9

Iterative Learning - MPC: An Alternative Strategy

Eduardo J. Adam[1] and Alejandro H. González[2]
[1]*Chemical Engineering Department - Universidad Nacional del Litoral*
[2]*INTEC (CONICET and Universidad Nacional del Litoral)*
Argentina

1. Introduction

A repetitive system is one that continuously repeats a finite-duration procedure (operation) along the time. This kind of systems can be found in several industrial fields such as robot manipulation (Tan, Huang, Lee & Tay, 2003), injection molding (Yao, Gao & Allgöwer, 2008), batch processes (Bonvin et al., 2006; Lee & Lee, 1999; 2003) and semiconductor processes (Moyne, Castillo, & Hurwitz, 2003). Because of the repetitive characteristic, these systems have two count indexes or time scales: one for the time running within the interval each operation lasts, and the other for the number of operations or repetitions in the continuous sequence. Consequently, it can be said that a control strategy for repetitive systems requires accounting for two different objectives: a short-term disturbance rejection during a finite-duration single operation in the continuous sequence (this frequently means the tracking of a predetermined optimal trajectory) and the long-term disturbance rejection from operation to operation (i.e., considering each operation as a single point of a continuous process[1]). Since in essence, the continuous process basically repeats the operations (assuming that long-term disturbances are negligible), the key point to develop a control strategy that accounts for the second objective is to use the information from previous operations to improve the tracking performance of the future sequence.

Despite the finite-time nature of every individual operation, the within-operation control is usually handled by strategies typically used on continuous process systems, such as PID ((Adam, 2007)) or more sophisticated alternatives as Model Predictive Control (MPC) (González et al., 2009a;b). The main difficulty arising in these applications is associated to the stability analysis, since the distinctive finite-time characteristic requires an approach different from the traditional one; this was clearly established in (Srinivasan & Bonvin, 2007). The operations sequence control can be handled by strategies similar to the standard Iterative Learning Control (ILC), which uses information from previous operations. However, the ILC exhibits the limitation of running open-loop with respect to the current operation, since no feedback corrections are made during the time interval the operation lasts.

In order to handle batch processes (Lee et al., 2000) proposed the Q-ILC, which considers a model-based controller in the iterative learning control framework. As usual in the ILC literature, only the iteration-to-iteration convergence is analyzed, as the complete input and

[1] In this context, continuous process means one that has not an end time.

output profiles of a given operation are considered as fix vectors (open-loop control with respect to the current operation). Another example is an MPC with learning properties presented in (Tan, Huang, Lee & Tay, 2003), where a predictive controller that iteratively improves the disturbance estimation is proposed. Form the point of view of the learning procedure, any detected state or output disturbance is taken like parameters that are updated iteration to iteration. Then, in (Lee & Lee, 1997; 1999) and (Lee et al., 2000), a real-time feedback control is incorporated into the Q-ILC (BMPC). As the authors declare, some cares must be taken when combining ILC with MPC. In fact, as read in Lee and Lee 2003, a simple-minded combination of ILC updating the nominal input trajectory for MPC before each operation does not work.

The MPC proposed in this Chapter 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. More precisely, the MPC propose here is an Infinite Horizon MPC (IHMPC) that includes an underlying control sequence as a (deficient) reference candidate to be improved for the tracking control. Then, by solving on line a constrained optimization problem, the input sequence is corrected, and so the learning updating is performed.

1.1 ILC overview

Iterative Learning Control (ILC) associates three main concepts. The concept *Iterative* refers to a process that executes the same operation over and over again. The concept *Learning* refers to the idea that by repeating the same thing, the system should be able to improve the performance. Finally, the concept *control* emphasizes that the result of the learning procedure is used to control the plant.

The ILC scheme was initially developed as a feedforward action applied directly to the open-loop system ((Arimoto et al., 1984) ; (Kurek & Zaremba, 1993); among others). However, if the system is integrator or unstable to open loop, or well, it has wrong initial condition, the ILC scheme to open loop can be inappropriate. Thus, the feedback-based ILC has been suggested in the literature as a more adequate structure ((Roover, 1996); (Moon et al., 1998); (Doh et al., 1999); (Tayebi & Zaremba, 2003)). The basic idea is shown in Fig. 1.

This scheme, in its discrete version time, operates as follows. Consider a plant which is operated iteratively with the same set-point trajectory, $y^r(k)$, with k going from 0 to a final finite value T_f, over and over again, as a robot or an industrial batch process. During the i-th trail an input sequence $u^i(k)$, with k going from 0 to a final finite value T_f, is applied to the plant, producing the output sequence $y^i(k)$. Both sequences, that we will call $\mathbf{u}^i$ and $\mathbf{y}^i$, respectively, are stored in the memory devise. Thus, two vectors with length T_f are available for the next iteration. If the system of Fig. 1 operates in open loop, using $\mathbf{u}^i$ in the $(i+1)$-th trail, it is possible to obtain the same output again and again. But, if at the $i+1$ iteration information about both, $\mathbf{u}^i$ and $\mathbf{e}^i = \mathbf{y}^i - \mathbf{y}^r$, where $\mathbf{y}^r = \left[y^r(0), \cdots, y^r(T_f)\right]$, is considered, then new sequences $\mathbf{u}^{i+1}$ and $\mathbf{y}^{i+1}$, can be obtained. The key point of the input sequence modification is to reduce the tracking error as the iterations are progressively increased. The

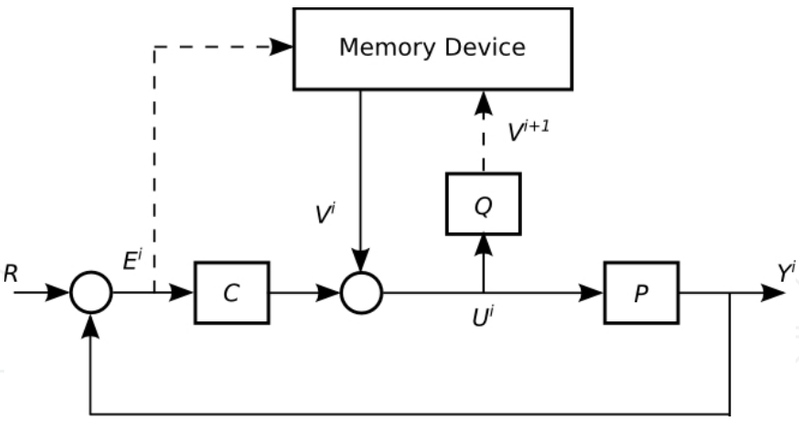


Fig. 1. Feedback-based ILC diagram. Here, continuous lines denote the sequence used during the i-th trail and dashed lines denote sequence that will be used in the next iteration.

purpose of an ILC algorithm is then to find a unique input sequence $\mathbf{u}^\infty$ which minimizes the tracking error.

The ILC formulation uses an iterative updating formula for the control sequence, given by

$$\mathbf{u}^{i+1} = f(\mathbf{y}^r, \mathbf{y}^i, ..., \mathbf{y}^{i-l}, \mathbf{u}^i, \mathbf{u}^{i-1}, ..., \mathbf{u}^{i-l}), \qquad i, l \geq 0. \tag{1}$$

This formula can be categorized according to how the information from previous iteration is used. Thus, (Norrlöf, 2000) among other authors define,

DEFINITION 0.1. *An ILC updating formula that only uses measurements from previous iteration is called* first order ILC. *On the other hand, when the ILC updating formula uses measurements from more than previous iteration, it is called a* high order ILC.

The most common algorithm suggested by several authors ((Arimoto et al., 1984); (Horowitz, 1993); (Bien & Xu, 1998); (Tayebi & Zaremba, 2003); among others), is that whose structure is given by

$$V^{i+1} = Q(z)(V^i + C(z)E^i), \tag{2}$$

where $V^1 = 0$, $C(z)$ denotes the controller transfer function and $Q(z)$ is a linear filter.

Six postulates were originally formulated by different authors ((Chen & Wen, 1999); (Norrlöf, 2000); (Scholten, 2000), among others).

1. Every iteration ends in a fixed discrete time of duration T_f.
2. The plant dynamics are invariant throughout the iterations.
3. The reference or set-point, $\mathbf{y}^r$, is given a priori.
4. For each trail or run the initial states are the same. That means that $x^i(0) = x^0(0)$, $i \geq 0$.
5. The plant output $y(k)$ is measurable.
6. There exists a unique input sequence, $\mathbf{u}^\infty$, that yields the plant output sequence, $\mathbf{y}$, with a minimum tracking error with respect to the set-point, $\mathbf{e}^\infty$.

Regarding the last postulate, we present now the key concept of *perfect control*.

DEFINITION 0.2. *The perfect control input trajectory,*

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

is one that, if injected into the system, produces a null output error trajectory

$$\boldsymbol{e}^i = \left[e_1^{i^T} \dots e_{T_f}^{i^T} \right]^T = [0 \dots 0]^T.$$

It is interesting to note that the impossibility of achieving discrete perfect control, at least for discrete nominal non-delayed linear models, is exclusively related to the input and/or states limits, which are always present in real systems and should be consistent with the control problem constraints. In this regard, a system with slow dynamic might require high input values and input increments to track an abrupt output reference change, producing in this way the constraint activation. If we assume a non-delayed linear model without model mismatch, the perfect control sequence can be found as the solution of the following (unconstrained) open-loop optimization problem

$$\mathbf{u}^{perf} = \arg\min_{\mathbf{u}^i} \sum_{k=1}^{T_f} \left\| e_k^i \right\|^2.$$

On the other hand, for the constrained case, the best possible input sequence, i.e., $\mathbf{u}^{\infty}$, is obtained from:

$$\mathbf{u}^{\infty} = \arg\{\min_{\mathbf{u}^i} \sum_{k=1}^{T_f} \left\| e_k^i \right\|^2, \ s.t. \ \mathbf{u} \in \mathbf{U}\},$$

where $\mathbf{U}$ represents the input sequence limits, and will be discussed later.

A no evident consequence of the theoretical concept of perfect control is that only a controller that takes into account the input constraints could be capable of actually approach the perfect control, i.e. to approximate the perfect control up to the point where some of the constraints become active. A controller which does not account for constraints can maintain the system apart from those limits by means of a conservative tuning only. This fact open the possibility to apply a constrained Model Predictive Control (MPC) strategy to account for this kind of problems.

1.2 MPC overview

As was already said, a promising strategy to be used to approach good performances in an iterative learning scheme is the constrained model predictive control, or receding horizon control. This strategy solves, at each time step, an optimization problem to obtain the control action to be applied to the system at the next time. The optimization attempt to minimizes the difference between the desired variable trajectories and a forecast of the system variables, which is made based on a model, subject to the variable constraints (Camacho & Bordons, 2009). So, the first stage to design an MPC is to choose a model. Here, the linear model will be given by:

$$x_{k+1} = Ax_k + Bu_k \tag{3}$$

$$d_{k+1} = d_k \tag{4}$$

$$y_k = Cx_k + d_k \tag{5}$$

where $x_k \in \mathbb{R}^n$ is the state at time k, $u_k \in \mathbb{R}^m$ is the manipulated input, $y_k \in \mathbb{R}^l$ is the controlled output, A, B and C are matrices of appropriate dimension, and $d_k \in \mathbb{R}^l$ is an integrating output disturbance (González, Adam & Marchetti, 2008).

Furthermore, and as a part of the system description, input (and possibly input increment) constraints are considered in the following inclusion:

$$u \in U, \tag{6}$$

where U is given by:

$$U = \{u \in \mathbb{R}^m : u_{min} \leq u \leq u_{max}\}.$$

A simplified version of the optimization problem that solves on-line (at each time k) a typical stable MPC is as follows:

Problem P0

$$\min_{\{u_{k|k},\ldots,u_{k+N-1|k}\}} V_k = \sum_{j=0}^{N-1} \ell\left(e_{k+j|k}, u_{k+j|k}\right) + F\left(e_{k+N|k}\right)$$

subjet to:

$$\begin{aligned} e_{k+j|k} &= Cx_{k+j|k} + d_{k+j} - y^r_{k+j}, & j &= 0,\ldots,N, \\ x_{k+j+1|k} &= Ax_{k+j|k} + Bu_{k+j|k}, & j &= 0,\ldots,N-1, \\ u_{k+j|k} &\in U, & j &= 0,1,\ldots,N-1, \end{aligned}$$

where $\ell(e,u) := ||e||^2_Q + ||u||^2_R$, $F(e) := ||e||^2_P$. Matrices Q and R are such that $Q > 0$ and $R \geq 0$. Furthermore, a terminal constraint of the form $x_{k+N|k} \in \Omega$, where Ω is a specific set, is usually included to assure stability. In this general context, some conditions should be fulfilled by the different "*components*" of the formulation (i.e., the terminal matrix penalization P, the terminal set, Ω, etc) to achieve the closed loop stability and the recursive feasibility [2] ((Rawlings and Mayne, 2009)). In the next sections, this basic formulation will be modified to account for learning properties in the context of repetitive systems.

2. Preliminaries

2.1 Problem index definition

As was previously stated, the control strategy proposed in this chapter consists of a basic MPC with learning properties. Then, to clarify the notation to be used along the chapter (that comes form the ILC and the MPC literature), we start by defining the following index variables:

- i: is the iteration or run index, where $i = 0$ is the first run. It goes from 0 to ∞.

[2] Recursive feasibility refers to the guarantee that once a feasible initial condition is provided, the controller will guide the system trough a feasible path

- k: is the discrete time into a single run. For a given run, it goes from 0 to T_{f-1} (that is, T_f time instants).
- j: is the discrete time for the MPC predictions. For a given run i, and a given time instant k, it goes from 0 to $H = T_f - k$. To clearly state that j represents the time of a prediction made at a given time instant k, the notation $k + j|k$, wich is usual in MPC literature, will be used.

The control objective for an individual run i is to find an input sequence defined by

$$\mathbf{u}^i := \left[u_0^{i^T} \dots u_{T_f-1}^{i^T} \right]^T \tag{7}$$

which derives in an output sequence

$$\mathbf{y}^i = \left[y_0^{i^T} \dots y_{T_f}^{i^T} \right]^T \tag{8}$$

as close as possible to a output reference trajectory

$$\mathbf{y}^r := \left[y_0^{r\,T} \dots y_{T_f}^{r\,T} \right]^T. \tag{9}$$

Furthermore, assume that for a given run i there exists an input reference sequence (an input candidate) given by

$$\mathbf{u}^{i^r} := \left[u_0^{i^r\,T} \dots u_{T_f-1}^{i^r\,T} \right]^T \tag{10}$$

and that the output disturbance profile,

$$\mathbf{d}^i = \left[d_0^{i^T} \dots d_{T_f}^{i^T} \right]^T,$$

is known. During the learning process the disturbance profile is assumed to remain unchanged for several operations. Furthermore, the value $u_{T_f-1}^{i^r}$ represents a stationary input value, satisfying $u_{T_f-1}^r = G^{-1}(y_{T_f}^r - d_{T_f})^i$, for every i, with $G = [C(I - A)^{-1}B]$.

2.2 Convergence analysis

In the context of repetitive systems, we will consider two convergence analyses:

DEFINITION 0.3 (Intra-run convergence). *It concerns the decreasing of a Lyapunov function (associated to the output error) along the run time k, that is,* $V\left(y_{k+1}^i - y_{k+1}^r\right) \leq V\left(y_{k+1}^i - y_k^r\right)$ *for* $k = 1, \dots, T_{f-1}$*, for every single run. If the execution of the control algorithm goes beyond* T_f*, with* $k \to \infty$*, and the output reference remains constant at the final reference value (*$y_k^r = y_{T_t}^r$ *for* $T_f \leq k < \infty$*) then the intra-run convergence concerns the convergence of the output to the final value of the output reference trajectory* $\left(y_{k+1}^i \to y_k^r \text{ as } k \to \infty\right)$*. This convergence was proved in (González et al., 2009a) and presented in this chapter.*

DEFINITION 0.4 (Inter-run convergence). *It concerns the convergence of the output trajectory to the complete reference trajectory from one run 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

In this subsection, a first approach to a new MPC design, which includes learning properties, is presented. It will be assumed that an appropriate input reference sequence $\mathbf{u}^{i^r}$ is available (otherwise, it is possible to use a null constant value), and the disturbance d_k^i as well as the states $x_{k|k}^i$ are estimated. Given that the operation lasts T_f time instants, it is assumed here a *shrinking output horizon* defined as the distance between the current time k and the final time T_f, that is, $H := T_f - k$ (See Figure 2). Under these assumptions the optimization problem to be solved at time k, as part of the single run i, is described as follows:

Problem P1

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

subjet to:

$$\begin{aligned}
e_{k+j|k}^i &= Cx_{k+j|k}^i + d_{k+j}^i - y_{k+j}^r, \qquad j = 0,\ldots,H,\\
x_{k+j+1|k}^i &= Ax_{k+j|k}^i + Bu_{k+j|k}^i, \qquad j = 0,\ldots,H-1,\\
u_{k+j|k}^i &\in U, \qquad j = 0,1,\ldots,H-1, \qquad (11)\\
u_{k+j|k}^i &= u_{k+j}^{i^r} + \overline{u}_{k+j|k}^i, \qquad j = 0,1,\ldots,H-1, \qquad (12)\\
\overline{u}_{k+j|k}^i &= 0, \qquad j \geq N_s, \qquad (13)
\end{aligned}$$

where the (also shrinking) control horizon N_s is given by $N_s = \min(H, N)$ and N is the fixed control horizon introduced before (it is in fact a controller parameter). Notice that predictions with indexes given by $k + H|k$, which are equivalent to $T_f|k$, are in fact prediction for a fixed future time (in the sense that the horizon does not recedes). Because this formulation contains some new concepts, a few remarks are needed to clarify the key points:

Remark 0.1. *In the ith-operation, T_f optimization problems P1 must be solved (from $k = 0$ to $k = T_f - 1$). Each problem gives an optimal input sequence $u_{k+j|k}^{i^{opt}}$, for $j = 0,\cdots,H-1$, and following the typical MPC policy, only the first input of the sequence, $u_{k|k}^{i^{opt}}$, is applied to the system.*

Remark 0.2. *The decision variables $\overline{u}_{k+j|k}^i$, are a correction to the input reference sequence $u_{k+j}^{i^r}$(see Equation (12)), attempting to improve the closed loop predicted performance. $u_{k+j}^{i^r}$ can be seen as the control action of an* underlying stabilizing controller *acting along the whole output horizon, which could be corrected, if necessary, by the control actions $\overline{u}_{k+j|k}^i$. Besides, because of constraints (13), $\overline{u}_{k+j|k}^i$ is different from zero only in the first N_s steps (or predictions) and so, the optimization problem P1 has N_s decision variables (See Figure 2). All along every single run, the input and output references, $u_{k+j}^{i^r}$ and y_{k+j}^r , as well as the disturbance d_{k+j}^i may be interpreted as a set of fixed parameters.*

Remark 0.3. *The convergence analysis for the operation sequence assumes that once the disturbance appears it remains unchanged for the operations that follow. In this way the cost remains bounded despite it represents an infinite summation; this happens because the model used to compute the predictions leads to a final input (and state) that matches $(y_{T_f}^r - d_{T_f}^i)$. Thus, the model output is guided to $(y_{T_f}^r - d_{T_f}^i)$, and the system output is guided to $y_{T_f}^r$.*

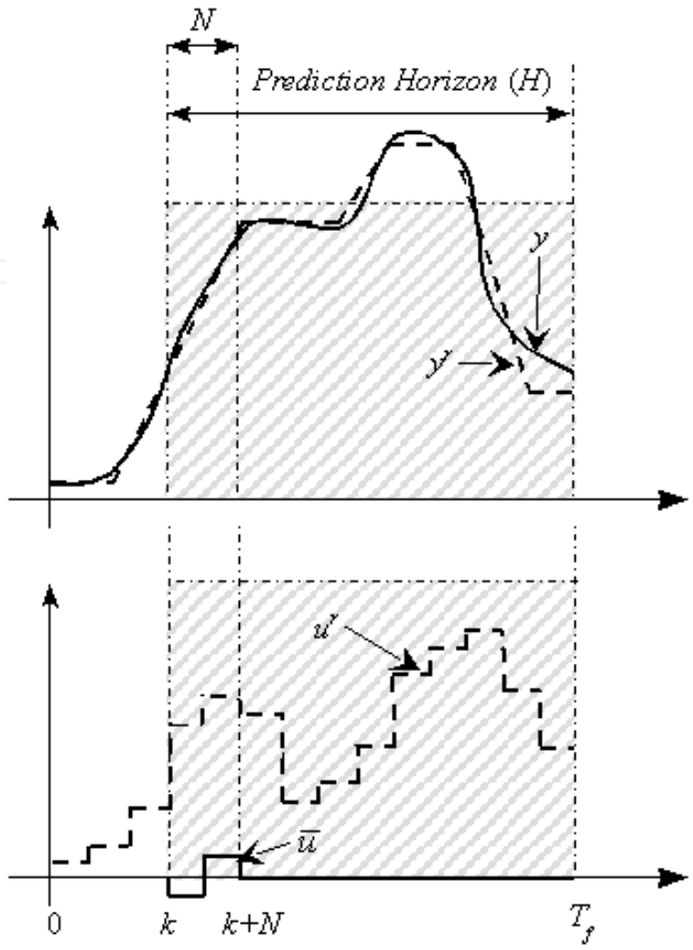


Fig. 2. Diagram representing the MPC optimization problem at a given time k.

3.1 Decreasing properties of the closed-loop cost for a single run

The concept of stability for a finite-duration process is different from the traditional one since, except for some special cases such finite-time escape, boundless of the disturbance effect is trivially guaranteed. In (Srinivasan & Bonvin, 2007), the authors define a quantitative concept of stability by defining a variability index as the induced norm of the variation around a reference (state) trajectory, caused by a variation in the initial condition. Here, we will show two controller properties (Theorem 0.1). 1) The optimal IHMPC cost monotonically decreases w.r.t time k, and 2) if the control algorithm execution goes beyond T_f with $k \to \infty$, and the output reference remains constant at the final reference value ($y^r_k = y^r_{T_f}$ for $k \geq T_f$) then, the IHMPC cost goes to zero as $k \to \infty$, which implies that $y^i_k \to y^r_{T_f}$ as $k \to \infty$.

Theorem 0.1 (intra-run convergence)**.** *Let assume that the disturbance remains constant from one run to the next. Then, for the system (3-5), and the constraint (6), by using the control law derived from the on-line execution of problem P1 in a shrinking horizon manner, the cost is decreasing, that is,* $V^{i^{opt}}_k - V^{i^{opt}}_{k-1} + \ell(e^i_{k-1}, \overline{u}^i_{k-1}) \leq 0$, *for* $0 \leq k \leq T_f - 1$.

Furthermore, the last cost of a given operation "i" is given by:

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

and since current and one steps predictions are coincident with the actual values, it follows that:

$$V_{T_f-1}^{i^{opt}} = \ell(e_{T_f-1}^{i^{opt}}, \overline{u}_{T_f-1}^{i^{opt}}) + F(e_{T_f}). \tag{14}$$

Proof See the Appendix. □

Remark 0.4. *The cost $V_k^{i^{opt}}$ of Problem P1 is not a strict Lyapunov function, because the output horizon is not fixed and then, $V_k^{i^{opt}}(e_k^i)$ changes as k increases (in fact, as k increases the cost becomes less demanding because the output horizon is smaller). However, if a* virtual infinite output horizon for predictions *is defined, and stationary values of output and input references are assumed for $T_f < \infty$ (i.e. $u_{ss}^{i^r} = (C(I-A)^{-1}B)^{-1}(y_{ss}^r - d_{ss}^i)$, where d_{ss}^i is the output disturbance at T_f), then by selecting the terminal cost $F(e_{T_f|k}^i)$ to be the sum of the stage penalization $\ell(\cdot,\cdot)$ from T_f to ∞, it is possible to associate $V_k^{i^{opt}}(e_k^i)$ with a fixed (infinite) output horizon. In this way $V_k^{i^{opt}}(e_k^i)$ becomes a Lyapunov function since it is an implicit function of the actual output error e_k^i. To make the terminal cost the infinite tail of the output predictions, it must be defined as*

$$F(e_{T_f|k}^i) = \left\| Cx_{T_f|k}^i + d_{ss}^i - y_{ss}^r \right\|_P^2 = \left\| x_{T_f|k}^i + x_{ss}^{i^r} \right\|_{C^TPC}^2 = \sum_{j=T_f}^{\infty} \left\| x_{j|k}^i + x_{ss}^{i^r} \right\|_{C^TQC}^2$$

$$i = 0, 1, \ldots, T_f - 1,$$

where $x_{ss}^{i^r} = (I-A)^{-1}Bu_{ss}^{i^r}$ and C^TPC is the solution of the following Lyapunov equation: $A^TC^TPCA = C^TPC - C^TQC$. With this choice of the terminal matrix P, the stability results of Theorem 0.1 is stronger since the closed loop becomes Lyapunov stable.

3.2 Discussion about the stability of the closed-loop cost for a single run

Theorem 0.1, together with the assumptions of Remark 0.4, shows convergence characteristics of the Lyapunov function defined by the IHMPC strategy. These concepts can be extended to determine a variability index in order to establish a quantitative concept of stability (β-stability), as it was highlighted by (Srinivasan & Bonvin, 2007). To formulate this extension, the MPC stability conditions (rather than convergence conditions) must be defined, following the stability results presented in ((Scokaert et al., 1997)). An extension of this remark is shown below.

First, we will recall the following exponential stability results.

Theorem 0.2 ((Scokaert et al., 1997))**.** *Let assume for simplicity that state reference x_k^r is provided, such that $y_k^r = Cx_k^r$, for $k = 0, \ldots, T_f - 1$, and no disturbance is present. If there exist constants a_x, a_u, b_u, c_x, c_u and d_x such that the stage cost $\ell(x,u)$, the terminal cost $F(x)$, and the model matrices A, B and C, in Problem P1, fulfill the following conditions:*

$$\underline{\gamma}.\|x\|^{\sigma} \leq \ell(x,u) = \|x\|_Q^2 + \|u\|_R^2 \leq c_x.\|x\|^{\sigma} + c_u.\|u\|^{\sigma} \tag{15}$$

$$\|u_{k+j|k}^{i^{opt}}\| \leq b_u\|x_k\|, \ \ for \ \ j = 0, \ldots, H-1 \tag{16}$$

$$\|Ax + Bu\| \leq a_x\|x\| + a_u\|u\| \tag{17}$$

$$F(x) \leq d_x\|x\|^{\sigma} \tag{18}$$

then, the optimal cost $V_k^{i^{opt}}(x_k)$ *satisfies*

$$\underline{\gamma}.\|x_k\|^{\sigma} \leq V_k^{i^{opt}}(x_k) \leq \overline{\gamma}.\|x_k\|^{\sigma} \tag{19}$$

$$V_k^{i^{opt}}(x_k) \leq -\underline{\gamma}.\|x_k\|^{\sigma} \tag{20}$$

with $\overline{\gamma} = \left(c_x.\sum_{i=0}^{N-1}\alpha_i^{\sigma} + N.c_u.b_u^{\sigma} + d_x.\alpha_N^{\sigma}\right)$, $\alpha_j = a_x.\alpha_{j-1} + a_u.b_u$ *and* $\alpha_0 = a_x + a_u.b_u$.

Proof The proof of this theorem can be seen in (Scokaert et al., 1997). □

Condition (15) is easy to determine in terms of the eigenvalues of matrices Q and R. Condition (16), which are related to the Lipschitz continuity of the input, holds true under certain regularity conditions of the optimization problem.

Now, we define the following *variability index*, which is an induced norm, similar to the one presented in (Srinivasan & Bonvin, 2007):

$$\xi = \max_{V_0^{i^{opt}}=\delta} \left(\frac{\sum_{k=0}^{T_f-1} V_k^{i^{opt}}}{V_0^{i^{opt}}} \right)$$

for a small value of $\delta > 0$. With the last definition, the concept of β -stability for finite-duration systems is as follows.

DEFINITION 0.5 ((Scokaert et al., 1997)). *The closed-loop system obtained with the proposed IHMPC controller is intra-run* β*-stable around the state trajectory* x_k^r *if there exists* $\delta > 0$ *such that* $\xi \leq \beta$.

Theorem 0.3 (quantitative β -stability)**.** *Let assume for simplicity that a state reference,* x_k^r *, is provided, such that* $y_k^r = Cx_k^r, k = 0, \ldots, T_f - 1$, *and no disturbance is present. If there exist constants* a_x, a_u, b_u, c_x, c_u *and* d_x *as in Theorem 0.2, then, the closed-loop system obtained with system(3) -(5) and the proposed IHMPC controller law is intra-run* β*-stable around the state trajectory* x_k^r *, with*

$$\beta = \frac{\left[\overline{\gamma} + \sum_{n=1}^{T_f-1}\left(\overline{\gamma} - \underline{\gamma}\right)^n\right]}{\underline{\gamma}}.$$

Proof See the Appendix. □

4. IHMPC with learning properties

In the last section we studied the single-operation control problem, where we have assumed that an input reference is available and the output disturbance is known. However, one alternative is defining the input reference and disturbance as the input and disturbance obtained during the last operation (i.e. the last implemented input and the last estimated disturbance, beginning with a constant sequence and a zero value, respectively). In this way,

a dual MPC with learning properties accounting for the operations sequence control can be derived. The details of this development are presented next.

4.1 Additional MPC constraints to induce learning properties

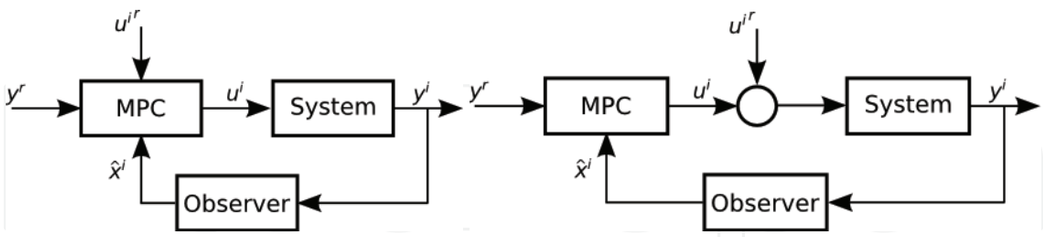
For a given operation i, consider the problem P1 with the following additional constraints:

$$\begin{aligned} u^{i^r}_{k+j} &= u^{i-1^{opt}}_{k+j|k+j}, & k &= 0,\dots,T_f-1, & j &= 0,\dots,H-1 \\ d^{i}_{k+j} &= \hat{d}^{i-1}_{k+j}, & k &= 1,\dots,T_f, & j &= 0,\dots,H \end{aligned} \tag{21}$$

where $\hat{d}^{i-1}_{k+j}$ represents the disturbance estimation. The first constraint requires updating the input reference for operation i with the last optimal sequence executed in operation $i-1$ (i.e. $\mathbf{u}^{i^r} = \mathbf{u}^{i-1}$, for $i = 1,2,\cdots$, with an initial value given by $\mathbf{u}^0 := [G^{-1}y^r_{T_f} \ \cdots \ G^{-1}y^r_{T_f}]$). The second one updates the disturbance profile for operation i with the last estimated sequence in operation $i-1$ (i.e. $\mathbf{d}^i = \hat{\mathbf{d}}^{i-1}$, for $i = 1,2,\cdots$, with an initial value given by $\mathbf{d}^0 = [0 \ \cdots \ 0]$).

Besides, notice that the vector of differences between two consecutive control trajectories, $\boldsymbol{\delta}^i := \mathbf{u}^i - \mathbf{u}^{i-1}$, is given by $\boldsymbol{\delta}^i = \left[\overline{u}^{i^{opt^T}}_{0|0} \ \cdots \ \overline{u}^{i^{opt^T}}_{T_f-1|T_f-1}\right]$, i.e., the elements of this vector are the collection of first control movements of the solutions of each optimization problem P1, for $k = 0,\cdots,T_f-1$.

Remark 0.5. *The input reference update, together with the correction presented in Remark 0.2, has the following consequence: the learning procedure is not achieved by correcting the implemented input action with past information but, by correcting the predicted input sequence with the past input profile , which represents here the learning parameter. In this way better output forecast will be made because the optimization cost has predetermined input information. Figure 3 shows the difference between these two learning procedures.*



(a) Proposed learning procedures. (b) Typical learning procedures.

Fig. 3. Learning procedures.

Remark 0.6. *The proposed disturbance update implies that the profile estimated by the observer at operation $i-1$ is not used at operation $i-1$, but at operation i. This disturbance update works properly when the disturbance remains unmodified for several operations, i.e., when permanent disturbances, or model mismatch, are considered. If the disturbance substantially changes from one operation to next (that is, the disturbance magnitude or the time instant in which the disturbance enter the system change), it is possible to use an additional "current" disturbance correction given by . This correction is then added to permanent disturbance profile at each time k of the operation i.*

4.2 MPC formulation and proposed run cost

Let us consider the following optimization problem:

Problem P2

$$\min_{\{\overline{u}^i_{k|k},\ldots,\overline{u}^i_{k+T_f-1|k}\}} V^i_k$$

subject to (3 - 13) and (21): Run to run convergence means that both, the output error trajectory $\mathbf{e}^i$ and the input difference between two consecutive implemented inputs, $\boldsymbol{\delta}^i = \mathbf{u}^i - \mathbf{u}^{i-1}$, converges to zero as $i \to \infty$. Following an Iterative Learning Control nomenclature, this means that the implemented input, $\mathbf{u}^i$, converges to the perfect control input $\mathbf{u}^{perf}$.

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:

$$J_i := \sum_{k=0}^{T_f-1} V_k^{i^{opt}}, \tag{22}$$

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, once the optimization problem P2 is solved and an optimal input sequence is obtained, this MPC cost is a function of only $e^{i^{opt}}_{k|k} = \left(y^{i^{opt}}_{k|k} - y^r_k\right) = e^i_k$. Therefore, it makes sense using (22) to define a run cost, since it represents a (finite) sum of positive penalizations of the current output error, i.e., a positive function of e^i. However, since the new run 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). Then, we should demonstrate that the perfect control input corresponds to a null cost. These two properties, together with an additional one, are presented in the next subsection.

4.3 Some properties of the formulation

One interesting point is to answer what happens if the MPC controller receives as input reference trajectory the *perfect control sequence* presented in the first section. The simplest answer is to associate this situation with a null MPC cost. However, since the proposed MPC controller does not add the input reference (given by the past control profile) to the implemented inputs but to the predicted ones, some care must be taken. Property 0.1, below, assures that for this input reference the MPC cost is null. Without loss of generality we consider in what follows that no disturbances enter the system.

Property 0.1. *If the MPC cost penalization matrices, Q and R, are definite positive ($Q \succ 0$ and $R \succ 0$) and the perfect control input trajectory is a feasible trajectory, then $\boldsymbol{u}^{i^r} = \boldsymbol{u}^{perf} \Leftrightarrow V_k^{i^{opt}} = 0$ for $k = 0, \ldots, T_f - 1$; where*

$$V_k^{i^{opt}} = \sum_{j=0}^{H-1} l\left(e^{i^{opt}}_{k+j|k}, \overline{u}^{i^{opt}}_{k+j|k}\right) + F\left(x^{i^{opt}}_{k+H|k}\right).$$

Proof See the Appendix. □

This property allow as to formulate the following one:

Property 0.2. *If the MPC cost penalization matrices, Q and R, are definite positive ($Q \succ 0$ and $R \succ 0$) and perfect control input trajectory is a feasible trajectory, cost (12), which is an implicit function of e^i, is such that, $e^i = 0 \Leftrightarrow J_i = 0$.*

Proof See the Appendix. □

Finally, as trivial corollary of the last two properties, it follows that:

Property 0.3. *If the MPC cost penalization matrices, Q and R, are definite positive, then $u^{ir} = u^{perf} \Leftrightarrow J^i = 0$. Otherwise, $u^{ir} \neq u^{perf} \Rightarrow J^i \neq 0$.*

Proof It follows from Property 0.1 and Property 0.2. □

4.4 Main convergence result

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

Theorem 0.4. *For the system (3)-(5), by using the control law derived from the on-line execution of problem P2 in a shrinking horizon manner, together with the learning updating (21), 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 $\boldsymbol{u}^{ir}$ converges to $\boldsymbol{u}^{perf}$.*

Remark 0.7. *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 previous proof), it will converge to the smallest possible value. Given that, as was already said, the impossibility to reach perfect control is exclusively related to the input and/or states limits (which should be consistent with the control problem constraints), the proposed strategy will find the best approximation to the perfect control, which constitutes an important advantage of the method.*

Remark 0.8. *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-duration systems (Theroem 0.3); the inter-run convergence can be extended to establish stability conditions similar to the ones presented in (Srinivasan & Bonvin, 2007).*

5. Illustrative examples

Example 1. In order to evaluate the proposed controller performance, we consider first a linear system (Lee & Lee, 1997) given by $G(s) = 1/15s^2 + 8s + 1$. The MPC parameters were tuned as $Q = 1500$, $R = 0.5$ and $T = 1$. Figure 4 shows the obtained performance in the controlled variable where the difference with the reference is undistinguished. Given that the problem assumes that no information about the input reference is available, the input sequence u and u are equals.

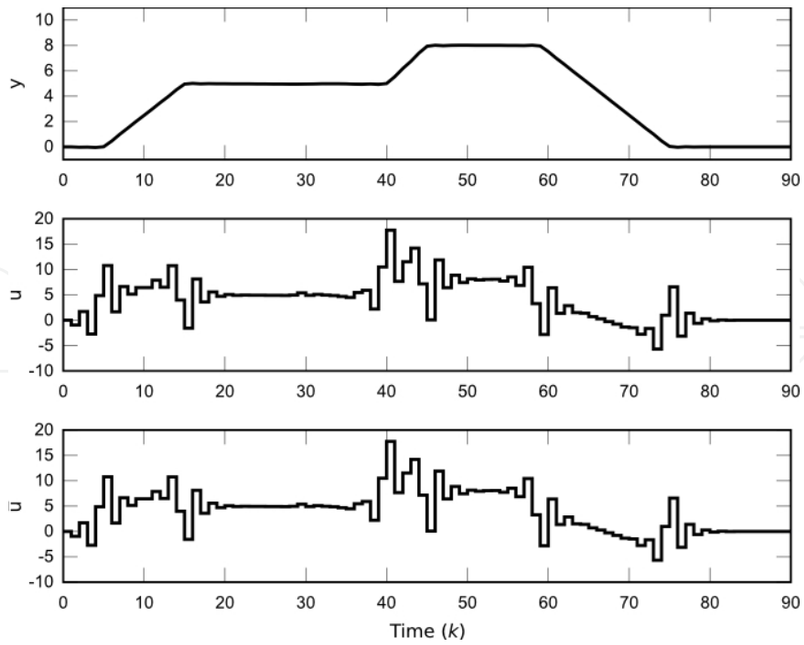


Fig. 4. Reference, output response according to the input variables u and $\overline{u}$

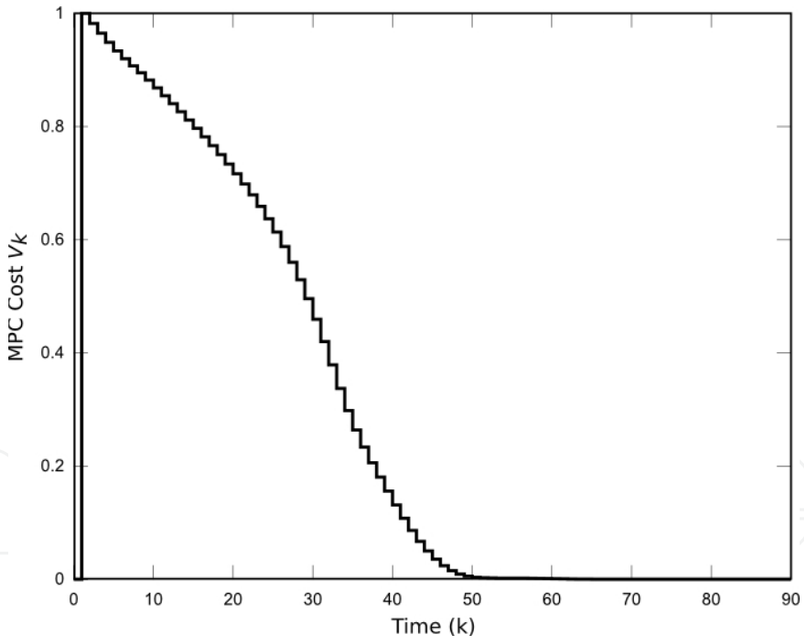


Fig. 5. Normalized MPC cost function. Here, the normalized cost function is obtained as $V_k / V_{k\max}$.

The MPC cost function is showed in Fig. 5. According to the proof of Theorem 0.1 (nominal case), this cost function is monotonically decreasing.

Example 2. Consider now a nonlinear-batch reactor where an exothermic and irreversible chemical reaction takes place, (Lee & Lee, 1997). The idea is to control the reactor temperature

by manipulating the inlet coolant temperature. Furthermore, the manipulated variable has minimum and maximum constrains given by: $Tc_{\min} \leq Tc \leq Tc_{\max}$, where $Tc_{\min} = -25[^{\circ}C]$, $Tc_{\max} = 25[^{\circ}C]$ and, Tc is written in deviation variable. In addition, to show how the MPC controller works, it is assumed that a previous information about the cooling jacked temperature ($u = Tc$) is available.

Here the proposed MPC was implemented and the MPC parameters were tuned as, $Q = 1000$, $R = 5$ and $T = 1$[min]. The nominal linear model used for predictions is the same proposed by (Adam, 2007).

Figure 6 shows both the reference and the temperature of the batch reactor are expressed in deviation variable. Furthermore, the manipulated variable and the correction made by the MPC, u are shown.

Notice that, 1) the cooling jacked temperature reaches the maximum value and as a consequence the input constraints becomes active in the time interval from 41 minutes to 46 minutes; 2) similarly, when the cooling jacked temperature reaches the minimum value, the other constraint becomes active in the time interval from 72 minutes to 73 minutes; 3) the performance is quite satisfactory in spite of the problem is considerably nonlinear and, 4) given that it is assumed that a previous information about the cooling jacked temperature is available, the correction u is somewhat smaller than u (Fig. 6).

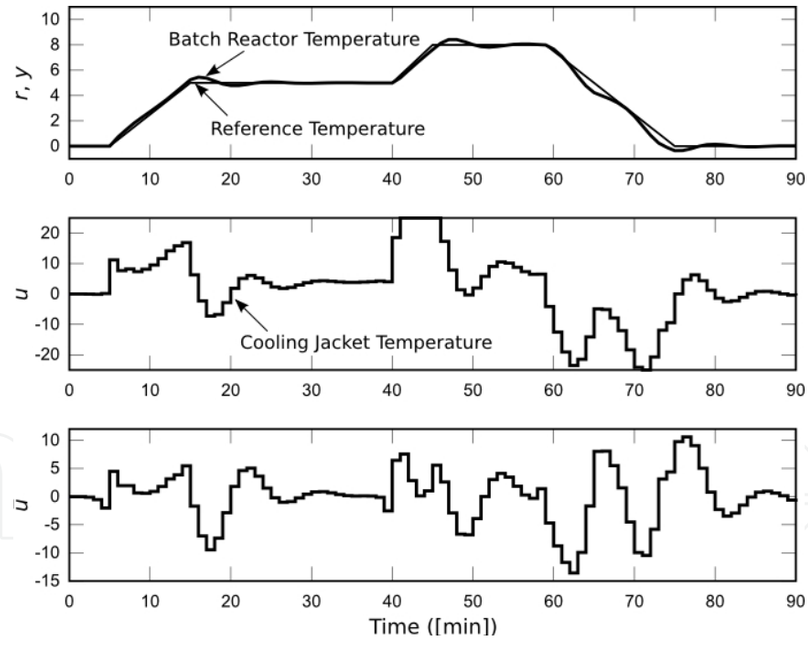


Fig. 6. Temperature reference and controlled temperature of the batch reactor. Also, the cooling jacked temperature (u) and the correction ($\overline{u}$) are showed.

Example 3. In order to evaluate the proposed controller performance we assume a true and nominal process given by (Lee et al., 2000; Lee & Lee, 1997) $G(s) = 1/15s^2 + 8s + 1$ 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

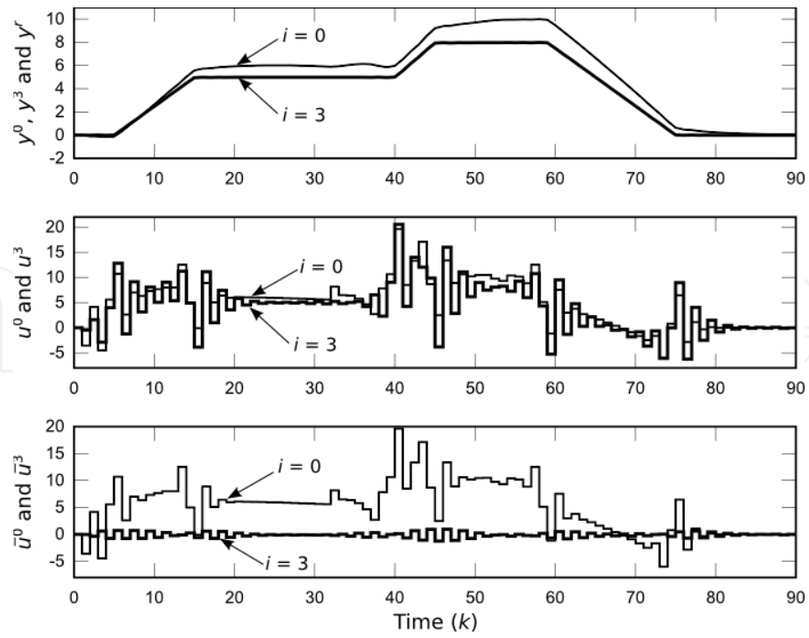
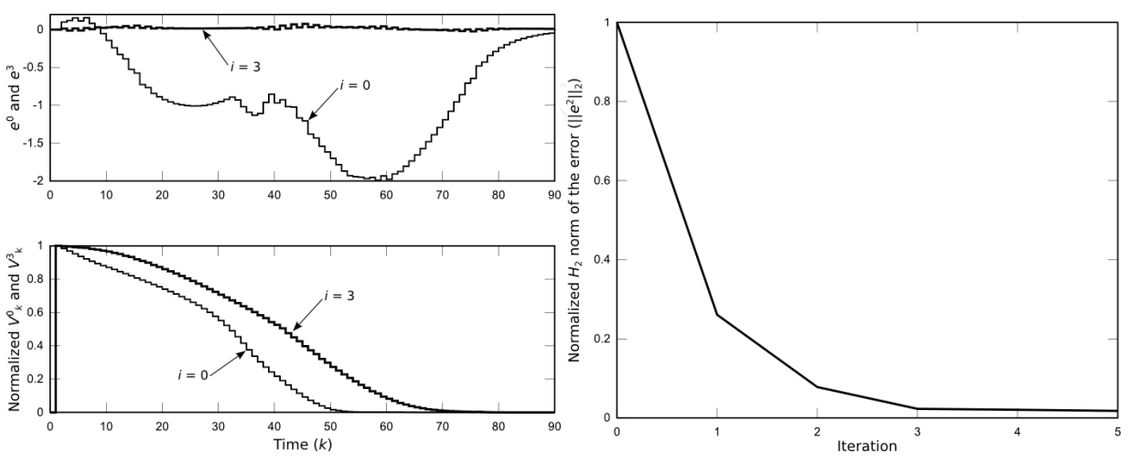


Fig. 7. Output and input responses.

reduced control horizon. The controller parameters are as follows: $Q = 1500$, $R = 0.05$, $N = 5$. Figure 7 shows the output response together with the output reference, and the inputs u^i and $\overline{u}^i$, for the first and third iteration. At the first iteration, since the input reference is a constant value ($u^{i^r}_{T_f-1} = 0$), u^i and $\overline{u}^i$ 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 0.1) for each run (Fig. 8a). Notice that the MPC cost is normalized taking into account the maximal value $\left(V^i_k / V^i_{\max}\right)$,



(a) Error and MPC cost. (b) Norm of the iteration error.

Fig. 8. Error and MPC cost, and Norm of the iteration error for the example 3.

where $V^1_{\max} \approx 1.10^6$ and $V^1_{\max} \approx 286.5$. This shows that the MPC cost J^i decrease from one run to the next, as was stated in Theorem 0.4. Finally, Fig. 8b shows the normalized norm of the error corresponding to each run.

6. Conclusion

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.

7. Appendix

Proof of Theorem 0.1

Proof Let $\overline{\mathbf{u}}^{i^{opt}}_{k-1} := \left\{ \overline{u}^{opt}_{k-1|k-1}, \ldots, \overline{u}^{i^{opt}}_{k+N_s-2|k-1}, 0, \ldots, 0 \right\}$ and $\mathbf{x}^{i^{opt}}_{k-1} := \left\{ x^{i^{opt}}_{k-1|k-1}, \ldots, x^{i^{opt}}_{k+T_f|k-1} \right\}$ be the optimal input and state sequence that are the solution to problem P1 at time $k-1$, with $k = 1, \cdots, T_f - N$ (that means that the last N optimization problem of a given run i are not considered). The cost corresponding to these variables are

$$\begin{aligned} V^{i^{opt}}_{k-1} &= \sum_{j=0}^{N_s-1} \ell\left(e^{i^{opt}}_{k+j-1|k-1}, \overline{u}^{i^{opt}}_{k+j-1|k-1}\right) + \sum_{j=N_s}^{H-1} \ell\left(e^{i^{opt}}_{k+j-1|k-1}, 0\right) + F\left(x^{i^{opt}}_{k+H-1|k-1}\right) \\ &= \sum_{j=0}^{H-1} \ell\left(e^{i^{opt}}_{k+j-1|k-1}, \overline{u}^{i^{opt}}_{k+j-1|k-1}\right) + F\left(x^{i^{opt}}_{T_f|k-1}\right). \end{aligned} \tag{23}$$

Notice that at time $k-1$, $H = T_f - k + 1$, since H is a shrinking horizon. Now, let $\overline{\mathbf{u}}^{i^{feas}}_{k} := \left\{ \overline{u}^{i^{opt}}_{k|k-1}, \ldots, \overline{u}^{opt}_{k+N_s-2|k-1}, 0, \ldots, 0 \right\}$ be a feasible solution to problem P1 at time k. Since no new input is injected to the system from time $k-1$ to time k, and no unknown disturbance is considered, the predicted state at time k, using the feasible input sequence, will be given by $\overline{\mathbf{x}}^{i^{feas}}_{k} := \left\{ x^{i^{opt}}_{k|k-1}, \ldots, x^{i^{opt}}_{k+H-1|k-1}, x^{i^{opt}}_{k+H|k-1} \right\} = \left\{ x^{i^{opt}}_{k|k-1}, \ldots, x^{i^{opt}}_{k+T_f|k-1}, x^{i^{opt}}_{k+T_f+1|k-1} \right\}$. Then, the cost at time k corresponding to the feasible solution $\overline{\mathbf{u}}^{i^{feas}}$ is as follows:

$$\begin{aligned} V^{i^{feas}}_{k} &= \sum_{j=0}^{N_s-1} \ell\left(e^{i^{opt}}_{k+j|k-1}, \overline{u}^{i^{opt}}_{k+j|k-1}\right) + \sum_{j=N_s}^{H-1} \ell\left(e^{i^{opt}}_{k+j|k-1}, 0\right) + F\left(e^{i^{opt}}_{k+H|k-1}\right) \\ &= \sum_{j=0}^{H-1} \ell\left(e^{i^{opt}}_{k+j|k-1}, \overline{u}^{i^{opt}}_{k+j|k-1}\right) + F\left(e^{i^{opt}}_{T_f|k-1}\right). \end{aligned} \tag{24}$$

Notice that now $H = T_f - k$, because predictions are referred to time k. Now, subtracting (23) from (24) we have

$$V^{i^{feas}}_{k} - V^{i^{opt}}_{k-1} = -\ell\left(e^{i^{opt}}_{k-1|k-1}, \overline{u}^{i^{opt}}_{k-1|k-1}\right).$$

This means that the optimal cost at time k, which is not grater than the feasible one at the same time, satisfies

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

Finally, notice that $e_{k-1|k-1}^{i^{opt}}$ and $\overline{u}_{k-1|k-1}^{i^{opt}}$ represent actual (not only predicted) variables. Thus, we can write

$$V_k^{i^{opt}} - V_{k-1}^{i^{opt}} + \ell\left(e_{k-1}^{i^{opt}}, \overline{u}_{k-1}^{i^{opt}}\right) \leq 0. \tag{25}$$

This shows that, whatever the output error is different from zero, the cost decreases when time k increases.

Finally, the decreasing property for $k = T_f - N + 1, \cdots, T_f - 1$, and the last part of the theorem, can be proved following similar steps as before (i.e., finding a feasible solution). □

Proof of Theorem 0.3

Proof From the recursive use of (25), together with (15), (19) and (20), we have

$$V_{k+1}^{i^{opt}} \leq V_k^{i^{opt}} - l\left(\overline{x}_k^i, \overline{u}_k^i\right) \leq \overline{\gamma}.\|\overline{x}_k^i\|^\sigma - \underline{\gamma}.\|\overline{x}_k^i\|^\sigma = (\overline{\gamma} - \underline{\gamma})\|\overline{x}_k^i\|^\sigma,$$

for $k = 0, \ldots, T_f - 2$. So we can write:

$$\sum_{k=0}^{T_f-1} V_k^{i^{opt}} \leq \left[r + \sum_{n=1}^{T_f-1} (\overline{\gamma} - \underline{\gamma})^n\right].\|\overline{x}_0^i\|^\sigma.$$

Therefore,

$$\frac{\sum_{k=0}^{T_f-1} V_k^{i^{opt}}}{V_0^{i^{opt}}} \leq \frac{\left[\overline{\gamma} + \sum_{n=1}^{T_f-1}(\overline{\gamma} - \underline{\gamma})^n\right]}{\underline{\gamma}},$$

since $\underline{\gamma}.\|x_0\|^\sigma$ is a lower bound of $V_0^{i^{opt}}$ (that is, $\underline{\gamma}.\|x_0\|^\sigma \leq V_0^{i^{opt}}$).

Finally,

$$\beta = \frac{\left[\overline{\gamma} + \sum_{n=1}^{T_f-1}\left(\overline{\gamma} - \underline{\gamma}\right)^n\right]}{\underline{\gamma}}.$$

□

Proof of Property 0.1

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 $e_{k+j|k}^{i^{opt}} = 0$, $j = 0, ..., T_f$ and $\overline{u}_{k+j|k}^{i^{opt}} = 0$, for $j = 0, ..., T_f - 1$, respectively. If $e_{k+j|k}^{i^{opt}}$ adn $\overline{u}_{k+j|k}^{i^{opt}} = 0$ simultaneously, it follows that $u_k^{i^r} = u_k^{perf}$ for $k = 0, \ldots, T_f - 1$, 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 $Q \succ 0$ and $R \succ 0$ by hypothesis). Consequently, $\mathbf{u}^{i^r} = \mathbf{u}^{perf}$.

$\Rightarrow$) Let us assume that $\mathbf{u}_k^{i^r} = \mathbf{u}_k^{perf}$. 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

$$e_{k|k}^i = 0$$

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

$$\vdots$$

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

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

Proof of Property 0.2

Proof $\Rightarrow$) Let us assume that $\mathbf{e}^i = 0$. This means that $e_{k|k}^{i^{opt}} = 0$, for $k = 0, \ldots, T_f$. Now, assume that the input reference vector is different from the perfect control input, $\mathbf{u}^{ir} \neq \mathbf{u}^{perf}$, 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^{i^r} is not an element of the perfect control input, then $\left[Ax_{k|k}^i + Bu_k^{i^r}\right] \neq 0$. 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 $\|e_{k+1|k}^{i^{opt}}\| > 0$ and $\left\|\overline{u}_{k|k}^{i^{opt}}\right\| < \left\|\overline{u}_{k|k}^{i^*}\right\|$, since $Q \succ 0$ and $R \succ 0$ by hypothesis. This implies that [3] $e_{k+1|k}^{i^{opt}} = e_{k+1|k+1}^{i^{opt}} \neq 0$, 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 $e^i = 0$ will be the perfect control input ($\mathbf{u}^{i^r} = \mathbf{u}^{perf}$). If this is the case, it follows that $V_k^{i^{opt}} = 0$, for $k = 0, ..., T_f$ (Property 0.1), and so, $J^i = 0$

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

$\Leftarrow$) 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 $\mathbf{e}^i = 0$. □

Proof of Theorem 0.4

Proof The idea here is to show that $V_k^{i^{opt}} \leq V_k^{i-1^{opt}}$ for $k = 0, \ldots, T_f - 1$ 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

$$\boldsymbol{\delta}^i = \left[\overline{u}_{0|0}^{i^{opt}T} \ \ldots \ \overline{u}_{T_f-1|T_f-1}^{i^{opt}T} \right] = [0 \ldots 0]^T,$$

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

$$\mathbf{u}^i = \mathbf{u}^{i-1} = \left[u_0^{i-1^T} \ \ldots u_{T_f-1}^{i-1^T} \right]^T = \left[u_{0|0}^{i-1^{opt}T} \ \ldots u_{T_f-1|T_f-1}^{i-1^{opt}T} \right]^T$$

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

$$\tilde{V}_k^i := \sum_{j=0}^{N_s-1} \ell \left(e_{k+j|k+j}^{i-1^{opt}}, \overbrace{\overline{u}_{k+j|k+j}^i}^{=0} \right) + \sum_{j=N_s}^{H-1} \ell \left(e_{k+j|k+j}^{i-1^{opt}}, 0 \right) + F \left(x_{k+H|k+H-1}^{i-1^{opt}} \right)$$

$$= \sum_{j=0}^{H-1} l \left(e_{k+j}^{i-1} \right) + F \left(x_{k+H}^{i-1} \right) \tag{26}$$

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 $u_{T_f-1}^{i-V}$, 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 (12) we have

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

$$\vdots$$

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

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

$$V_{T_f-1}^{i-1^{opt}} + \ell\left(e_{T_f-2}^{i-1}, \overline{u}_{T_f-2}^{i-1}\right) + \cdots + \ell\left(e_0^{i-1}, \overline{u}_0^{i-1}\right) \leq V_0^{i-1^{opt}} \tag{27}$$

From (14), 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,

$$V_{T_f-1}^{i-1^{opt}} = \ell\left(e_{T_f-1}^{i-1}, \overline{u}_{T_f-1}^{i-1}\right) + F\left(x_{T_f}^{i-1}\right). \tag{28}$$

Therefore, by substituting (28) in (27), we have

$$F\left(x_{T_f}^{i-1}\right) + \ell\left(e_{T_f-1}^{i-1}, \overline{u}_{T_f-1}^{i-1}\right) + \cdots + l\left(e_0^{i-1}, \overline{u}_0^{i-1}\right) \leq V_1^{i-1^{opt}} \tag{29}$$

Now, the pseudo cost (26) at time $k=0$, $\tilde{V}_0^i$, can be written as

$$\tilde{V}_0^i = \sum_{j=0}^{T_f-1} l\left(e_j^{i-1}, 0\right) + F\left(x_{T_f}^{i-1}\right)$$

$$= \sum_{j=0}^{T_f-1} l\left(e_j^{i-1}, \overline{u}_j^{i-1}\right) + F\left(x_{T_f}^{i-1}\right) - \sum_{j=0}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\| \tag{30}$$

and from the comparison of the left hand side of inequality (29) with (30), it follows that

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

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

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

Therefore, from the definition of the run cost J i we have[4]

$$\tilde{J}_i \leq J_{i-1} - \sum_{k=0}^{T_f-1} \sum_{j=k}^{T_f-1} \left\|\overline{u}_j^{i-1}\right\|. \tag{31}$$

[4] Notice that, if the run i implements the manipulated variable $u_j^i = u_j^{i-1} + \overline{u}_j^{i-1}$, $j = 0, 1, \ldots, T_f - 1$ and $\overline{u}_j^{i-1} \neq 0$ for some j; then, according to 31 $\tilde{J}_i < J_{i-1}$. 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.

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

$$J_i \leq \tilde{J}^i. \tag{32}$$

From (31) and (32) we have

$$J_i \leq \tilde{J}_i \leq J_{i-1} - \sum_{k=0}^{T_f-1} \sum_{j=k}^{T_f-1} \left\| \overline{u}_j^{i-1} \right\|. \tag{33}$$

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 $\overline{u}_{k+j|k}^{i-1} \neq 0$. As a result, two options arise:

I) Let us assume that $\mathbf{u}^{ir} \neq \mathbf{u}^{perf}$. Then, by property 0.3, $J^i \neq 0$ and following the reasoning used in the proof of Property 0.2, $\overline{u}_j^i \neq 0$, for some $1 \leq j \leq T_f$. Then, according to 33, $J_{i+1} \leq \tilde{J}_{i+1} \leq J_i - \sum_{k=0}^{T_f-1} \sum_{j=k}^{T_f-1} \left\| \overline{u}_j^{i-1} \right\|$ with $\left\| \underline{u}_j^i \right\| > 0$ for some $1 \leq j \leq T_f - 1$.

The sequence J^i will stop decreasing only if $\sum_{j=0}^{T_f-1} \left\| \overline{u}_j^i \right\| = 0$. In addiction, if $\sum_{j=0}^{T_f-1} \left\| \overline{u}_j^i \right\| = 0$, then $\mathbf{u}^{ir} = \mathbf{u}^{perf}$, which implies that $J_i = 0$. Therefore: $\lim_{i\to\infty} J_i = 0$, which, by Property 0.2 implies that $\lim_{i\to\infty} \mathbf{e}_i = 0$.

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

II) Let us assume that $\mathbf{u}^{ir} = \mathbf{u}^{perf}$. Then, by Corollary 0.3, $J_i = 0$, and according to (33), $J_{i+1} = \tilde{J}_{i+1} = J_i = 0$. Consequently, by Property 0.2, $\mathbf{e}^i = 0$. □

8. References

Adam, E. J. (2007). Adaptive iterative learning control applied to nonlinear batch reactor, *XII Reunión de Trabajo en Procesamiento de la Información y Control (RPIC 2007)*, Río Gallegos, Santa Cruz, Argentina, (CD version).

Arimoto, S., Kawamura, S. & Miyazaki, F. (1984). Bettering operation of robots by learning, *Journal of Robotic System* 1(2): 123–140.

Bien, Z. & Xu, J.-X. (1998). *Iterative Learning Control: Analysis, Design, Iteration and Application*, Kluwer Academic Publishers.

Bonvin, D., Srinivasan, B. & Hunkeler, D. (2006). Control and optimization of bath processes - improvement of process operation in the production of specialty chemicals, *IEEE Trans. Control Syst. Mag.* 26(2): 34–45.

Camacho E. F. & Bordons C. (2009). *Model Predictive Control*, 2nd Edition, Springer-Verlag.

Chen, Y. & Wen, C. (1999). *Iterative Learning Control*, Springer Verlag.

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

Doh, T. Y., J. H. moon, K. B. J. & Chung, M. J. (1999). Robust ilc with current feedback for uncertain linear system, *Int. J. Syst. Sci* 30(1): 39–47.

González, H. A., Adam, E. J. & Marchetti, J. L. (2008). Conditions for offset elimination in state space receding horizon controllers: A tutorial analysis, *Chemical Engineering and Processing* 47: 2184–2194.

González, H. A., Adam, E. J., Odloak, D. O. & Marchetti, J. L. (2009). Infinite horizon mpc applied to batch processes. Part I, *XII Reunión de Trabajo en Procesamiento de la Información y Control (RPIC 2009)*, Rosario, Santa Fe, Argentina, (CD version).

González, H. A., Adam, E. J., Odloak, D. O. & Marchetti, J. L. (2009). Infinite horizon mpc applied to batch processes. Part II, *XII Reunión de Trabajo en Procesamiento de la Información y Control (RPIC 2009)*, Rosario, Santa Fe, Argentina, (CD version).

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

Horowitz, R. (1993). Learning control of robot manipulators, *Journal of Dynamic Systems, Measurement and Control* 115: 402–411.

Kurek, J. E. & Zaremba, M. B. (1993). Iterative learning control synthesis based on 2-d system theory, *IEEE Trans. Automat. Contr.* 38(1): 121–125.

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

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

Lee, K. S. & Lee, J. H. (1999). Model predictive control technique combined with iterative learning for batch processes, *AIChE Journal* 45: 2175–2187.

Lee, K. S. & Lee, J. H. (2003). Iterative learning control-based batch process control technique for integrated control of end product properties and transient profiles of process variables, *Computer chemical Engng* 24: 873–879.

Moon, J. H., Doh, T. Y. & Chung, M. J. (1998). A robust approach to iterative learning control design for uncertain system, *Automatica* 34(8): 1001–1004.

Moyne, J. E., Castillo E. & Hurwitz A.M. (2001). *Run to run control in semiconductor manufacturing*, CRC Press.

Norrlöf, M. (2000). *Iterative Learning Control. Analysis, design, and experiments*, Ph. D. Thesis, LinkÃűpings Universtet, Sweden.

Rawlings J. B. and Mayne D. Q. (2009). *Model Predictive Control: Theory and Design*, Nob Hill Publishing.

Roover, D. D. (1996). Synthesis of a robust iterative learning control using an h approach, *Proc. 35th Conf. Decision Control*, Kobe, Japan, pp. 3044–3049.

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

Scholten, P. (2000). *Iterative Learning Control: Analysis. A design for a linear motor motion system*, M. Sc. Thesis University of Twente.

Scokaert, P. O. M., Rawlings, J. B. & Meadows, E. S. (1997). Discrete-time stability with perturbations: application to moel predictive control.

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

Tan, K.K., Huang S.N., Lee T.H. & Tay, A. (2007). Disturbance compensation incorporated in predictive control system using a repetitive learning approach, *System and Control Letters* 56: 75–82.

Tayebi, A. & Zaremba, M. B. (2003). Robust iterative learning control design is straightforward for uncertain lti system satisfying the robust performance condition, *IEEE Trans. Automat. Contr.* 48(1): 101–106.

Yao, K., Gao F., & Allgöwer, F. (2007). Barrel temperature control during operation transition in injection molding, *Control Engineering Practice* 16(11): 1259-1264.

IntechOpen

IntechOpen