Two-time dimensional dynamic matrix control for batch processes with convergence analysis against the 2D interval uncertainty

Shengyong Mo, Limin Wang, Yuan Yao, Furong Gao

Abstract

A batch process can be treated as a 2-dimensional (2D) system with a time dimension within each batch and a batch dimension from batch to batch. This paper integrates the learning ability of iterative learning control (ILC) into the prediction model of model predictive control (MPC). Based on this integrated model, a 2D dynamic matrix control (2D-DMC) algorithm with a feedback control and an optimal feed-forward control is proposed. The sufficient conditions for exponentially asymptotic and monotonic convergence of the proposed 2D-DMC are established with proof under certain assumptions, in the presence of not only the completely repeatable uncertainties but also the non-repeatable interval uncertainties. The effectiveness of the proposed control scheme is tested through simulation and experimental implementation in the context of injection molding, a typical batch process. The results show that the batch process control performance is significantly improved.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Since the first ever paper on model predictive control (MPC) [1] was published, it triggered great interests and has been gaining a wide acceptance in process control community [2,3]. Nowadays, MPC is the most popular advanced control strategy and has been successfully applied in process industries including refinery, petrochemicals and chemicals, mostly in continuous processes. The range of application of MPC continues to expand to this day. Batch production or batch processing is the preferred method for manufacturing high-value-added products such as specialty chemicals and consumer products [4]. In order to improve product quality and quality consistency in batch processes, it is critical to enhance the tracking performance of key process variables. MPC has the potential to achieve this. However, most applications of MPC [5,6] utilize linear prediction models to predict the future responses of the manufacturing process. Due to the low production volume and the significant nonlinearity, the batch process normally needs to work over a wide range of operating conditions, making the use of linear prediction models in batch process control unsuitable. Researchers have investigated the extension of MPC to batch process control by incorporating nonlinear prediction models [7,8]. However, finding a reliable model is time consuming and demands huge efforts in most cases. Moreover, in batch processes, the product specifications may change frequently in response to the dynamic market demand [9], which results in the frequent adjustment of the corresponding nonlinear prediction model. Consequently, as a model-based control strategy initially designed for continuous processes, the conventional MPC is ill-suited for batch processes as the natures of these two types of processes are quite different.

Different from the continuous process, a batch process has the following distinct features: (a) a batch process has finite operation duration; and (b) repeats itself until the specified amount of products has been made. To exploit the repetitive nature of batch processes for control performance improvement, iterative learning control (ILC) [10–13] – originally introduced to control the repetitive motions of robots and systematically studied in the past three decades – is the most suitable strategy. The conventional ILC is essentially a feed-forward control strategy that rejects repeated uncertainties but is useless against non-repetitive uncertainties. Due to environmental factors and changes in operating conditions, non-repeatability is inevitable and auto correlated in batch processes. To guarantee the control performance in the presence of both repeatable and non-repeatable disturbances, researchers [14,15] have combined ILC with feedback control strategies. ILC is adopted for batch-to-batch learning, while feedback information is utilized to handle within-batch uncertainty. This combination of two separately designed controllers triggers two questions: Is the combination of two separately optimized controllers still
optimal? How do we guarantee the convergence and robustness of the combined controllers? A two-dimensional (2D) framework has been proposed [17–19], which attempted to answer these two questions by considering the control actions/dynamics within a batch as one dimension (the time dimension) and the control actions/dynamics from batch to batch as another dimension (the batch dimension). Thus batch processes can be mathematically described using 2D models such as the Roesser type model [20,21] or the Fornasini–Marchesini type model [22,23]. This allows the natural integration of feedback control and ILC and guarantees the convergence and stability along both dimensions.

Motivated by the above considerations, a 2D dynamic matrix control (2D-DMC) algorithm is proposed by integrating the learning nature of ILC into the prediction model of DMC [24,25]. The prediction model of DMC has had tremendous impacts on the process industry [3] and is relatively easy to design and analyze. The specific prediction model used in this paper takes into account the effects from the outputs from both the previous batch and the current batch and control input changes (along both time and batch dimensions). Based on this 2D prediction model, the developed 2D-DMC can be inherently decomposed into a conventional DMC, and an ILC optimized in the framework of MPC. Thus, the proposed algorithm is expected to retain the good control performance of DMC and the convergence property of ILC [26,27]. Convergence and robustness are critical for a control algorithm. This paper provides the sufficient conditions for the exponentially asymptotic and monotonic convergence of 2D-DMC, with the process operating repeatedly even with interval uncertainties. The ultimate tracking performance can be reached in spite of the existence of model-plant-mismatch and batch wise repeated disturbances. In industries, non-repeatable uncertainties are unavoidable. The designed controller guarantees robustness against non-repeatable interval uncertainties including the variant initial condition, to ensure the tracking error is within a known bound. The convergence and robustness of the proposed control scheme are demonstrated through application to thermoplastic injection molding. The results show that the batch process control performance is improved significantly.

The rest of this paper is organized as follows. Section 2 formulates a batch process using a step response. Section 3 presents the controller design by first deriving a 2D prediction model and then proposing the 2D-DMC algorithm. Section 4 provides the sufficient conditions for the convergence of the proposed algorithm with mathematic proof under repeatable and non-repeatable interval uncertainties. Simulation and application examples are given to illustrate the convergence and robustness of the proposed algorithm in Sections 5 and 6, respectively. Section 7 concludes.

2. Problem formulation

It is well known that DMC algorithms provide excellent ways of controlling multivariable processes. For simplicity, the derivation in this paper is based on a single-input single-output (SISO) discrete-time batch process. The results can be extended to the case of the multi-input multi-output (MIMO) system. Assume the following SISO process, described by the step response model, operates repetitively over a finite period of time (a batch):

$$\Sigma_p : \ y(t, k) = \sum_{i=1}^{\infty} a_i \Delta t u(t - i, k), \quad t = 0, 1, \ldots, T_k, \ k = 1, 2, \ldots$$

(1)

where $t$ and $k$ represent the discrete time index and the batch index, respectively. $T_k$ is the time duration of the $k$th batch. $y(t, k) \in \mathbb{R}$ and $u(t, k) \in \mathbb{R}$ denote the controlled variable (output) and the manipulated variable (input). $a_i$ are the sampled output values for the unitary step input; and $\Delta_t$ indicates the timewise back ward difference operator,

$$\Delta t f(t, k) = f(t, k) - f(t - 1, k).$$

(2)

In a batch process, the input affects the process from time 0 to $t$ in the following way:

$$\Delta_t u(t, k) = \begin{cases} 0, & t \in [-\infty, -1] \\ \Delta_t u(t, k), & t \in [0, T_k] \end{cases}$$

(3)

Substituting Eq. (3) into Eq. (1), model $\Sigma_p$ can be rewritten as

$$\Sigma_p : \ y(t, k) = \sum_{i=1}^{t} a_i \Delta t u(t - i, k), \quad t = 0, 1, \ldots, T_k, \ k = 1, 2, \ldots$$

(4)

We introduce the following notation:

$$g(t_1^k, k) = [g(t_1, k), g(t_1 + 1, k), \ldots, g(t_2, k)]^T, \ g \in \{y, u, r, w\}$$

(5)

where $g$ can be the output, input, updating law or noise variable from time $t_1$ to $t_2$. Model (4) can be reorganized into a more compact form as follows:

$$\Sigma_p : \ y(t, k) = \sum_{i=1}^{t} a_i \Delta t u(t - i, k) = A(t) \times \Delta t u(t - 1, k),$$

(6)

where $A(t) = [a_0 \quad \cdots \quad a_t]$.

Remark 1. In a continuous process, the process is required to be asymptotically stable for DMC to work; otherwise the output cannot be computed by the step response model, since the number of items included in the model will tend to infinity with time. However, in a batch process, each batch operates over a finite period of time. Thus, the number of items included in the model is inherently finite. In this study, a sufficiently large number $N$ is selected to represent the truncated horizon.

3. Controller design

To exploit the repetitive nature of a batch process, ILC is a good choice. To compensate for non-repeatable uncertainties, DMC will be adopted based on the step response model. This will lead to the integration of a feed-forward controller (for ILC) and a feedback controller (for DMC), giving rise to 2D-DMC. In this section, the 2D equivalent model will first be obtained by utilizing the ILC control law, so that it will have the learning ability. The 2D prediction model for DMC will then be derived from the 2D equivalent model. Finally, the control law of 2D-DMC will be developed based on the 2D prediction model in the framework of MPC.

3.1. The 2D equivalent model with repetitive nature

The ILC control law used in this study takes the following form:

$$r(t, k) = u(t, k) - u(t - 1, k) - (u(t, k - 1) - u(t - 1, k - 1))$$

$$= \Delta t u(t, k) - \Delta t u(t, k - 1)$$

(7)

where $r(t, k)$ is the updating law to be determined by the DMC strategy. Different from the conventional ILC algorithm, Eq. (7) is a 2D control law with a timewise integrator and a batchwise integrator cascaded [19].

From model (6), the input–output model can be derived as

$$y(t, k) = A(t) \times \Delta t u(t - 1, k) + w(t, k),$$

$$t = 0, 1, \ldots, T_k, \ k = 1, 2, \ldots$$

(8)
where \( w(t, k) \) is the uncertain and unmodeled dynamics. Substituting (7) into (8) leads to

\[
\Sigma_{2D-p} : y(t, k) = y(t, k - 1) + A(t) \times r(t_{\text{ret}, k}) + \Delta_k(w(t, k))
\]

where \( \Delta_k \) represents the batch wise backward difference operator, \( \Delta_k(f_k(t)) = f_k(t) - f_{k-1}(t) \). Model (9) is the 2D equivalent model and has the input \( r(t, k) \) instead of \( u(t, k) \) in model (8).

### 3.2. The 2D prediction model

To obtain the updating law, a 2D prediction model needs to be derived to obtain the prediction of the process. At time \( t \) of batch \( k \), according to the 2D equivalent model (9), the prediction \( \hat{y}(t+j | t, k) \) can be determined as follows:

\[
\hat{y}(t+j | t, k) = \hat{y}(t+j, k-1) + A(t+j) \times \hat{r}(t_{\text{ret}, k}) + \Delta_k(\hat{w}(t+j | t, k))
\]

\[
= y(t+j, k-1) + A(t) \times \hat{r}(t_{\text{ret}, k}) + \Delta_k(\hat{w}(t+j | t, k))
\]

\[
= y(t+j, k-1) + A(t) \times \hat{r}(t_{\text{ret}, k}) + \Delta_k(\hat{w}(t+j | t, k))
\]

Disturbances in the current batch from time \( t \) to \( t+j \) are considered to be constant and can be obtained from the 2D equivalent model (9) as

\[
\Delta_k(\hat{w}(t+j, k)) = \Delta_k(\hat{w}(t, k)) = y(t, k-1) - y(t, k-1) - A(t)
\]

\[
\times \hat{r}(t_{\text{ret}, k}) = \Delta_k(\hat{w}(y(t, k)) - A(t) \times \hat{r}(t_{\text{ret}, k})
\]

Then the prediction at time \( t+j \) can be estimated based on known past input \( \hat{r}(t_{\text{ret}, k}) \) and unknown current input and future input \( \hat{r}(t_{\text{ret}, k}) \) as

\[
\hat{y}(t+j | t, k) = y(t+j, k-1) + A(t) \times \hat{r}(t_{\text{ret}, k}) + A(t) \times \hat{r}(t_{\text{ret}, k}) + \Delta_k(y(t, k)) - A(t) \times \hat{r}(t_{\text{ret}, k}) + \Delta_k(\hat{w}(t, k)) + f(t+j, k)
\]

where \( f(t+j, k) \) is the free response of the system.

Assume \( p \) and \( m \) are the prediction horizon and control horizon, respectively, and \( p \geq m \geq 1 \). The prediction model is

\[
\begin{bmatrix}
    \hat{y}(t_{\text{ret}, k}) \\
    f(t_{\text{ret}, k})
\end{bmatrix} = G \times \begin{bmatrix}
    y(t+1) \\
    y(t+2) \\
    \vdots \\
    y(t+p)
\end{bmatrix} + H(t) \times \begin{bmatrix}
    \hat{r}(t+1) \\
    \hat{r}(t+2) \\
    \vdots \\
    \hat{r}(t+p)
\end{bmatrix}
\]

(13)

Define the system’s dynamic matrices of unknown input \( G \) and known input \( H(t) \) respectively as follows:

\[
G = \begin{bmatrix}
    a_1 & 0 & \cdots & 0 \\
    a_2 & a_1 & \cdots & 0 \\
    \vdots & \vdots & \ddots & \vdots \\
    a_m & a_{m-1} & \cdots & a_1 \\
    a_p & a_{p-1} & \cdots & a_{p-m+1}
\end{bmatrix}
\]

\[
H(t) = \begin{bmatrix}
    a_{t+1} & a_t & \cdots & a_1 \\
    a_{t+1} & a_t & \cdots & a_1 \\
    \vdots & \vdots & \ddots & \vdots \\
    a_{t+p} & a_{t+p-1} & \cdots & a_{1}
\end{bmatrix}
\]

(15)

\[
f(t+j+1 | t, k) \]

(16)

\[
effect of previous batch output
\]

\[
effect of current and future control actions on current batch
\]

(17)

\[
effect of pass control actions on the previous and current batches
\]

### 3.3. Cost function and control law

To design the controller, a sequence of control effort \( r \) is computed by minimizing the tracking error over a prediction horizon. At the same time, the excessive input movement necessary for doing so should be penalized. In this study, the cost function is chosen as

\[
f(t, k, m, p) = \frac{\|y(t+j | t, k) - y_k(t+j, t)\|^2}{Q} + \lambda \|f(t+j, k)\|^2_R
\]

(18)

The explicit solution to the above problem provides the control law of the proposed 2D-DMC scheme:

\[
\Sigma_{2D-DMC} : \hat{r}(t, k) = d\hat{T}(y(t+1 | t, k) - f(t+1 | t, k))
\]

(19)

where

\[
d\hat{T} = C_T(G^T Q + \lambda R)^{-1} C_T Q = \begin{bmatrix}
    d_1 & \cdots & d_p
\end{bmatrix}
\]

(20)

In this updating law, \( C_T \) with the dimension \( m \) is used to obtain the first row of the matrix; \( f(t+1 | t, k) \) and \( G(t, k) \) can be obtained from models (13) to (15). Fig. 1 shows the block diagram of the proposed 2D-DMC. It can be seen that the controller is actually an integration of prediction, control and tuning in the framework of 2D systems, using the information in both the current batch and previous batches.
2D-DMC is a 2D, step response, model-based control strategy. It utilizes the idea of online optimization by tuning in real-time. From the 2D system viewpoint, control law (19) can be inherently decomposed as
\[
\Delta_t(u(t, k)) = u_{DMC}(t, k) + u_{ILC}(t, k)
\]
where \(u_{DMC}(t, k)\) and \(u_{ILC}(t, k)\) are respectively determined by
\[
\Sigma_{DMC} : u_{DMC}(t, k) = d^T(y_i(t_{i+1}^t) - l_{p+1} \times y(t, k) - H(t) \\
\times \Delta_t u(t_{i+1}^t, k))
\]
and
\[
\Sigma_{ILC} : u_{ILC}(t, k) = u_{ILC}(t, k-1) + d^T(e(t_{i+1}^t, k-1))
\]
Clearly \(\Sigma_{DMC}\) has the same form as the usual DMC control strategy, which penalizes within-batch non-repetitive disturbances. \(\Sigma_{ILC}\) is an ILC control strategy that adjusts the input using the error from the previous batch over the prediction horizon. This strategy can reject disturbances along the batch dimension over the horizon. Thus, both repeatable and non-repeatable disturbances can be rejected in 2D-DMC. It is well known that MPC is a powerful tool for dealing with multiple variables and constraints [28]. 2D-DMC is proposed in the framework of MPC, thus it inherently can handle the interactions among variables and the constraints. The implementation procedure of 2D-DMC is given in Algorithm 1.

Algorithm 1.

1. Obtain the step response model coefficients \(a_i\).
2. Specify the design parameters by selecting the appropriate \(Q, R, \lambda, m\) and \(p\).
3. Perform initialization: \(k \leftarrow 1\) and \(t \leftarrow 0\).
4. Determine the dynamic matrices \(G\) and \(H(t)\) using Eqs. (14) and (15).
5. Calculate the control gains \(d^T = [d_1 \cdots d_p]\) using Eq. (20) and \(G\).
6. Given the output of batch \(k - 1\), the current output \(t\) of batch \(k\) and the past input of batch \(k\), calculate the free response \(f(t_{i+1}^t, k)\) based on \(H(t)\) using Eq. (13).
7. Given the set-point, design the updating law at step \(t\) of batch \(k\), \(P(t, k)\), based on \(d^T\) and \(f(t_{i+1}^t, k)\) using Eq. (19).
8. If the end of batch \(k\) is reached, jump to step (10); otherwise continue with step (9).
9. Let \(t \leftarrow t + 1\) and go back to step (4).
10. Let \(k \leftarrow k + 1\) and \(t \leftarrow 1\) and go back to step (4).

In the 2D-DMC framework, the prediction output will be out of a batch as the batch approaches the end. For the open-loop stable process, the output prediction out of a batch can be taken as the same value as that at the end of a batch, thus the prediction horizon will not shorten. For the open-loop unstable process, the prediction horizon will shorten and there is no control action for the output prediction out of a batch. In both cases, the duration of the current batch needs to be pre-determined. Actually, the end of a batch can be determined by, for example, checking the elapsing of time or certain conditions. The duration of a batch is available and constant if the elapsing of time is taken as the end condition of the batch.
process. If a logic check of certain conditions is to be performed, since the batch process operates repeatedly, the end of a batch process can be predicted using information from past batches. For ease of implementation, the duration of the current batch can be taken as that of previous batch. Note that in the following experiment, the injection velocity process ends when the preset injection stroke is reached and good control performance is obtained.

4. Analysis of convergence and robustness

In this section, both the convergence and robustness properties of 2D-DMC are investigated. When the uncertainties are repeatable, the sufficient conditions that guarantee that the tracking error decreases to zero will be provided. When interval non-repeatable uncertainties exist, the tracking error converges to a known bound, which indicates the robustness of the proposed algorithm. The tracking error of 2D-DMC evolving in 2D complicates the analysis. In this paper, the tracking errors in each batch will be combined to form an error vector. Thus, the 2D analysis problem is transformed into a one-dimensional (1D) one. The following definition and assumptions are made for the analysis.

Definition 1. The tracking error is defined as

\[ e(t, k) = y_r(t) - y(t, k) \]  

which is assumed to be zero, i.e.,

\[ e(T + 1, k) = e(T + 2, k) = \cdots = e(T + p - 1, k) = 0 \]  

when the time index exceeds the duration of a batch.

Assumption 1. The durations of different batches are assumed to be constant, i.e.,

\[ T_k \triangleq T \]  

Assumption 2. The uncertainties are described in general sense as follows:

\[ w(t, k) = w_1(t) + w_2(t, k), \quad t = 0, 1, \ldots, T, \quad k = 1, 2, \ldots \]  

where

\[ u_k^0 \triangleq \Delta_k(w_2(0, k)), \quad u_k \triangleq \Delta_k(w_2(k^0, k)) \]  

where \( w_1(t) \) and \( w_2(t, k) \) represent the repeatable and non-repeatable uncertainties, respectively. \( u_k^0 \) and \( u_k \) denote the initial non-repeatable uncertainties and the non-repeatable uncertainties within a batch, respectively. They vary in the following interval:

\[ ||u_{k+1}^0 - u_k^0|| \leq b_{\alpha}, \quad \text{and} \quad ||u_{k+1} - u_k|| \leq b_v \]  

4.1. The model of the closed-loop control system

Before moving to the convergence proof, the model of the closed-loop control system needs to be acquired first. Substituting Eq. (27) into the 2D equivalent model (9), the following 2D equivalent model without repeatable uncertainties can be obtained:

\[ \Sigma_{2D,p} : y(t, k) = y(t, k - 1) + A(t) \times r(t_{t-1}^{t_{t+1}}) + \Delta_k(w_2(t)) \] 

\[ + \Delta_k(w_2(t, k)) = y(t, k - 1) + A(t) \times r(t_{t-1}^{t_{t+1}}) + \Delta_k(w_2(t, k)), \quad t = 0, 1, \ldots, T, \quad k = 1, 2, \ldots \]  

Substituting Eq. (29) into the control law (19), the relationship between the input and the error is obtained as

\[ \Sigma_{2D-DMC} : r(t, k) = d^T(e(t_{t-1}^{t_{t+1}}) - y(t_{t-1}^{t_{t+1}}) - 1) - I_p \times (A(t)) \times r(t_{t-1}^{t_{t+1}}) + \Delta_k(w_2(t, k)) = d^T(e(t_{t-1}^{t_{t+1}}) - y(t_{t-1}^{t_{t+1}}) - 1) \] 

\[ - I_p \times \Delta_k(w_2(t, k)) - H(t) \times r(t_{t-1}^{t_{t+1}}) \]  

where

\[ r(0, k) = \sum_{i=1}^{p} d_i e(i, k) - \sum_{i=1}^{p} d_i \cdot \Delta_k(w_2(0, k)) \] 

\[ r(1, k) = \sum_{i=1}^{p} d_i e(i + 1, k - 1) - \sum_{i=1}^{p} d_i a_{i+1} r(0, k) \] 

\[ - \sum_{i=1}^{p} d_i \cdot \Delta_k(w_2(1, k)) \] 

\[ r(2, k) = \sum_{i=1}^{p} d_i e(i + 2, k - 1) - \sum_{i=1}^{p} d_i a_{i+2} r(0, k) \] 

\[ - \sum_{i=1}^{p} d_i a_{i+1} r(1, k) - \sum_{i=1}^{p} d_i \cdot \Delta_k(w_2(2, k)) \] 

\[ \vdots \] 

\[ r(t, k) = \sum_{i=1}^{p} d_i e(i + t - 1, k - 1) - \sum_{j=0}^{t-2} \left( \sum_{i=1}^{p} d_i a_{i+j} \right) r(j, k) \] 

\[ - \sum_{i=1}^{p} d_i \cdot \Delta_k(w_2(t - 1, k)), \]  

which can be rewritten in the vector form as follows:

\[ \begin{bmatrix} r(t + 1, k) \\ \vdots \\ r(t - 1, k) \\ r(0, k) \end{bmatrix} = \begin{bmatrix} d_1 & d_2 & \cdots & d_p & 0 & \cdots & 0 \\ 0 & d_1 & d_2 & \cdots & d_p & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & d_1 & d_2 & \cdots & \cdots & \cdots \end{bmatrix}_{r \times (p+1)} \times \begin{bmatrix} e(1, k-1) \\ e(2, k-1) \\ \vdots \\ e(t-2, k-1) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{p} d_i a_{i+1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix}_{r \times (p+1)} \times \begin{bmatrix} e(1, k-1) \\ e(2, k-1) \\ \vdots \\ e(t-2, k-1) \end{bmatrix} = \begin{bmatrix} \Delta_k(w_2(t)) \\ \Delta_k(w_2(t, k)) \end{bmatrix} = \sum_{i=1}^{p} d_i \]  

To obtain the model of the closed-loop control system, the explicit description of \( r(t_{t-1}^{t_{t+1}}, k) \) should be derived from the control law (30) and then substituted into model (29). Taking \( t \) from 0 to \( t - 1 \) in (30) leads to
Thus, the explicit description of \( r^i_{t-1, k} \) is
\[
\begin{align*}
\begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
\sum_{i=1}^{p} d_i a_{i+1} & 1 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
\sum_{i=1}^{p} d_i a_{i+p-1} & \sum_{i=1}^{p} d_i a_{i+1} & 1 & \cdots & 0 \\
0 & 0 & d_1 & d_2 & \cdots & d_p \\
0 & 0 & 0 & d_1 & d_2 & \cdots & d_p \\
\end{bmatrix}
\end{align*}
\]
(33)

\( DA(t) \) is a lower triangular matrix with ones in the diagonal, so the inverse of \( DA(t) \) is always valid. The model of the closed-loop system can be achieved by substituting Eq. (33) into model (29), giving
\[
\Sigma_{2D-p-c} : y(t, k) = y(t, k - 1) + A^t_{\text{SOH}} \times \Delta_k(w_2(t, k)) \]
+ \( \Delta_k(w_2(t, k)) \)
\[
+ \Delta_k(w_2(t, k)) 
\]
(34)

Based on this model, the corresponding tracking error evolution law will be derived next.

4.2. Tracking error evolution law and convergence conditions

From Definition 1, it is easy to derive
\[
\Delta_k(y(t, k)) = -\Delta_k(e(t, k)) 
\]
(35)

Combining Eq. (34) with Eq. (35) leads to
\[
\Sigma_{2D-p-c} : e(t, k) = LE_t \times e(t|_{t-1, k}) + LW_t \times \Delta_k(w_2(t, k)) 
\]
where
\[
LE_t = \begin{bmatrix}
0 & \cdots & 0 & 1_t & 0 & \cdots & 0 \\
\end{bmatrix}
\]
\[
LW_t = \sum_{i=1}^{p} d_i A^t_{\text{SOH}} \times DA(t)^{-1}
\]
(36)

\[
[0 \cdots 1_t \ 0 \cdots 0]_{t+ \cdots + (t+p-1)} \quad \text{represents a} \quad (t+p-1) \quad \text{-dimensional row vector, where the} \quad t \text{-th element is 1 and the}
\]

| \( e(1, k) \) | \( e(2, k) \) | \( \cdots \) | \( e(T, k) \) |
| eapply \( LE_t(1) \) | \( LE_t(2) \) | \( \cdots \) | \( LE_t(p) \) | \( 0 \) | \( 0 \) | \( \cdots \) | \( 0 \) |
| \( LE_t(1) \) | \( LE_t(2) \) | \( \cdots \) | \( LE_t(p+1) \) | \( 0 \) | \( 0 \) | \( \cdots \) | \( 0 \) |
| \( LE_t(1) \) | \( LE_t(2) \) | \( \cdots \) | \( LE_t(p) \) | \( LE_t(p+1) \) | \( \cdots \) | \( \cdots \) | \( \cdots \) |
| \( e(p+T-1, k) \) | \( LW_t(1) \) | \( \cdots \) | \( \cdots \) | \( \cdots \) | \( \cdots \) | \( \cdots \) | \( \cdots \) |

\[
\begin{bmatrix}
-L1 & 0 & \cdots & 0 \\
LW_t(2) & -1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
LW_t(2) & LW_t(2) & \cdots & -1 \\
\end{bmatrix} \times \Delta_k(w_2(T, k)) 
\]
(38)
Combining the above equation with Eq. (25) leads to the 1D (the batch dimension only) tracking error evolution law

\[ E_k = \phi \times E_{k-1} + \xi \times u_k^0 + \psi \times u_k \]

where

\[
\begin{bmatrix}
LE_1(1) & LE_2(1) & \cdots & LE_p(1) \\
LE_1(2) & LE_2(2) & \cdots & LE_p(2) \\
\vdots & \vdots & \ddots & \vdots \\
LE_1(p) & LE_2(p) & \cdots & LE_p(p)
\end{bmatrix} \times
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

beginning of the batch to \( T_{\min} \) by Theorems 1 and 2, where \( T_{\min} \) is the minimum duration of \( k \) batch given by

\[ T_{\min} = \min(T_1, T_2, \ldots, T_k) \quad (43) \]

where \( \phi, \xi, \) and \( \psi \) are known constant matrices determined by the system step response coefficients. For the process with completely repeatable uncertainties, where \( b_i = 0 \) and \( b_i \) are both zero, the tracking error evolution law is simplified to

\[ E_k = \phi \times E_{k-1} \quad (40) \]

\( E_k \) is exponentially asymptotically convergent, if the \( \phi \) is stable, that is, the eigenvalue of \( \phi \) is less that 1 [29]. The convergence condition can also be defined in the norm topology [30]. Therefore, the convergence theorems can be stated in two different topologies as follows.

**Theorem 1.** For the system (8) with Assumptions 1 and 2 and \( b_i = 0 \), using the control updating law (19), \( E_k \) will exponentially asymptotically converge to zero as \( k \to \infty \),

\[ \lim_{k \to \infty} \| E_k \| = 0 \quad (41) \]

if \( \| \lambda \| < 1 \).

**Theorem 2.** For the system (8) with Assumptions 1 and 2 and \( b_i = 0 \), using the control updating law (19), \( E_k \) will monotonically converge to zero as \( k \to \infty \) in the \( l_i \), \( i \in \{1, 2, \infty\} \) norm topology,

\[ \lim_{k \to \infty} \| E_k \| = 0 \quad (42) \]

if \( \| \phi \| < 1 \).

**Remark 2.** From Theorems 1 and 2, it can be seen that the 2D-DMC guarantees the ultimate tracking performance with repeatable uncertainties. Moreover, the performance does not depend on the preset trajectory in the derivation procedure of theorems. From the application viewpoint, the proposed algorithm can track the profile tightly even under model-plant-mismatches, effectively combating the main deficiency of MPC.

**Assumption 1** is used in Eq. (37) to obtain the 1D tracking error evolution law Eq. (39). If the duration \( T_k \) of the batch varies, the closed-loop control system is guaranteed to converge from the beginning of the batch to \( T_{\min} \) by Theorems 1 and 2, where \( T_{\min} \) is the minimum duration of \( k \) batch given by

\[ T_{\min} = \min(T_1, T_2, \ldots, T_k) \quad (43) \]

For the output at \( t > T_{\min} \), the closed-loop control system can still converge. That is, the control performance at time \( t \) can be improved once when there is an output at time \( t \) at any following batch. Actually, the duration of batch will converge as well with the convergence of the closed-loop system. The difference is that the output generated during the time before \( T_{\min} \) is up will converge faster than the output generated during the time after \( T_{\min} \) is up. In the following experimental test, it should be noted that \( T_k \) varies along the batch dimension, but the process can still be made to converge.

The 2D-DMC is proposed under the framework of DMC and ILC, the convergence of which can be guaranteed by Theorems 1 and 2. It should be noted that the convergence analysis conducted in this study is along the batch dimension. The smaller the value of the convergence condition leads to the faster the convergence rate. However, the tracking performance along the time dimension may be degraded during the convergence process. The guidelines for tuning the parameters of the proposed 2D-DMC are summaries as follows.

The weighting factor \( \lambda \) allows the variable being manipulated to be weighted. Decreasing the value of \( \lambda \) tends to increase the convergence rate of the closed-loop system along the batch dimension by reducing the value of the convergence condition. This will however lead to large fluctuations of the input. The weighting factor has the most important role in determining the control performance of 2D-DMC.

As the prediction horizon \( P \) decreases, the controller tends to increase the convergence rate of the closed-loop system along the batch dimension by reducing the value of the convergence condition. This will however lead to poor control performance before the process is convergent. For example, the variable under control will oscillate in the dynamics process in the early batches. The prediction horizon is often selected to cover the entire dynamics of the process.

Decreasing the value of the control horizon \( M \) tends to decrease the convergence rate of the closed-loop system along the batch dimension by increasing the value of the convergence condition. Moreover, the computational load increases with the control horizon.
4.3. Robustness analysis

The convergence condition for the process with repeatable uncertainties has been studied above. However, non-repeatable uncertainties exist inevitably in practice. In the following, the robustness of the control algorithm against non-repeatable uncertainties will be analyzed.

Firstly, let us introduce the following Lyapunov function for the analysis of the tracking error in Eq. (39),

$$V_k = E_k^T E_k$$

where $F$ is the solution of the Lyapunov function following

$$\phi^T F\phi - F = -I$$

$I$ is the unit matrix with a proper dimension. There is a feasible solution for $F$, if $\phi$ is stable, which means the convergence theorem derived in the previous subsection must be satisfied. Let $\lambda_{\min}(F)$ and $\lambda_{\max}(F)$ be the smallest and largest eigenvalues of $F$, respectively. Then,

$$\lambda_{\min}(F)\|E_k\|^2 \leq V_k \leq \lambda_{\max}(F)\|E_k\|^2$$

From the Lyapunov function (44) and Eq. (39),

$$\Delta_k V_{k+1} = V_{k+1} - V_k$$

$$= (\phi \times E_k + \xi \times u_{k+1} + \psi \times u_k + \psi \times u_{k+1})^T F(\phi \times E_k + \xi \times u_{k+1} + \psi \times u_k + \psi \times u_{k+1})$$

$$- E_k^T F E_k = E_k^T \phi^T F \phi E_k - E_k^T F E_k - (\xi \times u_{k+1})^T F (\xi \times u_{k+1})$$

$$+ 2E_k^T \phi^T F \psi u_{k+1} + 2E_k^T \phi^T F \psi u_k + 2(\xi \times u_{k+1})^T F (\xi \times u_{k+1})$$

Based on Assumption 2, the following triangle inequalities can be obtained,

$$(\xi \times u_{k+1})^T F (\xi \times u_{k+1}) \leq \|\xi\|^2 F_{i=1}^2$$

$$\psi^T F \psi u_{k+1} \leq T^2\|\psi\|^2 F_{i=1}^2$$

$$2E_k^T \phi^T F \psi u_k \leq \frac{1}{4}E_k^T F E_k + 4T\|\psi\|^2 F_{i=1}^2$$

$$2E_k^T \phi^T F \psi u_{k+1} \leq \frac{1}{4}E_k^T F E_k + 4T\|\psi\|^2 F_{i=1}^2$$

$$2(\xi \times u_{k+1})^T F (\xi \times u_{k+1}) \leq 2T\|\xi\|^2 F_{i=1}^2$$

By using Eq. (48), one can transform Eq. (47) into

$$\Delta_k V_{k+1} \leq \left(-1 + \frac{1}{4} + \frac{1}{4}\right)\|E_k\|^2 + f(b_0, b_0)$$

$$f(b_0, b_0) = \|\xi\|^2 F_{i=1}^2 + T^2\|\psi\|^2 F_{i=1}^2 + 2T\|\xi\|^2 F_{i=1}^2 + 4T\|\psi\|^2 F_{i=1}^2 + 4T\|\psi\|^2 F_{i=1}^2$$

On the basis of Eq. (46), it is easy to obtain

$$V_{k+1} - V_k \leq -\frac{1}{2}\|E_k\|^2 + f(b_0, b_0)$$

It is also easy to find a constant $\varepsilon$ satisfying

$$\frac{1}{2\lambda_{\max}(F)} \geq \varepsilon > 0 \text{ and } \varepsilon < 1$$

Thus, Eq. (50) can be rewritten as

$$V_{k+1} - V_k \leq -\varepsilon V_k + f(b_0, b_0)$$

It is easy to obtain the following inequality from Eq. (52),

$$V_k \leq (1 - \varepsilon)^k V_0 + \frac{1 - (1 - \varepsilon)^k}{1 - (1 - \varepsilon)} f(b_0, b_0)$$

since $0 < \varepsilon < 1$,

$$\lim_{k\to\infty} V_k \leq \lim_{k\to\infty} (1 - \varepsilon)^k V_0 + \frac{1 - (1 - \varepsilon)^k}{1 - (1 - \varepsilon)} f(b_0, b_0)$$

Combining the Lyapunov functions (44) and Eq. (46) with Eq. (54), it is easy to achieve

$$\lim_{k\to\infty} \|E_k\| \leq \frac{f(b_0, b_0)}{\varepsilon\lambda_{\min}(F)}$$

Eq. (55) gives a description of the tracking error in the system with non-repeatable uncertainties. The system can still converge to a known bound. The following theorems summarize the above derivations.

**Theorem 3.** Based on Assumptions 1 and 2 and the control updating law (19), the tracking error $E_k$ of the system (8) will exponentially asymptotically converge to a known bound as $k \to \infty$,

$$\lim_{k\to\infty} \|E_k\| \leq \frac{f(b_0, b_0)}{\varepsilon\lambda_{\min}(F)} \text{ if } \|\phi\|_1 < 1.$$
Fig. 3. The convergence control test with model mismatches.

Fig. 4. An enlarged view of Fig. 3.
5. Simulation illustration

5.1. Simulation 1: injection velocity

In this study, injection molding, a typical batch process, is adopted as an example for demonstration purpose. Injection molding is a key manufacturing process in which plastic materials (granules) are converted into various products. The injection molding process consists of several phases, with the filling phase being an important one as it is in this phase that the product qualities such as surface quality and dimension are determined. The injection velocity (IV, y(t, k)) is the variable to be controlled in the filling phase and is manipulated by the valve opening (VO, u(t, k)) following the transfer function [19].

\[ Y(s, k) = \frac{320}{10,000s^3 + 3050s^2 + 215s + 1}U(s, k) + W(s, k) \]

\[ k = 1, 2, \ldots \] (58)

Eq. (58) is the model of a real manufacturing plant with the disturbance W(s, k). Applying a step change to the control valve gives the response model

\[ y(t, k) = \sum_{i=1}^{t} a_i \Delta u(t-i, k) + w(t, k), \quad t = 1, \ldots, T_k, \quad k = 1, 2, \ldots \] (59)

where \( T_k \) is 200 steps and is assumed to be invariant in this simulation, the coefficient sequences \( a_i \) are shown in Table 1 and the responses form the dashed line in Fig. 2. The corresponding transfer function is

\[ Y(s, k) = \frac{290e^{-10s}}{2605s + 1}U(s, k), \quad k = 1, 2, \ldots \] (60)

The model-plant-mismatch between Eq. (60) and the plant is illustrated in Fig. 2, which shows that the mismatch is significant. The set-point of \( y(t, k) \) is chosen as the step change with a 5-step transition time from time 101 to 104.

\[ y_r(t, k) = \begin{cases} 40, & t = 1, 2, \ldots, 100 \\ 40 - (t - 100) \times 4, & t = 101, 102, \ldots, 104 \\ 20, & t = 105, 106, \ldots, 200 \end{cases} \] (61)

which is difficult to track.

To demonstrate the effectiveness of the proposed control scheme, several cases are studied. In the following simulations and experiments, the sum of squared error (SSE) between the set-point

![Fig. 5. Tracking performance without disturbance.](image)

![Fig. 6. Result of the simulation of the proposed 2D-DMC with repeatable disturbance.](image)
and the output measurement is adopted to evaluate the tracking performance.

\[ \text{SSE}(k) = \sum_{t=1}^{T_k} (y_r(t, k) - y(t, k))^2 \]  

**Case 1 (Convergence test).** The procedures for tuning the DMC parameters are given by Morari and Lee [3]. In this simulation, considering the trade-off between the convergence rate and the robustness of the controller, the following parameters are chosen: \( Q = I, R = I, \lambda = 50, p = 50 \) and \( m = 10 \). The disturbance \( w(t, k) \) is set to zero for the convergence test. Fig. 3 shows the output response and the input signal of the control system. It can be clearly seen that the trajectory is tightly tracked even with significant model-plant-mismatch. Since this mismatch can be considered as repetitive disturbance, it can be handled by the ILC framework in the proposed control scheme. Fig. 4 provides an enlarged view of the start-up and transition part of Fig. 3. It is shown that the dynamic tracking performance improves gradually from batch to batch. Fig. 5 shows the fast convergence rate of the control system. In the following case studies, the total number of test batches is set to 40.

Two cases are conducted to test the robustness of the proposed control scheme.

**Case 2 (Repetitive disturbances).** The disturbance in this case is chosen as

\[ w(t, k) = 5 \times \sin(0.04\pi t) \]  

\( w(t, k) \) changes in the time dimension only and can be simplified as \( w(t) \). It represents the sinusoidal repeatable disturbances. Fig. 6 shows that the output tracks the set-point perfectly by adopting the input signal in a sinusoidal fashion to reject the sinusoidal disturbance. Fig. 7 shows the fast convergence rate which converges monotonically. This indicates that the implementation of ILC in the proposed control scheme helps to handle the repetitive disturbances. The traditional ILC is also used to control the process. The results are presented in Figs. 8 and 9 which show that the traditional ILC handles the repetitive disturbances in the final batches quite well. However, the control performance in the early batches leaves much to be desired as can be seen from Fig. 8 which shows significant oscillation due to the sinusoidal disturbances and the lack of feedback action. Thus, more batches are required for tracking the set point to converge as shown in Fig. 9.
Case 3 (Non-repetitive disturbances). The disturbance in this case is assumed as
\[ w(t, k) = 5 \times \sin(0.04\pi t) + 0.5 \times \eta(t, k) \]  
(64)

Compared with Case 2, an additional term \( 0.5 \times \eta(t, k) \) is added to \( w(t, k) \). \( \eta(t, k) \) is a pseudorandom variable evenly distributed between the interval of \([-1, 1]\) with zero mean and represents the non-repetitive disturbances. The control results are illustrated in Figs. 10 and 11. There exist small perturbations in the output response and around the set-point change which is due to the feedback actions of the proposed control scheme, indicating the good control performance of the proposed control scheme in the presence of non-repetitive disturbances.

5.2. Simulation 2: a non-stationary batch process

Consider the following process:
\[ y(t, k) = \frac{0.2z^{-1}}{1 - 1.1z^{-1}} u(t, k) + w(t, k), \quad k = 1, 2, \ldots; \quad t = 1, 2, \ldots, T \]
\[ w(t, k) = 5 \times \sin(0.04\pi t) + \eta(t, k) \]  
(65)

The step response of this process is shown in Fig. 12 and is obviously unstable. The preset profile is chosen as a series of set-point changes. The control results are depicted in Figs. 13 and 14; note that the process is open-loop unstable, yet stable control with
tight tracking performance can be achieved by the proposed 2D-DMC regardless.

6. Experiments

The proposed method is used to control the injection velocity in an industrial-size reciprocating-screw injection-molding machine (Chen-Hsong MINIJET Intelligent Series, model no. MJ55). The machine is powered by an electric motor (Model Number Y132M-4B35) with a capacity of 7.5 kW. A four-way three-position solenoid valve (Rexroth 4WE10L3X;DG4V-5-3CL) is used to control the screw operation. Proportional electro-hydraulic relief and flow valves (Rexroth 0811402017 and Rexroth 0811403017, ...
Table 2
Coefficients of the model used in application.

<table>
<thead>
<tr>
<th>i</th>
<th>(a_i)</th>
<th>i</th>
<th>(a_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>11</td>
<td>234.5</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>12</td>
<td>258.7</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>13</td>
<td>277.4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>14</td>
<td>291.4</td>
</tr>
<tr>
<td>5</td>
<td>14.1</td>
<td>15</td>
<td>301.3</td>
</tr>
<tr>
<td>6</td>
<td>46.3</td>
<td>16</td>
<td>308.1</td>
</tr>
<tr>
<td>7</td>
<td>86.8</td>
<td>17</td>
<td>312.4</td>
</tr>
<tr>
<td>8</td>
<td>129.1</td>
<td>18</td>
<td>314.8</td>
</tr>
<tr>
<td>9</td>
<td>169.2</td>
<td>19</td>
<td>315.9</td>
</tr>
<tr>
<td>10</td>
<td>204.7</td>
<td>20</td>
<td>316.2</td>
</tr>
</tbody>
</table>

Fig. 15. Output responses in the model and plant in experiment.

respectively) are equipped to control the hydraulic pressure and flow rate. A displacement sensor (115L6E) is installed to measure the screw stroke. It should be noted that, in this application, there is no dedicated injection velocity sensor. A smart data processing system is used for velocity measurement by differencing the displacement signal. A semi-crystalline material, high-density polyethylene, is used for the molding.

The injection velocity (IV, \(y(t, k)\)) and the valve opening (VO, \(u(t, k)\)) of the control valves are the controlled variable and the manipulated variable, respectively. The set-point of \(y(t, k)\) is chosen as the step change with five transition steps from step 81 to 84 as well, i.e.,

\[
y_r(t, k) = \begin{cases} 
25, & t = 1, 2, \ldots, 80 \\
25 + (t - 80) \times 2, & t = 81, 82, \ldots, 84 \\
35, & t = 85, 86, \ldots, 149 
\end{cases}
\]

An open loop test with the valves being 43–46% open based on the set-point is firstly applied to the process. A step response model with a four-step delay can be obtained as shown in Table 2. With this

Fig. 16. Tracking performance in application.

Fig. 17. Result of the experimental of the proposed 2D-DMC.
7. Conclusions

In this paper, a batch process is first formulated as a 2D step response model by taking into consideration the repetitive nature of the process. Based on this model, the 2D-DMC, an inherent integration of ILC and DMC, is developed. It is shown mathematically that convergence is guaranteed with the proposed method. The sufficient condition for convergence is established. The proposed control scheme is also successfully tested through its application to thermoplastic injection molding, a typical batch process. The obtained control results clearly indicate the good convergence and robustness of the proposed scheme under different operating conditions.

Acknowledgments

This work is supported in part by the NSFC/RGC Joint Research Scheme (project number N-HKUST639/09), two NSFC grants (project numbers 60931160440 and 61104058) and Guangzhou Scientific and Technological project (project number 11F11140010).

References


