumed that an input reference is available and the output disturbance is known. One way is by associating the current input reference and disturbance to the last batch ones (i.e. the implemented input and the estimated disturbance during the last run, beginning with a constant sequence and a zero value, respectively). In this way, a dual MPC with learning properties accounting for the run-to-run control is obtained. Next, we will try to elucidate this point.

Consider the problem P1 (González et al., 2009) for a given batch run i, with the following variation:

$$\begin{aligned}
u_k^{i^r} &= u_{k|k}^{i-1^{opt}}, \quad k = 0,\cdots,T_f - 1\\
d_k^i &= d_k^{i-1}, \quad k = 1,\cdots,T_f\\
u_k^{i^r} &= u_{T_f-1}^{i^r} = G^{-1}\left(y_{T_f}^r - d_{T_f}^{i-1}\right), \quad k = T_f, T_f + 1,\cdots,2T_f - 1\\
d_k^i &= d_{T_f}^{i-1}, \quad k = T_f, T_f + 1,\cdots,2T_f - 1
\end{aligned} \tag{7}$$

where the disturbance, as well as the states for prediction, are observer-based estimates.

The idea here is to associate the input reference and the disturbance corresponding to run i with the actual input and disturbance implemented at the run i-1 (See Figure 1). That is, $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{i-1}$, and $\boldsymbol{d}^i = \boldsymbol{d}^{i-1}$ for i=1,2,…, and $\boldsymbol{u}^0 = \begin{bmatrix} G^{-1} y_{T_f}^r & \ldots & G^{-1} y_{T_f}^r \end{bmatrix}$, $\boldsymbol{d}^0 := \begin{bmatrix} 0 & \ldots & 0 \end{bmatrix}$. In addition, it is possible to define a vector of differences between two consecutive implemented input sequences as $\delta^i := \boldsymbol{u}^i - \boldsymbol{u}^{i-1}$, and it is interesting to notice that this vector is given by $\delta^i = \begin{bmatrix} \overline{u}_{0|0}^{i^{opt}T} & \cdots & \overline{u}_{T_f-1|T_f-1}^{i^{opt}T} \end{bmatrix}$.

This means that this difference vector is made of the first element of the solution of each optimization problem, for k=0,…,T_f-1, used in a receding horizon manner.

4.1 New inter-run convergence constraints for batch process

Now, in order to achieve a run-to-run convergence, we replace the original constraint (5) of problem P1 by the following one:

$$\overline{u}_{k+j|k}^i = 0, \quad j = N_s, N_s + 1,\cdots,T_f - 1, \tag{8}$$

where

$$N_s = \min(H, N). \tag{9}$$

In this way, a new *shrinking control horizon* N_s is defined, i.e., for the last N time steps (k=T_f-N,…,T_f-1) of each run, the control horizon is reduced as the time steps k increases. As will be shown later, this modification allows the successive run costs to be matched.

Remark 2: The new *shrinking control horizon* allows the cost to be expressed by means of

$$V_k^i = \sum_{j=0}^{H-1} \ell\left(e_{k+j|k}^i, \overline{u}_{k+j|k}^i\right) + F\left(x_{k+H|k}^i\right), \tag{10}$$

regardless of the value of k.

The next property shows to be useful for the convergence proof:

Property 1: Assuming that a shrinking control horizon is used, then, Eq. (6) holds true for the last N MPC costs of a given run. Furthermore, the last cost of a given run "i" are given by:

$$V_{T_f-1}^{i^{opt}} = \ell\left(e_{T_f-1|T_f-1}^{i^{opt}}, \overline{u}_{T_f-1|T_f-1}^{i^{opt}}\right) + F\left(x_{T_f|T_f-1}^{i^{opt}}\right),$$

and since current and one steps prediction are coincident with the actual values (Remark 4), it follows that:

$$V_{T_f-1}^{i^{opt}} = \ell\left(e_{T_f-1}^i, \overline{u}_{T_f-1}^i\right) + F\left(x_{T_f}^i\right). \qquad (11)$$

Proof. Similar to the proof of theorem 1, in the first stage of the work (González et al., 2009), it is possible to define a feasible solution to the optimization problem at time *k*, based on the solution at time *k*-1. Then, showing that the cost corresponding to these solutions is not greater than the optimal cost at time *k*-1, inequality (6) holds.

4.2 Properties of the proposed algorithm

One interesting point here is to answer what happens if the MPC controller receives as input reference trajectory a control sequence that, if it is injected to the system, produces a null output error. Since the MPC controller does not add the input reference to the computed input (as typical correction) but to predicted inputs, some care must be taken. Property 2, above, assures that for this input reference the MPC cost is null. Without lose of generality, we will consider the nominal case (no difference between plant and model) for simplicity in what follows.

Definition 1: Let us consider the following perfect control input trajectory

$$\boldsymbol{u}^{perf} = \left[u_0^{perf^T} \quad \cdots \quad u_{T_f-1}^{perf^T}\right]^T,$$

which represents the control sequence that, if is injected into the system, produces a null output error trajectory $\boldsymbol{e}^i = \left[e_1^{i^T} \quad \cdots \quad e_{T_f}^{i^T}\right]^T = \left[0 \quad \cdots \quad 0\right]^T$. It is assumed at this point that the output reference is designed in such a way (smooth shape), and the disturbances are such that, the perfect control is possible.

Notice that the perfect control would be physically possible, if an infinite number of iterations are performed.

Property 2: If the MPC cost penalization matrices, *Q* and *R*, are definite positive (*Q*>0 and *R*>0) and perfect control input trajectory is a feasible trajectory, then $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf} \Leftrightarrow V_k^{i^{opt}} = 0$, for *k*=0,…,$T_f$-1; where $V_k^{i^{opt}} = \sum_{j=0}^{H-1} \ell\left(e_{k+j|k}^{i^{opt}}, \overline{u}_{k+j|k}^{i^{opt}}\right) + F\left(x_{k+H|k}^{i^{opt}}\right)$.

Proof.

$\Leftarrow$) Let us assume that $V_k^{i^{opt}} = 0$, for *k*=0,…,T_f-1. Then, the optimal predicted output error and input will be given by $e_{k+j|k}^{i^{opt}} = 0$, for *j*=0,…,T_f, and $\overline{u}_{k+j|k}^{i^{opt}} = 0$, for *j*=0,…,$T_f$-1, respectively. If $e_{k+j|k}^{i^{opt}} = 0$ and $\overline{u}_{k+j|k}^{i^{opt}} = 0$ simultaneously, it follows that $u_k^{i^r} = u_k^{perf}$ for *k*=0,…,T_f-1, since it is the only input sequence that produces null predicted output error (otherwise, the optimization will necessarily find an equilibrium such that $\left\|e_{k+1|k}^{i^{opt}}\right\| > 0$ and $\left\|\overline{u}_{k|k}^{i^{opt}}\right\| > 0$, provided that *Q*>0 and *R*>0 by hypothesis). Consequently, $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$.

$\Rightarrow$) Let us assume that $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$. Because of the definition of the perfect control input, the optimization problem without any input correction will produce a sequence of null output error predictions given by

$$\begin{aligned}
e_{k|k}^i &= 0 \\
e_{k+1|k}^i &= Cx_{k+1|k}^i - y_{k+1}^r = C\left[Ax_{k|k}^i + Bu_k^{perf}\right] - y_{k+1}^r = 0 \\
&\vdots \\
e_{k+T_f|k}^i &= Cx_{k+T_f|k}^i - y_{k+T_f}^r \\
&= C\left[A^{T_f}x_{k|k}^i + ABu_k^{perf} + \cdots + Bu_{k+T_f-1}^{perf}\right] - y_{k+T_f}^r = 0.
\end{aligned}$$

Consequently, the optimal sequence of decision variables (predicted inputs) will be $\overline{u}_{k+j|k}^{i^{opt}} = 0$ for *k*=0,…,T_f-1 and *j*=0,…,T_f-1, since no correction is needed to achieve null predicted output error. This means that $V_k^{i^{opt}} = 0$, for *k*=0,…,T_f-1.□

4.3 Inter-run convergence

Let us consider the following optimization problem:

Problem P2)

$$\min_{\left\{\overline{u}_{k|k}^i, \cdots, \overline{u}_{k+T_f-1|k}^i\right\}} V_k^i$$

subject to:

(1)-(4), (7)-(9).

When we say that the algorithm converges from run to run it means that both, the output error trajectory $\boldsymbol{e}^i$ and the input difference between two consecutive implemented inputs, $\boldsymbol{\delta}^i = \boldsymbol{u}^i - \boldsymbol{u}^{i-1}$, converges to zero as $i \to \infty$. Following an Iterative Learning Control nomenclature, this means that the implemented input, $\boldsymbol{u}^i$, converges to the perfect control input $\boldsymbol{u}^{perf}$ for a sufficiently large number of iterations.

To show this convergence, we will define a cost associated to each run, which penalizes the output error. As it was said, T_f MPC optimization problems are solved at each run *i*, that is, from *k*=0 to *k*=T_f -1. So, a candidate to describe the run cost is as follows:

$$J_i := \sum_{k=0}^{T_f-1} V_k^{i^{opt}}, \qquad (12)$$

where $V_k^{i^{opt}}$ represents the optimal cost of the on-line MPC optimization problem at time *k*, corresponding to the run *i*. Notice that this MPC cost, once the optimization problem P2 is solved and an optimal input sequence is obtained, is a function of only $e_{k|k}^{i^{opt}} = \left(y_{k|k}^{i^{opt}} - y_k^r\right) = e_k^i$.

Therefore, it makes sense using (12) to define a batch cost, since it represents a (finite) sum of positive penalizations of the current output error, that is to say, a positive function of $\boldsymbol{e}^i$. However, since the new batch index is made of outputs predictions rather than of actual errors, some cares must be taken into consideration. Firstly, as occurs with usual indexes, we should demonstrate that null output error vectors produce null costs (which is not trivial because of predictions). Secondly, we should demonstrate that the perfect control input corresponds to a null cost.

Property 3: If the MPC cost penalization matrices, *Q* and *R*, are definite positive (*Q*>0 and *R*>0) and perfect control input trajectory is a feasible trajectory, cost (12), which is an implicit function of $\boldsymbol{e}^i$, is such that, $\boldsymbol{e}^i = 0 \Leftrightarrow J_i = 0$.

Proof.

$\Rightarrow$) Let us assume that $\boldsymbol{e}^i$=0. This means that $e_{k|k}^{i^{opt}} = 0$, for k=0,…,T_f. Now, assume that the input reference vector is different from the perfect control input, $\boldsymbol{u}^{i^r} \neq \boldsymbol{u}^{perf}$, and consider the output error predictions necessary to compute the MPC cost V_k^i:

$$e_{k|k}^i = 0$$
$$e_{k+1|k}^i = Cx_{k+1|k}^i - y_{k+1}^r = C\left[Ax_{k|k}^i + Bu_k^{i^r} + B\overline{u}_{k|k}^i\right] - y_{k+1}^r$$
$$\vdots$$

Since $u_k^{i^r}$ is not an element of the perfect control input, then $C\left[Ax_{k|k}^i + Bu_k^{i^r}\right] - y_{k+1}^r \neq 0$. Consequently, (assuming that CB is invertible) the input $\overline{u}_{k|k}^{i*}$ necessary to make $e_{k+1|k}^{i^{opt}} = 0$, will be given by:

$$\overline{u}_{k|k}^{i*} = (CB)^{-1}\left(y_{k+1}^r - C\left[Ax_{k|k}^i + Bu_k^{i^r}\right]\right),$$

which is a non null value. However, the optimization will necessary find an equilibrium solution such that $\left\|e_{k+1|k}^{i^{opt}}\right\| > 0$ and $\left\|\overline{u}_{k|k}^{i^{opt}}\right\| < \left\|\overline{u}_{k|k}^{i*}\right\|$, since Q>0 and R>0 by hypothesis. This implies that[1] $e_{k+1|k}^{i^{opt}} = e_{k+1|k+1}^{i^{opt}} \neq 0$, contradicting the initial assumption of null output error.

From this reasoning for subsequent output errors, it follows that the only possible input reference to achieve $\boldsymbol{e}^i$=0 will be the perfect control input ($\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$). If this is the case, it follows that $V_k^{i^{opt}} = 0$, for k=0,…,T_f (Property 2), and so, J_i=0.

$\Leftarrow$) let us assume that J_i=0. Then, $V_k^{i^{opt}} = 0$, which implies that $e_{k+j|k}^{i^{opt}} = 0$, for k=0,…,T_f and for j=0,…,T_f. Particularly, $e_{k|k}^{i^{opt}} = 0$, for k=0,…,T_f, which implies $\boldsymbol{e}^i$=0.□

Corollary 1

If the MPC cost penalization matrices, Q and R, are definite positive, then $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf} \Leftrightarrow J_i = 0$. Otherwise, $\boldsymbol{u}^{i^r} \neq \boldsymbol{u}^{perf} \Rightarrow J_i \neq 0$.

Proof

It follows from Property 2 and Property 3.□

Now, we establish the run to run convergence with the following theorem.

Theorem 1

For the system (5)-(7) of González et al., 2009, by using the control law derived from the on-line execution of problem P2 in a receding horizon manner, together with the learning updating (7), and assuming that a feasible perfect control input trajectory there exists, the output error trajectory $\boldsymbol{e}^i$ converges to zero as $i \to \infty$. In addition, $\boldsymbol{\delta}^i$ converges to zero as $i \to \infty$, which means that the reference trajectory $\boldsymbol{u}^{i^r}$ converges to $\boldsymbol{u}^{perf}$.

Proof

See Appendix A.

Remark 3: In most real systems a perfect control input trajectory is not possible to reach (which represents a system limitation rather than a controller limitation). In this case, the costs $V_k^{i^{opt}}$ will converge to a non-null finite value as $i \to \infty$,and then, since the operation cost J_i is decreasing (see Appendix A), it will converge to the smallest possible value.

Remark 4: In the same way that the intra-run convergence can be extended to determine a *variability index* in order to establish a quantitative concept of stability (β-stability,), for finite-run systems (Remark 9 of González et al., 2009); the inter-run convergence can be extended to establish stability conditions similar to the ones presented in Srinivasan and Bonvin, 2007.

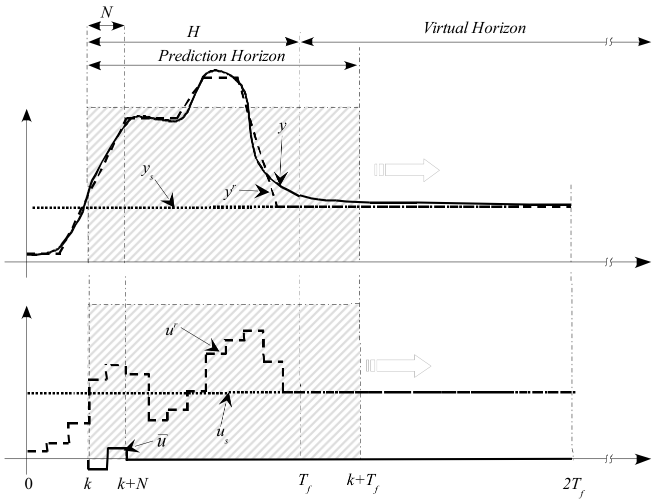

Fig. 1. MPC diagram corresponding to each batch.

5. ILUSTRATIVE EXAMPLE

In order to evaluate the proposed controller performance we assume a true and nominal process given by (Lee and Lee 1997, 2000) G(s)=1/15s^2+8s+1 and G(s)=0.8/12s^2+7s+1, respectively. The sampling time adopted to develop the discrete state space model is T=1 and the final batch time is given by T_f=90T. The proposed strategy achieves a good control performance in the first two or three iterations, with a rather reduced control horizon. The controller parameters are as follows: Q=1500, R=0.05, N=5. Fig. 2 shows the output response together with the output reference, and the inputs $\boldsymbol{u}^i$ and $\overline{\boldsymbol{u}}^i$, for the first and third iteration. At the first iteration, since the input reference is a constant value ($u_{T_f-1}^{i^r} = 0$), $\boldsymbol{u}^i$ and $\overline{\boldsymbol{u}}^i$ are the same, and the output performance is quite poor (mainly because of the model mismatch). At the third iteration, however, given that a disturbance state is estimated from the previous run, the output response and the output reference are undistinguishable. As expected, the batch error is reduced drastically from run 1 to run 3, while the MPC cost is decreasing (as was established in Theorem 1) for each run (Fig. 3). Notice that the MPC cost is normalized taking

[1] Note that for the nominal case is $e_{k+1|k+1}^i = e_{k+1|k}^i$.

into account the maximal value ($V_k^i / V_{\max}^i$), where $V_{\max}^1 \approx 1\cdot 10^6$ and $V_{\max}^1 \approx 286.5$. This shows that the MPC cost J_i decrease from one run to the next, as was stated in Theorem 2. Finally, Fig. 4 shows the normalized norm of the error corresponding to each run.

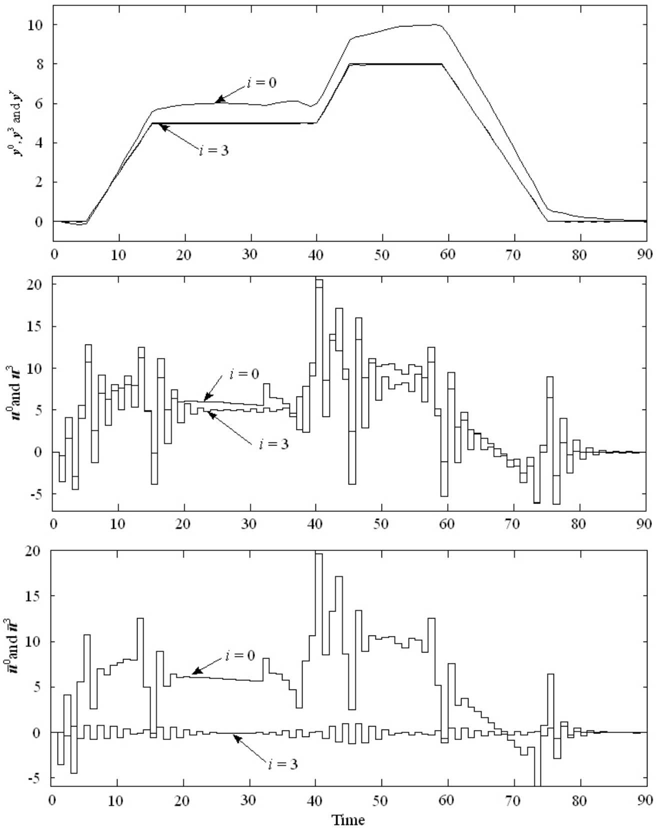

Fig. 2. Output and input responses.

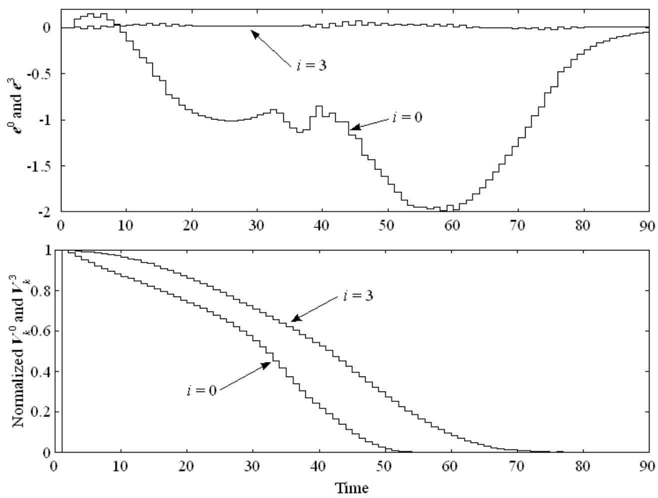

Fig. 3. Error and MPC cost.

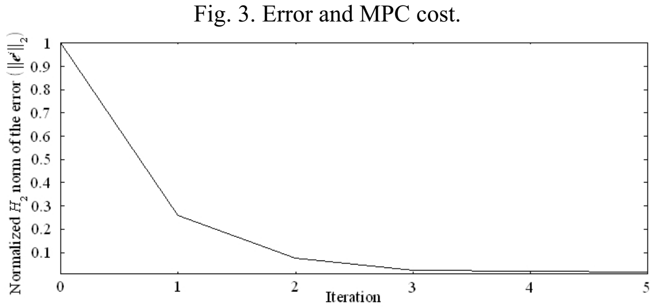

Fig. 4. Norm of the iteration error.

6. CONCLUSIONS

In this paper a different formulation of a stable IHMPC with learning properties applied to batch processes is presented. For the case in which the process parameters remain unmodified for several batch runs, the formulation allows a repetitive learning algorithm, which updates the control variable sequence to achieve nominal perfect control performance. Two extension of the present work can be considered. The easier one is the extension to linear-time-variant (LTV) models, which would allow representing the non-linear behavior of the batch processes better. A second extension is to consider the robust case (e.g. by incorporating multi model uncertainty into the MPC formulation). These two issues will be studied in future works.

APPENDIX A.

Proof of Theorem 1

The idea here is to show that $V_k^{i^{opt}} \leq V_k^{i-1^{opt}}$ for k=0,…,T_f-1, and so, $J_i \leq J_{i-1}$. First, let us consider the case in which the sequence of T_f optimization problems P2 "do nothing" at a given run i. That is, we will consider the case in which

$$\delta^i = \begin{bmatrix} \overline{u}_{0|0}^{i^{opt^T}} & \cdots & \overline{u}_{T_f-1|T_f-1}^{i^{opt^T}} \end{bmatrix}^T = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T,$$

for a given run i. So, for the nominal case, the total actual input will be given by

$$\boldsymbol{u}^i = \boldsymbol{u}^{i-1} = \begin{bmatrix} u_0^{i-1^T} & \cdots & u_{T_f-1}^{i-1^T} \end{bmatrix}^T = \begin{bmatrix} u_{0|0}^{i-1^{opt^T}} & \cdots & u_{T_f-1|T_f-1}^{i-1^{opt^T}} \end{bmatrix}^T.$$

and the run cost corresponding to this (fictitious) input sequence will be given by

$$\tilde{J}_i := \sum_{k=1}^{T_f} \tilde{V}_k^i ,$$

where

$$\begin{aligned} \tilde{V}_k^i &= \sum_{j=0}^{N_s-1} \ell\left(e_{k+j|k+j}^{i-1^{opt}}, \overbrace{\overline{u}_{k+j|k+j}^i}^{=0} \right) + \sum_{j=N_s}^{H-1} \ell\left(e_{k+j|k+j}^{i-1^{opt}}, 0\right) + F\left(x_{k+H|k+H-1}^{i-1^{opt}} \right) \\ &= \sum_{j=0}^{H-1} \ell\left(e_{k+j}^{i-1}, 0\right) + F\left(x_{k+H}^{i-1} \right). \end{aligned} \qquad \text{(A1)}$$

Since the input reference, $u_{k+j}^{i^r}$, that uses each optimization problems is given by $u_{k+j}^{i^r} = u_{k+j|k+j}^{i-1^{opt}}$, then the resulting output error will be given by $e_{k+j|k+j}^{i^{opt}} = e_{k+j}^{i-1}$ for j=0,…,H. In other words, the open loop output error predictions made by the MPC optimization at each time k, for a given run i, will be the actual (implemented) output error of the past run i-1. Here it must be noticed that e_{k+j}^{i-1} refers to the actual error of the system, that is, the error produced by the implemented input $u_{k+j-1}^{i-1} = u_{k+j-1|k+j-1}^{i-1^{opt}}$. Moreover, because of the proposed inter run convergence constraint, the implemented input will be $u_{T_f-1}^{i-1^r}$, for $j \geq H$.

Let now consider the optimal MPC costs corresponding to k=0,…,T_f-1, of a given run i-1. From the recursive use of (6) we have

$$V_1^{i-1^{opt}} + \ell\left(e_0^{i-1},\overline{u}_0^{i-1}\right) \le V_0^{i-1^{opt}}$$

$$\vdots$$

$$V_{T_f-1}^{i-1^{opt}} + \ell\left(e_{T_f-2}^{i-1},\overline{u}_{T_f-2}^{i-1}\right) \le V_{T_f-2}^{i-1^{opt}}$$

Then, adding the second term of the left hand side of each inequality to both sides of the next one, and rearranging the terms, we can write

$$V_{T_f-1}^{i-1^{opt}} + \ell\left(e_{T_f-2}^{i-1},\overline{u}_{T_f-2}^{i-1}\right) + \cdots + \ell\left(e_0^{i-1},\overline{u}_0^{i-1}\right) \le V_0^{i-1^{opt}}. \quad \text{(A2)}$$

From (11), the cost $V_{T_f-1}^{i-1^{opt}}$, which is the cost at the end of the run i-1, will be given by,

$$V_{T_f-1}^{i-1^{opt}} = \ell\left(e_{T_f-1}^{i-1},\overline{u}_{T_f-1}^{i-1}\right) + F\left(x_{T_f}^{i-1}\right). \quad \text{(A3)}$$

Therefore, by substituting (A3) in (A2), we have

$$\begin{aligned} F\left(x_{T_f}^{i-1}\right) + \ell\left(e_{T_f-1}^{i-1},\overline{u}_{T_f-1}^{i-1}\right) + \ell\left(e_{T_f-2}^{i-1},\overline{u}_{T_f-2}^{i-1}\right) + \cdots \\ + \ell\left(e_0^{i-1},\overline{u}_0^{i-1}\right) \le V_0^{i-1^{opt}} \end{aligned} \quad \text{(A4)}$$

Now, the pseudo cost (A1) at time k=0, $\tilde{V}_0^i$, can be written as

$$\begin{aligned} \tilde{V}_0^i &= \sum_{j=0}^{T_f-1} \ell\left(e_j^{i-1},0\right) + F\left(x_{T_f}^{i-1}\right) \\ &= \sum_{j=0}^{T_f-1} \ell\left(e_j^{i-1},\overline{u}_j^{i-1}\right) + F\left(x_{T_f}^{i-1}\right) - \sum_{j=0}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\| \end{aligned}, \quad \text{(A5)}$$

and from the comparison of the left hand side of inequality (A4) with (A5), it follows that

$$\tilde{V}_0^i \le V_0^{i-1^{opt}} - \sum_{j=0}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\|.$$

Repeating now this reasoning for k=1,…,T_f-1 we conclude that

$$\tilde{V}_k^i \le V_k^{i-1^{opt}} - \sum_{j=k}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\|, \qquad k = 0,\cdots,T_f-1.$$

Therefore, from the definition of the run cost $\tilde{J}_i$ we have[2]

$$\tilde{J}_i \le J_{i-1} - \sum_{k=0}^{T_f-1}\sum_{j=k}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\|. \quad \text{(A6)}$$

The MPC costs $\tilde{V}_k^i$ is such that $V_k^{i^{opt}} \le \tilde{V}_k^i$, since the solution $\overline{u}_{k+j|k}^i = 0$, for j=0,…,H is a feasible solution for problem P2 at each time k. This implies that

$$J_i \le \tilde{J}_i. \quad \text{(A7)}$$

From (A6) and (A7) we have

$$J_i \le \tilde{J}_i \le J_{i-1} - \sum_{k=0}^{T_f-1}\sum_{j=k}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\|, \quad \text{(A8)}$$

which means that the run costs are strictly decreasing if at least one of the optimization problems corresponding to the run i-1 find a solution $\overline{u}_{k+j|k}^{i-1} \ne 0$. As a result, two options arise:

I) Let us assume that $\boldsymbol{u}^{i^r} \ne \boldsymbol{u}^{perf}$. Then, by Corollary 1, $J_i \ne 0$ and following the reasoning used in the proof of Property 3, $\overline{u}_j^i \ne 0$, for some $1 \le j \le T_f$. Then, according to (A8), $J_{i+1} \le \tilde{J}_{i+1} \le J_i - \sum_{k=0}^{T_f-1}\sum_{j=k}^{T_f-1} \left\|\overline{u}_j^i\right\|$, with $\left\|\overline{u}_j^i\right\| > 0$ for some $1 \le j \le T_f$-1.

The sequence J_i will stop decreasing only if $\sum_{j=0}^{T_f-1}\left\|\overline{u}_j^i\right\| = 0$. In addition, if $\sum_{j=0}^{T_f-1}\left\|\overline{u}_j^i\right\| = 0$, then $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$, which implies that J_i =0. Therefore: $\lim_{i\to\infty} J_i = 0$, which, by Property 3 implies that $\lim_{i\to\infty} \boldsymbol{e}^i = 0$.

Notice that the last limit implies that $\lim_{i\to\infty} \delta^i = 0$, and consequently, $\lim_{i\to\infty} \boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$.

II) Let us assume that $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf}$. Then, by Corollary 1, J_i=0, and according to (A8), $J_{i+1} = \tilde{J}_{i+1} = J_i = 0$. Consequently, by Property 3, $\boldsymbol{e}^i$=0. □

REFERENCES

Cueli J. R. and C. Bordons (2008). Iterative nonlinear model predictive control. Stability, robustness and applications. *Control Engineering Practice*, **16**, 1023-1034.

González A. H., D. Odloak and O. A. Sotomayor (2008). Stable IHMPC for unstable systems. *IFAC* 2008.

González A. H., E. J. Adam, D. O. Odloak and J. L. Marchetti, "Infinite horizon MPC applied to batch processes. Part I", Reunión de Procesamiento de la Información y Control, (RPIC XIII), Rosario, Santa Fe – Argentina. (2009).

Lee, K. S. and J. H. Lee (1997). Model predictive control for nonlinear batch processes with asymptotically perfect tracking. *Computer chemical Engng*, **24**, 873-879.

Lee J. H., K. S. Lee and W. C. Kim (2000).Model-based iterative learning control with a quadratic criterion for time-varying linear systems. *Automatica*, *36*, 641-657.

Rossiter J.A. (2003). Model-Based Predictive Control. CRC Press.

Srinivasan B. and D. Bonvin (2007). Controllability and stability of repetitive batch processes. *Journal of Process Control*, **17**, 285-295.

[2] Notice that, if the run i implements the manipulated variable $u_j^i = u_j^{i-1} + \overline{u}_j^{i-1}$, j=0,1,...,T_f-1, and $\overline{u}_j^{i-1} \ne 0$ for some j; then, according to (A6) $\tilde{J}_i < J_{i-1}$. Unnaturally, to have found a non null optimal solution in the run i-1 is sufficient to have a strictly smaller cost for the run i.