Preprints of the
9th International Symposium on Advanced Control of Chemical Processes
The International Federation of Automatic Control
June 7-10, 2015, Whistler, British Columbia, Canada

A stable two-time dimensional (2D) Model Predictive Control with zero terminal state constraints for constrained batch processes

Jingyi LU, **Zhixing CAO**, **Furong GAO**

* *Department of Chemical and Bio-molecular Engineering, Hong Kong University of Science and Technology, Clear water bay, Hong Kong (e-mail: kefgao@ust.hk)*

Abstract: Batch processes are of great importance in process industry. However, the control algorithm design is difficult for those with constraints. This is because stability and recursive feasibility along directions of time and batch should be guaranteed simultaneously. In this paper, a stable model predictive control strategy with zero terminal state constraints is proposed. Stability and recursive feasibility along two directions are guaranteed and proved. Simulation results are given to show the effectiveness of the algorithm.

Keywords: stable Model Predictive Control, zero terminal state constraints, iterative learning control (ILC), recursive feasibility, linear constrained systems.

1. INTRODUCTION

Batch process plays an important role in chemical industries, such as injection molding, pharmaceutical industry and so on. It is often used to produce products with high value. An important characteristic of batch processes is repetitiveness. Based on repetitiveness, information of previous batch can be collected to improve current batch's performance. Iterative Learning Control (ILC)(Arimoto et al. (1984), Uchiyama (1978)) is a good alternative to exploit repetitiveness. The time and batch indexes are denoted as t and k respectively. Then a general iterative learning control law can be given as

$$u(t,k) = u(t,k-1) + Ke(t+1,k-1). \qquad (1)$$

It uses the tracking error and input signal from the previous batch to correct the input signal of current batch. In nature, it is a batch-wise feedback control. Currently, a lot of papers have discussed the convergence properties of ILC, referring to Amann et al. (1996), Norrlöf and Gunnarsson (2002) and Bristow et al. (2006). However, most of them only discuss about batch-wise convergence. From the view of time direction, ILC is a feedforward control. That is why conclusions about time-wise stability are hard to be obtained based on the simple ILC law in Equ. (1).

In order to guarantee time-wise stability, feedback control along the direction of time is combined with ILC. In general, a feedback controller incorporated with ILC is taken as a two-time dimensional controller (time and batch dimensions). In this way, rich results in two dimensional control theories, such as those shown in Kaczorek (1985) and Du et al. (2001), can be applied. Shi et al. (2005a) and Shi et al. (2005b) introduced a 2D Lyapunov function and used this to induce a control law based on LMIs to guarantee 2D stability. The above mentioned idea has been widely applied in process industries. In Wang et al. (2007), the idea was applied to a three-tank system with sensor faults. Shi et al. (2006) applied this idea to the nozzle packing pressure control in a injection molding process. In addition to this idea, Lee and Lee (2003) combined model predictive control with ILC to control a semi-batch reactor. However, these methods do not address issues on constrained systems.

From the perspective of practical requirements in process industries, constraints on either states, inputs or outputs are unavoidable. Without explicitly considering constraints, both control performance and system stability will deteriorate. Lee et al. (1999) and Wang et al. (2008) directly incorporated the constraints into on-line optimization, but stability was not proved. System stability depends on tuning of parameters. In Liu and Wang (2012), a 2D Lyapunov Function was used to guarantee 2D stability. This 2D Lyapunov Function, together with input and state constraints, was formulated as constraints in online optimization. One important issue is that they did not consider feasibility of the optimization. Without considering feasibility, at some time spot, constraints may be so stringent that the feasible solution set is empty. When an optimization-based control strategy occurs in infeasibility, performance will deteriorate rapidly, and stability can not be guaranteed either. Thus, the key concern is how to design a control strategy to guarantee 2D stability, constraint fulfillments and 2D recursive feasibility simultaneously.

Lee and Lee (2000) is an important work to guarantee 2D stability and recursive feasibility for batch processes. They combined MPC with ILC based on a FIR model. The results of this work are based on the following assumptions: 1, the process is stable. A FIR model can thereby be used; 2, the length of a batch is not long, since for each step of optimization, they have to predict the tracking error for the whole batch; 3, The reference is admissible, since the convergence of the algorithm is based on the existence of a feasible solution to make the objective function 0.

In this paper, we aim to propose a control strategy which can guarantee 2D stability and 2D feasibility at the same time with fewer assumptions compared with Lee and Lee (2000). The foundation of this work is a zero terminal state constraint. This type of constraint is common and usefull in MPC for continuous processes. In MPC, there are four ways to guarantee stability and feasibility, namely infinite prediction horizon (Keerthi and Gilbert (1988), Kothare et al. (1996)), zero terminal state constraints (Kwon and Pearson (1977), Rawlings and Muske (1993)), terminal weighting matrix (Kwon et al. (1983), Kwon and Byun (1989)), terminal set (Mayne et al. (2000), Wan and Kothare (2003)). For our problem, since not only the time-wise feasibility is considered, but also the batch-wise feasibility, the zero terminal state constraint method is adopted.

This paper is organized as follows. In Section 2, the formulation of the problem is given firstly, and then the control strategy design is described in details. In Section 3, properties on feasibility, stability and optimality are analyzed. In Section 4, simulations are conducted to show the effectiveness of the method. Finally, conclusions are made.

2. CONTROL STRATEGY DESIGN

2.1 Problem Formulation

A batch process can be described in state space as

$$\begin{aligned} x(t+1,k) =& Ax(t,k) + Bu(t,k) + d(t), \\ y(t,k) =& Cx(t,k), \quad t \in [0, t_n], \quad k \in [1, \infty). \end{aligned} \tag{2}$$

Constraints are

$$-\underline{u}_m \preceq u(t,k) \preceq \overline{u}_m \tag{3}$$

$$-\underline{X}_m \preceq x(t,k) \preceq \overline{X}_m \tag{4}$$

Here t is the index of time and k is the index of batch. t_n is the length of a batch. $x \in \mathbb{R}^{n_x}$ is system states. The initial states $x(0,k)$ for each batch are the same by assuming that the process resets. $y \in \mathbb{R}^{n_y}$ is system outputs. $u \in \mathbb{R}^{n_u}$ is system inputs. The reference to be tracked is denoted as $y_r(t) \in \mathbb{R}^{n_y}$ and it is identical for each batch. $d(t) \in \mathbb{R}^{n_x}$ denotes exogenous disturbances and it is repetitive along the direction of batch. The matrix A, B and C are arranged in proper dimensions according to x, y and u. The bounds $\overline{u}_m$, $\underline{u}_m$, $\overline{X}_m$ and $\underline{X}_m$ are all positive point-wisely. '$\preceq$' denotes '$\leq$' in terms of each coordinate of vector x. For simplicity, it is assumed that states and outputs are measurable.

In order to eliminate repetitive disturbances and incorporate ILC, difference of two consecutive batches is taken as

$$\begin{aligned} \Delta_k x(t+1,k) =& A\Delta_k x(t,k) + B\Delta_k u(t,k) \\ \Delta_k y(t,k) =& C\Delta_k x(t,k), \end{aligned} \tag{5}$$

with $\Delta_k x(0,k) = 0$ for each batch. The constraints in (3) and (4) become

$$-\underline{u}_m - u(t,k-1) \preceq \Delta_k u(t,k) \preceq \overline{u}_m - u(t,k-1), \tag{6}$$

$$-\underline{X}_m - x(t,k-1) \preceq \Delta_k x(t,k) \preceq \overline{X}_m - x(t,k-1). \tag{7}$$

In addition, in batch process control, constraints on $\Delta_k u$ and $\Delta_k x$ may also be added to control convergent rate as follows.

$$-\delta_k \underline{u}_m \leq \Delta_k u(t,k) \leq \delta_k \overline{u}_m \tag{8}$$

$$-\delta_k \underline{X}_m \preceq \Delta_k x(t,k) \preceq \delta_k \overline{X}_m \tag{9}$$

Feedback control strategies incorporated with ILC are in general designed based on system in (5) and directly taking $\Delta_k u$ as the input variables. Then, the constraints in (6) and (7) are not uniform since $u(t,k-1)$ and $x(t,k-1)$ can be any value within the bounds in (3) and (4). This poses great difficulties for controller design when batch-wise recursive feasibility is required.

Next, it will be shown that how to design a MPC with zero terminal state constraints based on the system in (5) to guarantee 2D stability and 2D feasibility simultaneously.

2.2 Control Algorithm

For simplicity, we denote

$$e(t,k) = y_r(t,k) - y(t,k).$$

System in (5) is equivalent to

$$\begin{aligned} \Delta_k x(t+1,k) =& A\Delta_k x(t,k) + B\Delta_k u(t,k) \\ e(t,k) =& -C\Delta_k x(t,k) + e(t,k-1). \end{aligned} \tag{10}$$

The constraints in (6), (7), (8) and (9) on $\Delta_k u(t,k)$ and $\Delta_k x(t+1,k)$ for the kth batch are denoted as $\Omega(t,k)$. Thus, constraints in (6), (7), (8) and (9) are simplified as

$$\Delta_k x(t+1,k), \Delta_k u(t,k) \in \Omega(t,k).$$

Assume that prediction horizon and control horizon are both p_n. Then, a prediction model can be derived as

$$\Delta_k x_p(t+i+1,k) = A\Delta_k x_p(t+i,k) + B\Delta_k u_p(t+i,k) \tag{11}$$

$$\begin{aligned} e_p(t+i+1,k) =& -C\Delta_k x_p(t+i+1,k) + \\ & e(t+i+1,k-1). \end{aligned} \tag{12}$$

with $i = 0, 1, \ldots, p_n - 1$.

For $k \geq 2$, the control law can be induced by an optimization as

$$\min_{\Delta_k x_p(^{t+1}_{t+p_n},k), \Delta_k u_p(^{t}_{t+p_n-1},k), e_p(^{t+1,k}_{t+p_n},k)} e(^{t+1}_{t+p_n},k)^T e(^{t+1}_{t+p_n},k) \tag{13}$$

subject to constraints

$$(11)(12)$$

$$\Delta_k x_p(t+i+1,k), \Delta_k u_p(t+i,k) \in \Omega(t+i,k), \tag{14}$$

$$\Delta_k x_p(t+p_n,k) = 0, \quad i = 0, 1, \ldots, p_n - 1. \tag{15}$$

With such an optimization, $\Delta_k u(^{t}_{t+p_n-1},k)$ are calculated. According to the philosophy of receding horizon strategy, only the first step is implemented and the input can be computed as

$$u(t,k) = u(t,k-1) + \Delta_k u(t,k). \tag{16}$$

Since a batch process has a finite length of t_n, when $t > t_n - p_n$, a shrinking prediction horizon strategy is adopted. That means when $t = t_n - p_n + i$, with $i = 0, 1, \ldots, p_n$, the prediction horizon becomes $p_n - i$.

Remark 1: As claimed above, the algorithm is implemented from the second batch, excluding the first batch. Some other methods guaranteeing stability and constraint fulfillments can be applied to the control of the first batch, such as that shown in Hu et al. (2002).

Remark 2: It is noted that the optimization in (13) is a quadratic programming with linear constraints. Computation complexity is of $\mathcal{O}(n^2)$ by interior point mehtod.

3. FEASIBILITY, STABILITY AND OPTIMALITY

In this section, feasibility, stability and optimality of the algorithm will be analyzed by the following theorems.

Theorem 1. (**Time-wise recursive feasibility**): If there is a feasible solution for the problem in (13) at (t, k), then there exists a feasible solution for (13) at $(t + 1, k)$.

Proof. Since the optimization in (13) is feasible at (t, k), denote the feasible solution as $\Delta_k \hat{u}_p(t, k), \Delta_k \hat{u}_p(t + 1, k), \ldots, \Delta_k \hat{u}_p(t+p_n-1, k)$, corresponding states $\Delta_k \hat{x}_p(t+1, k), \ldots, \Delta_k \hat{x}_p(t + p_n, k)$, and tracking errors $e_p(t + 1, k), \ldots, e_p(t + p_n, k)$. Since the terminal state $\Delta_k \hat{x}(t + p_n, k) = 0$, according to (15), we can prove that $\Delta_k \hat{u}_p(t + 1, k), \Delta_k \hat{u}_p(t+2, k), \ldots, \Delta_k \hat{u}_p(t+p_n-1, k), 0$ and the corresponding states and tracking errors are feasible solutions at $(t + 1, k)$. Fulfillments of (11), (12) and (14) are trivial. For (15), since $\Delta_k \hat{x}(t+p_n, k) = 0$, and $\Delta_k u(t+p_n, k) = 0$, according to Equ. (11), $\Delta_k \hat{x}(t + p_n + 1, k) = 0$. Thus, the constraint (15) is satisfied. The problem is feasible at $(t + 1, k)$.□

Theorem 2. (**Batch-wise recursive feasibility**): If the problem in (13) is feasible at $(0, k)$, then it is feasible at $(0, k + 1)$, $\forall k \in [1, \infty)$.

Proof. Due to the assumption $\Delta_k x(0, k) = 0, \forall k \in [2, \infty)$, take $\Delta_k \hat{u}_p(0, k) = \Delta_k \hat{u}_p(1, k) = \cdots = \Delta_k \hat{u}_p(p_n-1, k) = 0$. Then the inputs and induced states are feasible to all constraints.□

Lemma 3. (**2D recursive feasibility**): If the process can be stabilized by a control law that makes all constraints fulfilled in the first batch, the optimization problem in (13) is feasible for $\forall(t, k)$ with $t \in [0, t_n]$, $k \in [2, \infty)$.

Proof. Firstly, according to Theorem 2, (13) is feasible at $(0, 2)$. Furthermore, by Theorem 1, (13) is feasible at $\forall t \in [0, t_n]$ in the 2nd batch. By induction, (13) is feasible at $\forall(t, k)$ that $t \in [0, t_n]$, $k \in [2, \infty)$.□

Based on feasibility, we can further discuss stability and optimality. Before that, a performance index at (t, k) is defined as

$$\Phi(t, k) = \sum_{i=1}^{t} e(i, k)^2 + \sum_{i=t+1}^{t+p_n} e_p(i, k)^2 + \sum_{i=t+p_n+1}^{t_n} \bar{e}(i, k)^2 \tag{17}$$

At (t, k), the tracking error $e(1, k)$ to $e(t, k)$ can be directly measured. $e_p(t+1, k)$ to $e_p(t+p_n, k)$ can be calculated by implementing $\Delta_k u(t, k)$ to $\Delta_k u(t + p_n - 1, k)$ computed by (13). $\bar{e}(t + p_n + 1, k)$ to $\bar{e}(t_n, k)$ is the tracking error by applying $\Delta_k u(t + p_n, k) = \cdots = \Delta_k u(t_n, k) = 0$. Since $\Delta_k x_p(t + p_n, k) = 0$, $\bar{e}(t + p_n + i, k) = e(t + p_n + i, k - 1)$. Thus, this index can be calculated after the optimization in (13) is conducted at time t of the kth batch. Based on this $\Phi(t, k)$, stability of the method can be proved.

Theorem 4. (**Time-wise stability**:) For $t \in [0, t_n]$, $k \in [2, \infty)$, by deriving control law from the optimization in (13), it is guaranteed that $\Phi(t + 1, k) \leq \Phi(t, k)$.

Proof. Assume the optimal inputs at (t, k) obtained from (13) are $\Delta_k \hat{u}_p(t, k), \Delta_k \hat{u}_p(t+1, k), \ldots, \Delta_k \hat{u}_p(t+p_n-1, k)$. With these inputs, the tracking errors from $t+1$ to $t+p_n$ are $e_p(t+1, k), \ldots, e_p(t+p_n, k)$. According to Theorem 1, at $(t+1, k)$, $\hat{u}_p(t+1, k), \ldots, \hat{u}_p(t+p_n-1, k), 0$ is a group of feasible inputs. If these inputs are implemented, the tracking errors will be $e_p(t + 2, k), \ldots, e_p(t + p_n, k), \bar{e}(t + p_n+1, k)$. Assume the performance index corresponding to $\Delta_k \hat{u}_p(t+1, k), \ldots, \Delta_k \hat{u}_p(t+p_n-1, k), 0$ is $\hat{\Phi}(t+1, k)$. Then $\hat{\Phi}(t + 1, k) = \Phi(t, k)$. By the optimization (13), it can be guaranteed that the objective function corresponding to the optimal inputs $\sum_{i=t+2}^{t+p_n+1} \tilde{e}_p(i, k)^2 \leq \sum_{i=t+2}^{t+p_n} e_p(i, k)^2 + \bar{e}(t + p_n + 1)$. This implies that the optimal $\Phi(t + 1, k)$ satisfies

$$\Phi(t + 1, k) \leq \hat{\Phi}(t + 1, k) = \Phi(t, k).$$

□

Theorem 5. (**Batch-wise stability**): For $\forall k \in [0, t_n]$, $k \in [2, \infty)$, $\Phi(t, k + 1) \leq \Phi(t, k)$.

Proof. Similar to the proof of Theorem 4, according to Theorem 2, $\Delta_k \hat{u}_p(0, k) = \Delta_k \hat{u}_p(1, k) = \cdots = \Delta_k \hat{u}_p(p_n - 1, k) = 0$ is a group of feasible solution at $(0, k + 1)$. By implementing these inputs, $\hat{\Phi}(0, k + 1) = \Phi(t_n, k)$. According to the optimality of the optimal solution,, we can guarantee that

$$\Phi(0, k + 1) \leq \hat{\Phi}(0, k + 1) = \Phi(t_n, k).$$

By applying Theorem 4 t times, we have $\Phi(t, k + 1) \leq \Phi(0, k + 1)$. Similarly, $\Phi(t_n, k) \leq \Phi(t, k)$. Thus,

$$\Phi(t, k + 1) \leq \Phi(t, k).$$

□

Stability is easily guaranteed, but all the above inequalities are $'$ less than or equal to $'$, not necessarily strictly $'$ less than $'$. How to guarantee optimality is an important issue. Next, optimality of the algorithm is discussed.

Theorem 6. Define performance index Φ corresponding to a MPC with prediction horizon p_n as $\Phi_{p_n}(t, k)$, then $\Phi_{p_n}(t, k) \leq \Phi_{p_n+1}(t, k)$.

Proof. Assume at (t, k), when prediction horizon is p_n, the optimal inputs are $\Delta_k \hat{u}_p(t, k), \Delta_k \hat{u}_p(t + 1, k), \ldots, \Delta_k \hat{u}_p(t + p_n - 1, k)$, and the corresponding performance index is $\Phi_{p_n}(t, k)$. If the prediction horizon is increased to $p_n + 1$, it is easy to see $\Delta_k \hat{u}_p(t, k), \Delta_k \hat{u}_p(t + 1, k), \ldots, \Delta_k \hat{u}_p(t+p_n-1, k), 0$ is a group of feasible inputs. Assume the corresponding performance index for this group of inputs is $\bar{\Phi}_{p_n+1}(t, k)$. Furthermore, by the optimization in (13), it is possible that there exists another group of feasible inputs which can make the performance index smaller. Thus,

$$\Phi_{p_n+1}(t, k) \leq \bar{\Phi}_{p_n+1}(t, k) = \Phi_{p_n}(t, k).$$

□

Theorem 6 implies that larger prediction horizon induces better performance. The extreme case is to take $p_n = t_n$, which is the largest prediction horizon can be chosen. Next, with only constraints in (6) and (7) included in (13), we can prove the optimal solutions can be obtained.

Theorem 7. **(Optimality)**: Taking $p_n = t_n$, the inputs induced by (13) are the optimal solutions with (6) and (7) imposed.

Proof. By taking $p_n = t_n$, the objective function in (13) is the same as the performance index Φ. The optimal inputs, with the corresponding states and prediction errors are a group of feasible solutions to (13). Since (13) is convex, it is sure that optimality can be obtained.□

Remark 3: When the reference is admissible, the optimal solution $u(0,k),\dots,u(t_n,k)$ can make $\Phi_{t_n} = 0$. As $\Phi \geq 0$, $u(0,k),\dots,u(t_n,k)$ is the optimal solution to the problem (13). Thus, perfect tracking is obtained.

Remark 4: In general, for the sake of robustness against exogenous disturbances and model uncertainty, the objective function used in a MPC is of the form

$$\min_{\Delta_k x_p(^{t+1}_{t+p_n},k),\Delta_k u_p(^{t}_{t+p_n-1},k),e_p(^{t+1,k}_{t+p_n},k)} e(^{t+1}_{t+p_n},k)^T R e(^{t+1}_{t+p_n},k)$$
$$+ \Delta_k u(^{t}_{t+p_n-1},k)^T Q \Delta_k u(^{t}_{t+p_n-1},k) \qquad (18)$$

instead of that in (13).

Remark 5: When constraints in (8) and (9) are included in (13), or objective function (18) is adopted, conclusions in Section 3 still hold except for Theorem 7. However, Theorem 7 holds when $k \to \infty$. That means neither (8,9) nor (18) affects optimality of the method. They only determine the rate the method converges to the optimal solution along the batch direction. Due to limitation of space, the proof is omitted.

According to Theorem 7, by taking $p_n = t_n$, the optimal solution can be obtained. However, due to the significant computation burden, this is usually not realistic. Theorem 6 shows that there is a tradeoff between the length of prediction horizon and optimality. The larger prediction horizon is, the better the performance is. Thus, prediction horizon is regarded as a tuning parameter. One can use it to balance computation burdens and optimality of the performance.

4. SIMULATIONS

- Case 1: stable system

In this part, control of injection velocity in injection molding process is taken as an example to test the method proposed. Injection velocity in the filling phase is a key variable for product quality. Thereby, it is very important to make it follow a pre-designed optimal profile tightly. According to Wang et al. (2008), the dynamic model of injection velocity versus valve opening in state space is as

$$x(t+1,k) = \begin{bmatrix} 1.582 & -0.5916 \\ 1 & 0 \end{bmatrix} x(t,k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t,k) + d(t)$$
$$y(t,k) = [\,1.69 \;\; 1.419\,]\, x(t,k), \quad t \in [1,50]. \qquad (19)$$

$d(t)$ is taken as $\begin{bmatrix} 0.01sin(6t) \\ 0.01sin(5t) \end{bmatrix}$. It is repetitive for each batch. The eigenvalues of A are 0.9756 and 0.6064. The system is stable. Fig 1 shows the reference. The constraints are

$$-2 \leq u(t,k) \leq 2 \qquad (20)$$
$$-0.3 \leq \Delta_k u(t,k) \leq 0.3 \qquad (21)$$
$$-10 \leq x^i(t,k) \leq 10 \quad i = 1,2 \qquad (22)$$

For the first batch, firstly the dynamic model is integrated as

$$\begin{bmatrix} x(t+1,k) \\ \sum_{i=1}^{t} e(i,k) \end{bmatrix} = \begin{bmatrix} A & \mathbf{0} \\ -C & I \end{bmatrix} \begin{bmatrix} x(t,k) \\ \sum_{i=1}^{t-1} e(i,k) \end{bmatrix} + \begin{bmatrix} B \\ \mathbf{0} \end{bmatrix} u(t,k) + \begin{bmatrix} d(t) \\ y_r(t) \end{bmatrix}.$$

Based on this model, the method shown in Hu et al. (2002) can be applied to induce a feedback control law as

$$K = [-0.8088 \;\; 0.2016 \;\; 0.0506].$$

A performance index E_2 is defined as

$$E_2(k) = \sqrt{(\sum_{t=1}^{t_n} e^2(t,k))}.$$

Fig. 2 shows the outputs of the method. It is easy to see that the outputs gradually converge to the reference. This can also be seen from Fig. 3. After 10 batches, the tracking error for each batch converges to 0. Furthermore, from Fig. 4 and Fig. 5, we can see neither inputs nor states violate constraints.

In order to show the length of prediction horizon impacts optimality, simulation is conducted for prediction horizon taken as $2, 3, 5, 10, 25$. E_2 for each case is shown in Fig. 6. We can see when $p_n = 2$, constraints are too tight and feasible region is too small. There is no batch-wise learning. Thus, tracking errors are kept to be the same. With the increase of prediction horizon, the errors converge to 0 at a faster speed. When $p_n \geq 5$, the performance keeps to be the same and no longer gets better. This also shows, to choose a proper prediction horizon, not necessarily extremely large, the performance can be close to the optimal one.

- Case 2: an unstable system

In order to show the method is applicable to unstable systems, we assume system in (19) is disturbed and the dynamic model becomes

$$x(t+1,k) = \begin{bmatrix} 1.682 & -0.5916 \\ 1 & 0 \end{bmatrix} x(t,k) + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u(t,k) + d(t)$$
$$y(t,k) = [\,1.69 \;\; 1.419\,]\, x(t,k), \quad t \in [1,50].$$

Now the eigenvalues of A are 1.1811 and 0.5009. The system is unstable. The feedback control law for the first batch is taken as

$$K = [-0.8168 \;\; 0.2182 \;\; 0.041].$$

Prediction horizon is taken as $p_n = 10$. Constraints are the same as (20), (21) and (22). Fig. 7 and Fig. 8 show the tracking performance. From these figures, we can conclude that although the system is unstable, the method can still stabilize the system and make the tracking errors converge to 0 finally.

5. CONCLUSION

In this work, a two-dimensional model predictive control strategy with zero terminal state constraints is proposed. Feasibility, stability and optimality of the method are analyzed. Simulation results show that the method is applicable to both stable and unstable systems. In future, the method will be improved and extended to batch systems with non-repetitive disturbances.

ACKNOWLEDGEMENTS

This work is supported by Hong Kong Research Grant Council under project no. 612512.

REFERENCES

Amann, N., Owens, D.H., and Rogers, E. (1996). Iterative learning control for discrete-time systems with exponential rate of convergence. *IEE Proceedings-Control Theory and Applications*, 143(2), 217–224.

Arimoto, S., Kawamura, S., and Miyazaki, F. (1984). Bettering operation of dynamic systems by learning: A new control theory for servomechanism or mechatronics systems. *IEEE Proc on Decision and Control.*, 23, 1064–1069.

Bristow, D.A., Tharayil, M., and Alleyne, A.G. (2006). A survey of iterative learning control. *Control Systems, IEEE*, 26(3), 96–114.

Du, C., Xie, L., and Zhang, C. (2001). H_∞ control and robust stabilization of two-dimensional systems in roesser models. *Automatica*, 37(2), 205–211.

Hu, T., Lin, Z., and Chen, B.M. (2002). Analysis and design for discrete-time linear systems subject to actuator saturation. *Systems & Control Letters*, 45(2), 97–112.

Kaczorek, T. (1985). *Two-dimensional linear systems.* Springer Berlin.

Keerthi, S.a. and Gilbert, E.G. (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, 57(2), 265–293.

Kothare, M.V., Balakrishnan, V., and Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32(10), 1361–1379.

Kwon, W.H., Bruckstein, A., and Kailath, T. (1983). Stabilizing state-feedback design via the moving horizon method. *International Journal of Control*, 37(3), 631–643.

Kwon, W. and Pearson, A. (1977). A modified quadratic cost problem and feedback stabilization of a linear system. *Automatic Control, IEEE Transactions on*, 22(5), 838–842.

Kwon, W.H. and Byun, D.G. (1989). Receding horizon tracking control as a predictive control and its stability properties. *International Journal of Control*, 50(5), 1807–1824.

Lee, K. and Lee, J. (2000). Convergence of constrained model-based predictive control for batch processes. *Automatic Control, IEEE Transactions on*, 45(10), 1928–1932.

Lee, K.S., Chin, I.S., Lee, H.J., and Lee, J.H. (1999). Model predictive control technique combined with iterative learning for batch processes. *AIChE Journal*, 45(10), 2175–2187.

Lee, K.S. and 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. *Journal of Process Control*, 13(7), 607–621.

Liu, T. and Wang, Y. (2012). A synthetic approach for robust constrained iterative learning control of piecewise affine batch processes. *Automatica*, 48(11), 2762–2775.

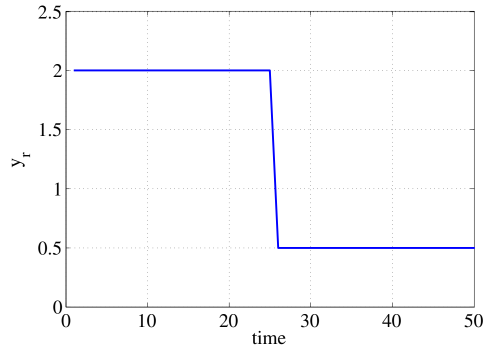

Fig. 1. Reference

Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.

Norrlöf, M. and Gunnarsson, S. (2002). Time and frequency domain convergence properties in iterative learning control. *International Journal of Control*, 75(14), 1114–1126.

Rawlings, J.B. and Muske, K.R. (1993). The stability of constrained receding horizon control. *Automatic Control, IEEE Transactions on*, 38(10), 1512–1516.

Shi, J., Gao, F., and Wu, T.J. (2005a). Integrated design and structure analysis of robust iterative learning control system based on a two-dimensional model. *Industrial & engineering chemistry research*, 44(21), 8095–8105.

Shi, J., Gao, F., and Wu, T.J. (2005b). Robust design of integrated feedback and iterative learning control of a batch process based on a 2D roesser system. *Journal of Process Control*, 15(8), 907–924.

Shi, J., Gao, F., and Wu, T.J. (2006). Robust iterative learning control design for batch processes with uncertain perturbations and initialization. *AIChE journal*, 52(6), 2171–2187.

Uchiyama, M. (1978). Formulation of high-speed motion pattern of a mechanical arm by trial (in japaness). *Trans. SICE(Soc. Instrum. Control Eng.)*, 14, 706–702.

Wan, Z. and Kothare, M.V. (2003). Efficient robust constrained model predictive control with a time varying terminal constraint set. *Systems & control letters*, 48(5), 375–383.

Wang, Y., Yang, Y., Zhou, D., and Gao, F. (2007). Active fault-tolerant control of nonlinear batch processes with sensor faults. *Industrial & Engineering Chemistry Research*, 46(26), 9158–9169.

Wang, Y., Zhou, D., and Gao, F. (2008). Iterative learning model predictive control for multi-phase batch processes. *Journal of Process Control*, 18(6), 543–557.

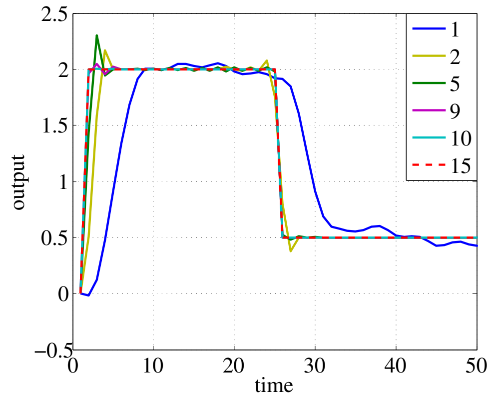

Fig. 2. Case 1: outputs for batch 1,2,5,9,10 and 15.

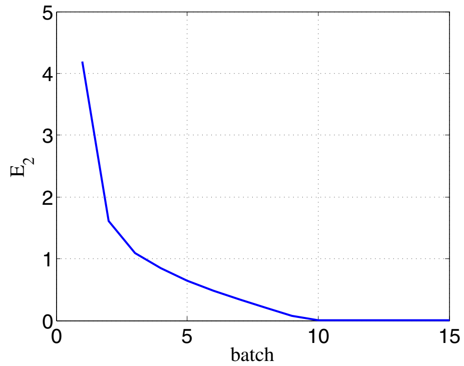

Fig. 3. Case 1: the performance index E_2

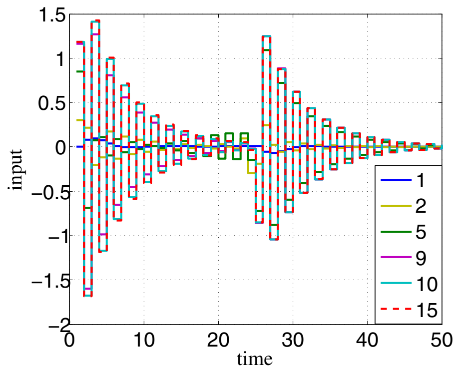

Fig. 4. Case 1: inputs for batch 1,2,5,9,10 and 15

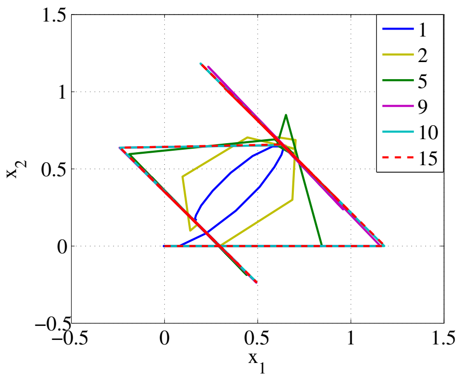

Fig. 5. Case 1: states for batch 1,2,5,9,10 and 15

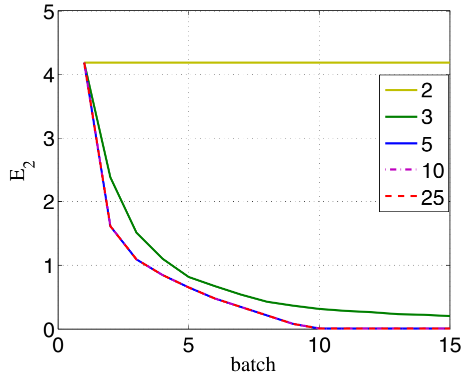

Fig. 6. Case 1: E_2 for prediction horizon taken as 2, 3, 5, 10, 25

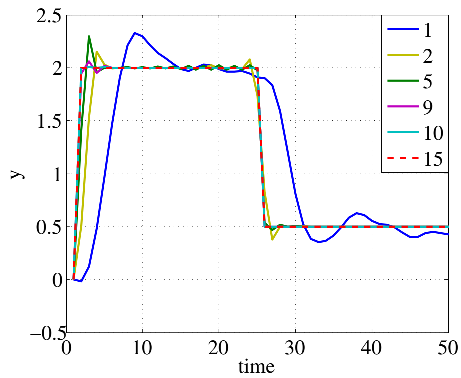

Fig. 7. Case 2: outputs for batch 1,2,5,9,10 and 15.

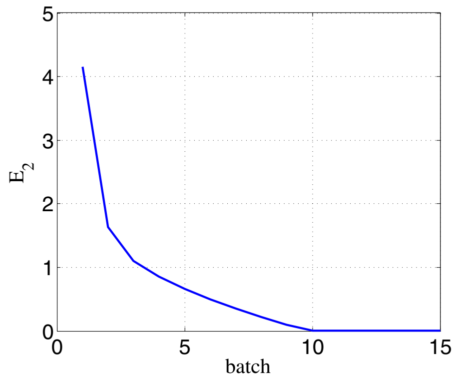

Fig. 8. Case 2: the performance index E_2