Journal of Manufacturing Systems 32 (2013) 372–381

Contents lists available at SciVerse ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

Coordination and control of batch-based multistage processes

Kai Han, Kaibo Wang*

Department of Industrial Engineering, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 3 July 2012
Received in revised form 31 October 2012
Accepted 11 December 2012
Available online 20 January 2013

Keywords:
Batch allocation
Multistage processes
Run-to-run control
Clustering algorithm

ABSTRACT

Run-to-run (R2R) process control has attracted much attention in research and has been widely used in practice. It has been proved effective at compensating for process disturbances by using R2R controllers at a single stage. However, most manufacturing processes span across multiple stages; variation in earlier stages can be magnified stage by stage if they are not properly eliminated. In addition, products are processed batch by batch in certain manufacturing processes. In such cases, the traditional EWMA controller might not effectively reduce the variation. This paper focuses on developing a process control strategy for batch production in a multistage process. In the newly proposed framework, a batch-allocation operation is introduced to group products into similar clusters before each stage; an R2R controller is then implemented to generate customized recipes for each batch. This framework emphasizes better coordination among the stages in a multistage process. Simulation results show that the proposed strategy is effective for the reduction of variation.

1. Introduction

In semiconductor manufacturing processes, products are always processed run by run. Each run is an undividable cycle within which measurements of quality results are unavailable. Process drifts and shifts can occur between different runs for many reasons, such as the tools wearing out, operators switching, the working conditions changing, and the machines becoming unstable. To compensate for these process drifts or shifts, run-to-run (R2R) process control methods have attracted extensive attention in quality control research and have been widely used in practice [1]. Such controllers usually use outputs from previous runs to generate optimal recipes for new runs, to seek better compensation for process disturbances and to achieve higher quality [2]. Because most manufacturing processes naturally span multiple stages, and variations in such multistage processes tend to propagate across stages [3], it is important to implement appropriate process control algorithms, coordinating inputs and outputs across multiple stages, optimizing recipes for each stage and minimizing the output variation [4].

Consider the wafer preparation process as an example. As shown in Fig. 1, there are almost ten major steps from crystal pulling to wafer packaging. Each stage takes the output from its preceding stage as its input and sends its output farther down, to its downstream stages. Thus, the deviations at one stage are transferred into subsequent stages; if no effective control actions are taken, then such deviations become larger and larger and could finally harm the quality of the finished products.

In the wafer preparation process, another important issue is that wafers are processed in batches with variable sizes. For example, a slicing run can generate 300 wafers. These wafers are put together and moved through the rest production stages. When these wafers reach the lapping stage, since the lapping machine can only handle 50 wafers each time (this is limited by the capacity of the lapping plate), these wafers have to be clustered into 6 batches and be processed sequentially. Similarly, when these 300 wafers reach the etching stage, since the capacity of the etching chamber is 100, these wafers need be clustered into 3 batches. Therefore, wafers must be grouped or un-grouped between stages to satisfy the batch size limitation of each stage.

This batch-based feature is in fact quite commonly seen in semiconductor manufacturing because of the equipment capacity constraint. Therefore, to achieve better quality in a multistage manufacturing process, the following two problems become critical:

(a) How to allocate products (wafers, in the example studied in this work) in appropriate batches so that the transmission of variation is minimized and the final product quality is optimized.
(b) How to design control strategies (generate optimal recipes) for each stage at each run by considering both the grouping information and the output information from previous runs.

Intuitively, unique opportunities exist in controlling a multistage process. For example, the bow of a wafer (defined in Fig. 2) measures its curvature. A wafer with a large bow value after slicing could be improved if a faster rotation speed is used in the lapping

* Corresponding author.
E-mail address: kbwang@tsinghua.edu.cn (K. Wang).

0278-6125/$ – see front matter © 2012 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.jmsy.2012.12.004

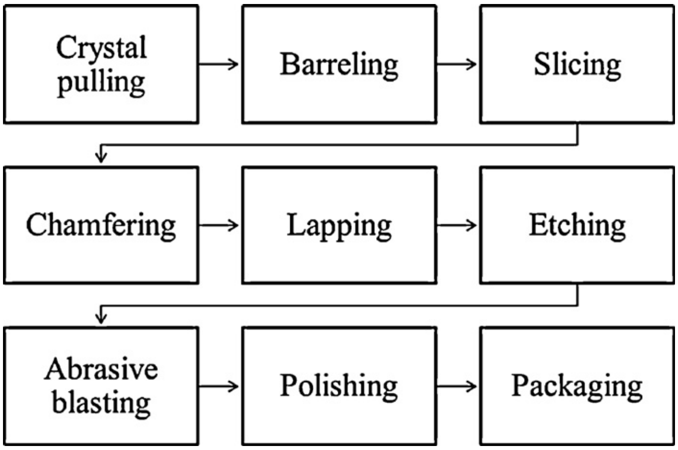

Fig. 1. The wafer preparation process.

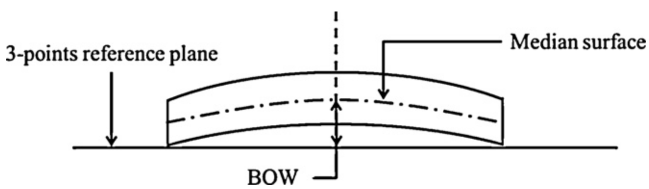

Fig. 2. Definition of "bow" for a wafer.

stage or less time is used in the etching stage. In other words, in a multistage process, there are opportunities for downstream stages to systematically compensate for deviations that resulted from upstream stages by using customized recipes. Therefore, the implementation of process control by considering the coordination of multiple stages is important in such a process.

The objective of this work is to develop a run-to-run process control framework for a multistage manufacturing process (MMP). In the multistage R2R control setting, the variation in the incoming information from an upstream stage could be treated as observable but uncontrollable noise that moves to a downstream stage. Such noises could be compensated by well-designed batch-allocation strategies and control actions. This idea is well illustrated in Fig. 3.

Compared to the conventional R2R control framework for a single stage, there are two key features in the framework in Fig. 3. First, a batch allocator is added to the model. The outputs from stage k, which have a large overall variation, are split into batches, each of which have a smaller variation, and are fed into stage $k+1$. The core of the batch allocator is a clustering algorithm for reducing *within-batch* variation. Second, the R2R controller equipped for each stage is updated to take batch information as one of its inputs. A customized recipe can be generated for each batch to reduce *batch-to-batch* variation. Therefore, the batch allocator makes it possible to have a finely tuned recipe, and an implementation of the R2R controller that is based on both the feedback and the feedforward information is expected to be effective at reducing the propagation of variation.

The rest of the paper unfolds as follows. In Section 2, a literature review on the recent research that is relevant to R2R process control and multistage process modeling is first presented. In Section 3, the batch-based R2R control strategy for a single stage is shown. Section 4 extends the framework to a process having multiple stages. In Section 5, the performance of the proposed control framework is studied and is compared with an existing controller. Finally, Section 6 concludes this work with suggestions for future research.

2. Literature review on run-to-run control and multistage process modeling

Conventionally, R2R controllers have been designed to compensate for process disturbances in a single stage. Among others, the exponentially weighted moving average (EWMA) controller [5] and the double EWMA controller [6] have been extensively studied because of their simplicity and robust performance. Various extensions of these controllers also appear in the literature. Del Castillo and Hurwitz [1] provided a literature review of R2R control methods from a statistical and control engineering point of view and proposed a self-tuning controller based on the recursive least square estimation method to provide better control performance. Those control filters are all developed using quantitative rather than qualitative measurements. Wang and Tsung [7], Shang et al. [8], Wang and Tsung [9] and Lin and Wang [10] proposed an R2R control scheme that uses categorical measurements for process adjustment. However, most of the methods in the literature utilize a single-output model.

Tseng et al. [11] investigated the control of a multiple-inputs–multiple-outputs (MIMO) process using a multivariate exponentially weighted moving average (MEWMA) controller with a variable discount factor. The authors showed the feasible region and the approximate solution for the optimal discount factor. Del Castillo and Rajagopal [12] also extended the univariate double EWMA method to a MIMO double EWMA controller and pointed out that the performance of the MIMO controller is superior to that of several single-input–single-output (SISO) controllers. Fan and Wang [13] extended the EWMA controller based on a neural network online tuning approach for SISO models [14] to a double EWMA controller for MIMO models. Jen et al. [15] also focused

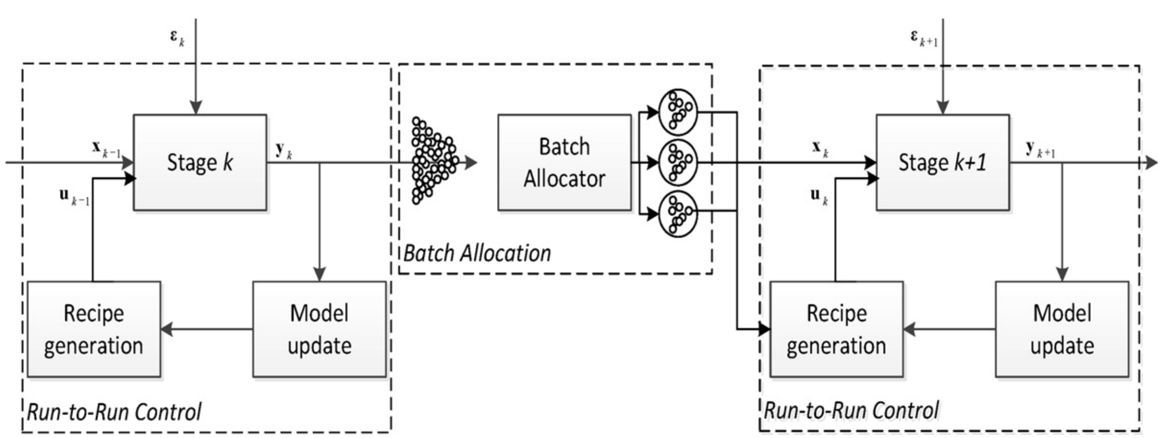

Fig. 3. A batch-based R2R control framework for multistage manufacturing processes.

on the development of an R2R controller for MIMO processes. These authors set up a self-tuning control framework instead of EWMA and updated process parameters by a recursive least squares (RLS) algorithm. He et al. [16] proposed a General Harmonic Rule controller. Details of several popular techniques for R2R control strategy other than the EWMA algorithm and their extensions to MIMO processes can also be found in the literature.

Campbell et al. [17] surveyed four types of popular R2R control methodologies, including EWMA control, predictor-corrector control (PCC), model predictive control (MPC), and optimizing adaptive quality control (QAQC). Wang et al. [18] discussed and compared the performance of EWMA, double EWMA, IMC, and RLS controllers in the presence of a drift, a step disturbance, measurement noises and modeling errors. Additionally, Chen et al. [19] presented a universal methodology for accessing the performance of R2R controllers in semiconductor manufacturing based on an internal model control (IMC) structure, and they validated their methodology by using examples of EWMA control, double EWMA control and recursive-least-squares locally linear-trend (RLS-LT) control. Good and Qin [20] discussed several common MIMO control laws and investigated the necessary and sufficient conditions for the stability of a MIMO EWMA R2R controller, without and with a metrology delay. In most of the previous research, the stability of the EWMA control is discussed for long production runs, and discount factors are fixed constants. Such controllers might not be practical for a process with short runs or for a model with a poorly estimated drift rate. For those cases, new control schemes are proposed, such as the initial intercept iteratively adjusted (IIIA) controller [21], the variable EWMA controller [22] and the adaptive variable controller [23].

The studies described above have one feature in common: these controllers are all designed for a single stage; this stage is optimized by considering its own dynamics, outputs and deviations. In real production scenarios, however, a stage is usually connected to preceding and subsequent stages. To improve the quality of the final product, it is important to accomplish coordination and optimization while considering all of the stages.

Joseph [24] discussed a parameter design methodology for when the process is under feedforward control, which means that the input information from a preceding stage is considered in the experimental design. However, this work is targeted to robust parameter design rather than to process control. In addition, feedback information is not considered in this work. Leang and Spanos [25] introduced a supervisory control framework and demonstrated the use of feedforward/feedback control in a photolithography process in semiconductor manufacturing. Incoming critical dimensions are first plotted on a control chart. If the chart produces a signal, which means that the incoming batch is very likely to be a defect after processing if no special measures are taken, then a feedforward controller is activated to compensate for the *known* deviation in the incoming batches and to drag the predicted output closer to the target, thereby minimizing the output variation. A feedback controller is also placed in the process and will be activated if control alarms are generated by a control chart.

There is an extensive set of research in the process control literature that combines feedforward and feedback control (see, e.g., [26,27]). However, because of the difference in the process model and the system dynamics, those algorithms cannot be used directly for an R2R process. Del Castillo [28] highlighted the uniqueness of statistical process adjustments in quality control.

Because the multistage approach is quite common in practice, it is very important to extend the control strategy to multistage processes. The propagation of variation in a multistage process has been extensively studied in the literature (see, e.g., [29–32]). Jin and Shi [30] incorporated engineering knowledge to depict the transition of quality variation in sheet metal assembly for dimension control; the authors developed a state-space modeling approach to characterize the overall body assembly variation propagation. This linear state space modeling approach was later extended to more complicated cases, such as serial-parallel multistage processes [32]. Based on such a MMP model, analysis of the process's diagnosability and the fault diagnosis of MMP have been studied by Ding et al. [33,34], Zhou et al. [35], Huang and Shi [36] and Li et al. [37]. Because of the specific engineering interests, most of the researchers focused on process modeling and fault diagnosis. Process control and monitoring based on multistage models have not yet been thoroughly studied. Tsung and Xiang [38] proposed a group-monitoring scheme for quality measurements based on a multistage model in linear state space form. However, they concentrated mainly on the development of control charts rather than control algorithms that minimize process variation. Therefore, there is still a lack of statistical control techniques for controlling multistage processes.

Wang and Han [39] discussed the use of the batch-allocation idea in a semiconductor manufacturing process; the authors studied the implementation of batch-allocation for a single stage with the considering of inputs transmitted from one upper stage. In this work, we focus on the development of a coordination and control framework for a multistage process with multiple variables. In the following, we first start from the modeling and control of a single stage, then extends the framework to a multistage scenario.

3. Batch-based R2R control strategy for a single stage

In this section, a MIMO process model for a single-stage manufacturing process is presented first; the variation in the process output is given in the form of the mean squared error (MSE) and is decomposed into within-batch variation and batch-to-batch variation. Following the decomposition, the objective functions for batch allocation and for the process controller are developed separately. The model and algorithms in this section will be extended to a multistage process in the next section.

For clarity and consistence, in this work, matrices are all written in capital bold letters, vectors in bold lower cases, and scalars in plain letters.

3.1. MIMO process model

Suppose one manufacturing station has r controllable variables and d response (quality) variables; the input (which is the output of the previous stage) has s variables. We propose to use the following equation to characterize the output of the batch-based R2R process:

$$\mathbf{y}_{ij} = \boldsymbol{\alpha} + \mathbf{A}\mathbf{x}_{ij} + \mathbf{B}\mathbf{u}_i + \boldsymbol{\varepsilon}_{ij}. \tag{1}$$

where $\mathbf{y}_{ij}$, $i=1, \ldots, n$, $j=1, \ldots, M$, is a d-dimensional vector, representing the measure of the jth workpiece in batch i after processing, and $\mathbf{x}_{ij}$ is a s-dimensional vector, representing the measure of the same workpiece before processing; $\boldsymbol{\alpha}=(\alpha_1, \ldots, \alpha_d)^T$ is a d-dimensional intercept vector, and $\mathbf{u}_i$ is a r-dimensional vector, which represents the recipe for batch i; $\mathbf{A}=(\mathbf{a}_1, \ldots, \mathbf{a}_d)^T$ is a $d \times s$ coefficient matrix that shows the impact of input quality status on output quality; $\boldsymbol{B} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_d)^T$ is a $d \times r$ matrix, which shows the impact of the control factors on the process outputs. Here, we assume that batch i, $i=1, 2, \ldots$, is processed sequentially, at time $t=i$. To emphasize the batch-based feature, we use subscript i instead of t to index each batch. The process disturbance is represented by a d-dimensional vector $\boldsymbol{\varepsilon}_{ij}$.

Eq. (1) is a natural extension of the conventional MIMO models used by Tseng et al. [11], Del Castillo and Rajagopal [12] and others; it can also be considered the simplified state-space model used in Jin and Shi [30]. According to our study, this equation is also adequate and convenient to model the wafer preparation process that is illustrated above. In addition, this regression model could be used to characterize any of the processes that have a similar linear relation between the input, the controllable variables and the output. Therefore, we treat this equation as a basic model for further study.

In an R2R controller designed for the process described in Eq. (1), the parameter estimate of $\boldsymbol{\alpha}$ should be updated from batch to batch to compensate for the potential initial bias and the batch-to-batch differences in the incoming materials. Furthermore, the incoming information for each workpiece, $\mathbf{x}_{ij}$, is assumed to be known in the model. This term could be used to represent any in priori known information that could affect the final output $\mathbf{y}_{ij}$.

To achieve a higher process capability and quality level, it is desirable to optimize the process recipes so that the mean sum of squared errors of the process output is minimized. We start the illustration using a simpler MISO system, which has a single target value τ for its output y. The MSE of n batches, each with a fixed batch size M, is defined as

$$\mathrm{MSE} = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ij}-\tau)^2 = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ij}-\mu_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(\mu_i-\tau)^2. \quad (2)$$

where μ_i is the average output value of batch i. The first term in the equation illustrates the within-batch variation, while the second term reflects the batch-to-batch variation.

When extended to a multiple-input–multiple-output (MIMO) process, a weighting matrix $\boldsymbol{\Sigma} = diag\{\omega_1^2, \cdots, \omega_d^2\}$ is required to account for the importance of the different components of $\mathbf{y}_{ij}$, i.e., $y_{ij1}, \ldots, y_{ijd}$. The weighting matrix could take any general form, while for simplicity, only a diagonal matrix is discussed in this work. Let $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_d)^T$ be the target of $\mathbf{y}_{ij}$. Correspondingly, the MSE for an MIMO system is given by

$$\mathrm{MSE} = \frac{1}{nM}\sum_{p=1}^{d}\omega_p^2\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ijp}-\mu_{ip})^2 + \frac{1}{n}\sum_{p=1}^{d}\omega_p^2\sum_{i=1}^{n}(\mu_{ip}-\tau_p)^2. \quad (3)$$

where μ_{ip} is the average of variable p of batch i. Similar to Eq. (2), the first term in Eq. (3) represents the within-batch variation, while the second term represents the batch-to-batch variation. Next, we shall seek methods for minimizing these terms.

3.2. Data clustering for batch allocation

To minimize the within-batch variation in Eq. (3), for the Pth variable of a wafer in the ith batch, it is given that

$$\mu_{ip} = \alpha_p + \mathbf{a}_p^T\bar{\mathbf{x}}_i + \mathbf{b}_p^T\mathbf{u}_i,$$

the expectation of the first term of Eq. (3) can therefore be rewritten as (apart from a constant)

$$M_1 = \frac{1}{nM}\sum_{p=1}^{d}\omega_p^2\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{a}_p^T(\mathbf{x}_{ij}-\bar{\mathbf{x}}_i))^2.$$

or equivalently,

$$M_1 = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{x}_{ij}-\bar{\mathbf{x}}_i)^T\boldsymbol{\Sigma}(\mathbf{x}_{ij}-\bar{\mathbf{x}}_i), \quad (4)$$

where $\boldsymbol{\Sigma} = diag\{q_1, \ldots, q_s\}$, $q_p = \sum_{p'=1}^{d}\omega_{p'}^2 a_{pp'}^2$, $p = 1, \ldots, s$, and s is the dimension of $\mathbf{x}_{ij}$. This function measures the distance of each input to a batch mean. That is, in order to reduce the within-batch variation in the process output, we should make the workpieces in the same batch more uniform.

To minimize Eq. (4), the conventional clustering algorithm, which minimizes the distance of each point to a cluster center, could be borrowed. In the process control scenario illustrated in this work, there are some unique constraints that are different from the usual cluster problem. For example, the cluster (batch) size is limited because of machine capacity constraints; the number of all of the clusters should be equal so that the machine utilization is maximized. The K-Means algorithm is a popular method for clustering (see [40] for a brief review). Wang and Han [39] modified the K-Means clustering algorithm to solve the univariate version of Eq. (4). This algorithm can be applied in this study to minimize the within-batch variation in a multivariate scenario. For more information about the K-Means algorithm, refer to Hastie et al. [41]. The pseudo-code of the modified K-Means algorithm for batch allocation used in this work is given in Appendix.

3.3. Run-to-run process control

The clustering strategy introduced above can be used to minimize the within-batch variation in Eq. (3); the between-batch variation will be minimized if the means of all of the clusters are close to the target. This result is expected to be achieved with the aid of an R2R controller.

In the R2R control scenario, the estimation accuracy of parameters should be considered. Let $\hat{\mathbf{A}}$ and $\hat{\mathbf{B}}$ be the estimates of the dynamic matrix $\mathbf{A}$ and control matrix $\mathbf{B}$, and $\boldsymbol{\alpha}_{i-1}$ be the estimate of $\boldsymbol{\alpha}$ obtained after batch $i-1$. It is known from Eq. (1) that the predicted batch mean of an MIMO system in run i (batch i) is given by

$$\boldsymbol{\mu}_i = \boldsymbol{\alpha}_{i-1} + \hat{\mathbf{A}}\bar{\mathbf{x}}_i + \hat{\mathbf{B}}\mathbf{u}_i, \quad i = 1, 2, \ldots, n. \quad (5)$$

To minimize the between-batch variation, the recipe should be set such that $\boldsymbol{\mu}_i = \boldsymbol{\tau}$ is satisfied. Considering that there are multiple choices of $\mathbf{u}_i$ when the number of controllable factors is larger than the number of output variables, the new recipe $\mathbf{u}_i$ can be obtained by solving the following constrained problem:

$$\begin{aligned} &\text{Min} \quad (\mathbf{u}_i-\mathbf{u}_{i-1})^T(\mathbf{u}_i-\mathbf{u}_{i-1}),\\ &\text{s.t.} \quad \boldsymbol{\tau} = \boldsymbol{\alpha}_{i-1} + \hat{\mathbf{A}}\bar{\mathbf{x}}_i + \hat{\mathbf{B}}\mathbf{u}_i. \end{aligned}$$

The intuitive meaning of this optimization problem is to bring the mean of the process output for batch i to the desired target with the smallest possible adjustments. Such a constraint is helpful to reduce the adjustment cost. Other types of adjustment cost models have been considered by Del Castillo et al. [42]. Tseng et al. [11] proved that the following equation is the solution to the above objective function:

$$\mathbf{u}_i = (\mathbf{I} - \hat{\mathbf{B}}^T(\hat{\mathbf{B}}\hat{\mathbf{B}}^T)^{-1}\hat{\mathbf{B}})\mathbf{u}_{t-1} + \hat{\mathbf{B}}^T(\hat{\mathbf{B}}\hat{\mathbf{B}}^T)^{-1}(\boldsymbol{\tau} - \boldsymbol{\alpha}_{i-1} - \hat{\mathbf{A}}\bar{\mathbf{x}}_i). \quad (6)$$

This control strategy is similar to the traditional MEWMA controller except that the mean of each batch is taken into consideration.

Because of a potential initial bias and process disturbance, it is essential to update the estimate of $\boldsymbol{\alpha}$ when the process evolves. Denote a constant λ as the discount factor for updating the model run by run. In Eq. (1), the incoming information $\mathbf{x}_{ij}$ is known; we propose to update $\boldsymbol{\alpha}$ by using the following equation:

$$\boldsymbol{\alpha}_i = \lambda(\mathbf{y}_i - \mathbf{A}\mathbf{x}_i - \mathbf{B}\mathbf{u}_i) + (1-\lambda)\boldsymbol{\alpha}_{i-1}. \quad (7)$$

In this way, the batch-allocation operation minimizes the within-batch variation, and the R2R controller minimizes the between-batch variation. The integrated use of batch allocation and R2R control is, therefore, expected to improve the quality of a single stage. In the following section, we extend this formulation to the framework shown in Fig. 3 and apply it to a multistage manufacturing process.

4. Batch-based R2R control strategy for multiple stages

In Section 3, we have presented a batch-based multivariate R2R process control strategy, which contains a clustering operation and a batch-based MEWMA controller. The above study is focused on a single stage. In this section, we develop a batch-based R2R control strategy for a multivariate multistage process. We call this strategy a Batch-MMP controller.

Consider an MIMO process with N stages (for notational simplicity, we remove the subscript i, j, p that were used in Section 3 and focus on stage differences instead)

$$\mathbf{y}_k = \boldsymbol{\alpha}_k + \mathbf{A}_k\mathbf{x}_{k-1} + \mathbf{B}_k\mathbf{u}_k + \boldsymbol{\varepsilon}_k, \quad k = 1, \ldots, N. \tag{8}$$

where k is the stage to be studied. In most cases, the output of stage k is the input of stage $k+1$, that is, $\mathbf{x}_k = \mathbf{y}_k$.

The purpose of controlling a multistage process is to improve the quality of the final output. Although a product could deviate from the target in the middle stages, as long as the deviation can be corrected by the other stages, the product will be treated as qualified in the final inspection. Therefore, we need to predict the final status of a product when it is still at an early or middle stage. Next, we expand the input–output model across all of the stages, for the analysis of the final output to be analyzed. For stage k, $k \in \{1, 2, \ldots, N\}$, we define the transition matrix as follows:

$$\mathbf{A}_{k,j} = \begin{cases} \mathbf{A}_k, & k = j, \\ \mathbf{A}_j, \ldots, \mathbf{A}_k, & k < j. \end{cases} \tag{9}$$

and the intercept vector as the following:

$$\boldsymbol{\alpha}_{k,j} = \begin{cases} \boldsymbol{\alpha}_k, & k = j, \\ \displaystyle\sum_{q=k}^{j-1} \mathbf{A}_{q+1,j}\boldsymbol{\alpha}_q + \boldsymbol{\alpha}_j, & k < j. \end{cases} \tag{10}$$

The noise vector under the condition that there is no re-clustering action after stage k is:

$$\boldsymbol{\varepsilon}_{k,j} = \begin{cases} \boldsymbol{\varepsilon}_k, & k = j, \\ \displaystyle\sum_{q=k}^{j-1} \mathbf{A}_{q+1,j}\boldsymbol{\varepsilon}_q + \boldsymbol{\varepsilon}_j, & k < j. \end{cases} \tag{11}$$

Thus, for $k = 1, \ldots, N$, the input–output model of stage $k, k+1, \ldots, N$ can be written as

$$\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_{k,k} \\ \boldsymbol{\alpha}_{k,k+1} \\ \vdots \\ \boldsymbol{\alpha}_{k,N} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{k,k} \\ \mathbf{A}_{k,k+1} \\ \vdots \\ \mathbf{A}_{k,N} \end{bmatrix} \cdot x_{k-1} + \begin{bmatrix} \mathbf{B}_k & 0 & \cdots & \mathbf{0} \\ \mathbf{A}_{k+1,k+1}\mathbf{B}_k & \mathbf{B}_{k+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{k+1,N}\mathbf{B}_k & \mathbf{A}_{k+2,N}\mathbf{B}_{k+1} & \cdots & \mathbf{B}_N \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_{k,k} \\ \boldsymbol{\varepsilon}_{k,k+1} \\ \vdots \\ \boldsymbol{\varepsilon}_{k,N} \end{bmatrix}. \tag{12}$$

Because $E(\boldsymbol{\varepsilon}_k) = \mathbf{0}$ for $k = 1, 2, \ldots, N$, we obtain the mean vectors from Eq. (12) as follows:

$$E\begin{bmatrix} \mathbf{x}_k \\ \mathbf{x}_{k+1} \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_{k,k} \\ \boldsymbol{\alpha}_{k,k+1} \\ \vdots \\ \boldsymbol{\alpha}_{k,N} \end{bmatrix} + \begin{bmatrix} \mathbf{A}_{k,k} \\ \mathbf{A}_{k,k+1} \\ \vdots \\ \mathbf{A}_{k,N} \end{bmatrix} \cdot E(\mathbf{x}_{k-1}) + \begin{bmatrix} \mathbf{B}_k & 0 & \cdots & \mathbf{0} \\ \mathbf{A}_{k+1,k+1}\mathbf{B}_k & \mathbf{B}_{k+1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{k+1,N}\mathbf{B}_k & \mathbf{A}_{k+2,N}\mathbf{B}_{k+1} & \cdots & \mathbf{B}_N \end{bmatrix} \begin{bmatrix} \mathbf{u}_k \\ \mathbf{u}_{k+1} \\ \vdots \\ \mathbf{u}_N \end{bmatrix}. \tag{13}$$

In addition, the covariance matrices are obtained as follows:

$$\begin{bmatrix} Cov(\mathbf{x}_k) \\ Cov(\mathbf{x}_{k+1}) \\ \vdots \\ Cov(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{k,k} \\ \mathbf{A}_{k,k+1} \\ \vdots \\ \mathbf{A}_{k,N} \end{bmatrix} \cdot Cov(\mathbf{x}_{k-1}) \cdot \begin{bmatrix} \mathbf{A}^T_{k,k} \\ \mathbf{A}^T_{k,k+1} \\ \vdots \\ \mathbf{A}^T_{k,N} \end{bmatrix} + \begin{bmatrix} Cov(\boldsymbol{\varepsilon}_{k,k}) \\ Cov(\boldsymbol{\varepsilon}_{k,k+1}) \\ \vdots \\ Cov(\boldsymbol{\varepsilon}_{k,N}) \end{bmatrix}. \tag{14}$$

The objective of MMP process control is to minimize the MSE of the final process output. Because of the propagation of variation across multiple stages, each stage should be well controlled. Suppose that we are now facing the control of stage k. To minimize the final MSE, we express the final output, $\mathbf{y}_N$, or equivalently, $\mathbf{x}_N$, as a function of the input to stage k, and all of the future settings of controllable variables:

$$\mathbf{x}_N = \boldsymbol{\alpha}_{k,N} + \mathbf{A}_{k,N}\mathbf{x}_{k-1} + f(\mathbf{u}_k, \ldots, \mathbf{u}_N) + \boldsymbol{\varepsilon}_{k,N}, \tag{15}$$

where $f(\mathbf{u}_k, \ldots, \mathbf{u}_N)$ is a linear combination of $\mathbf{u}_k, \ldots, \mathbf{u}_N$.

Therefore, at stage k, if all of the future control actions have been applied, then the expected output should be

$$\boldsymbol{\mu}_N = \boldsymbol{\alpha}_{k,N} + \mathbf{A}_{k,N}\boldsymbol{\mu}_{k-1} + f(\mathbf{u}_k, \ldots, \mathbf{u}_N). \tag{16}$$

If the recipe for each stage always makes the stage output on target, i.e., $\boldsymbol{\mu}_k = \boldsymbol{\tau}$, then we obtain the ideal case for Eq. (16):

$$\boldsymbol{\tau}_N = \boldsymbol{\alpha}_{k,N} + \mathbf{A}_{k,N}\boldsymbol{\tau}_{k-1} + f(\mathbf{u}_k, \ldots, \mathbf{u}_N). \tag{17}$$

Then, $\mathbf{x}_N - \boldsymbol{\tau}_N = \mathbf{A}_{k,N}(\mathbf{x}_{k-1} - \boldsymbol{\tau}_{k-1}) + \boldsymbol{\varepsilon}_{k,N}$, and thus,

$$\begin{aligned} \text{MSE} &= E((\mathbf{x}_N - \boldsymbol{\tau}_N)^T \boldsymbol{\Sigma}(\mathbf{x}_N - \boldsymbol{\tau}_N)) \\ &= E((\mathbf{x}_{k-1} - \boldsymbol{\tau}_{k-1})^T \mathbf{A}^T_{k,N} \boldsymbol{\Sigma}\mathbf{A}_{k,N}(\mathbf{x}_{k-1} - \boldsymbol{\tau}_{k-1})). \end{aligned}$$

Hence, to minimize the MSE of the final output, the input to stage k should be clustered so that the following function is minimized

$$Min \left\{ M_1 = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T\mathbf{Q}^{(k)}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i) \right\}. \tag{18}$$

where $\mathbf{Q}^{(k)} = \mathbf{A}^T_{k,N}\boldsymbol{\Sigma}\mathbf{A}_{k,N}$.

The same induction as illustrated above could be applied to all of the stages. In other words, to make the final process output uniform, the input of each stage should be clustered, such that the within-batch variation is minimized before entering a downstream stream.

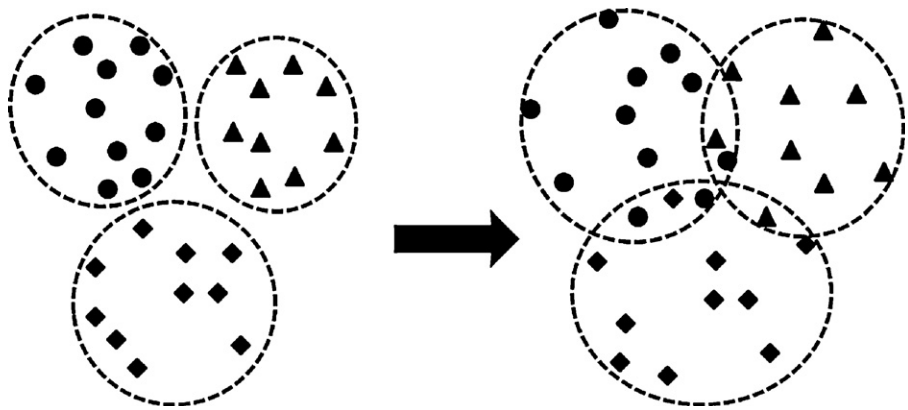

Fig. 4. Propagation of variation. Left: distribution of the points before processing; right: distribution of the points after processing.

However, Eq. (18) does not consider the re-clustering effect before stages and estimation uncertainties of $\mathbf{A}_{K,N}$. If the batch-allocation operation is carried out before each stage, it is more convenient to minimize the output of the immediate downstream stage. When each stage is controlled tightly, the variation of the final output is expected to be reduced.

To better understand the necessity for batch allocation after each processing stage, we use a two-dimensional example for illustration. As shown in Fig. 4, the distribution of points in the plane coordinate system become more diverse after processing; the enlarged circles are more overlapped. Therefore, after each stage, there is a chance to reduce the within-batch variation by re-clustering all of the points.

More precisely, the covariance matrix of $\mathbf{x}_N$ in Eq. (15) is given by

$$Cov(\mathbf{x}_N) = \mathbf{A}_{k,N} Cov(\mathbf{x}_{k-1}) \mathbf{A}_{k,N}^T + Cov(\boldsymbol{\varepsilon}_{k,N}). \tag{19}$$

Suppose that no clustering operation is applied after stage k; in other words, batch allocations remain the same after stage k. Thus, we can track the variation across stage k, $k+1$, ..., N, according to Eq. (14).

According to Eq. (11), $\boldsymbol{\varepsilon}_{k,k+1} = \mathbf{A}_{k+1}\boldsymbol{\varepsilon}_k + \boldsymbol{\varepsilon}_{k+1}$, and the second part of Eq. (19), $Cov(\boldsymbol{\varepsilon}_{k,N})$, can be written as

$$Cov(\boldsymbol{\varepsilon}_{k,N}) = \sum_{q=k}^{N-1} \mathbf{A}_{q+1,N} Cov(\boldsymbol{\varepsilon}_q) A_{q+1,N}^T + Cov(\boldsymbol{\varepsilon}_N).$$

In other words, the disturbances are accumulated and magnified stage by stage, and finally, it is applied to the final output. To reduce the accumulation effect, it is therefore necessary to first re-cluster all of the products after each stage; then, the customized recipes must be used to compensate for the deviation at each stage.

Once the batch allocation step is finished, the R2R controller shown in Eq. (6) could be implemented to generate recipes for each stage.

It should be noted that the main purpose of this work is to encourage the idea of improving quality through better coordination among the stages by using batch allocation and customized R2R control. The specific clustering algorithm and the R2R controller used in this work could be replaced by other algorithms. For example, batch-allocation could be formulated as an integer programming problem (see, e.g., [40]). Other R2R controllers that were reviewed in Section 1 of this work could be chosen on the basis of the process dynamics.

5. Performance study

In this section, we study the performance of the Batch-MMP controller by simulation. In the simulation, a two-stage manufacturing process is studied. Performances of the traditional EWMA controller and the Batch-MMP controller are compared. Because process disturbances are usually described by a white noise and IMA time series in the literature [1,7], both types of disturbances are considered here.

5.1. Performance study under white noise disturbance

Without loss of generality, we study here a process that has two stages, i.e., $N=2$. A two-stage MIMO process model can be described by

$$\begin{cases} \mathbf{x}_1 = \boldsymbol{\alpha}_1 + \mathbf{A}_1\mathbf{x}_0 + \mathbf{B}_1\mathbf{u}_1 + \boldsymbol{\varepsilon}_1, \\ \mathbf{x}_2 = \boldsymbol{\alpha}_2 + \mathbf{A}_2\mathbf{x}_1 + \mathbf{B}_2\mathbf{u}_2 + \boldsymbol{\varepsilon}_2. \end{cases}$$

The final output is given by

$$\mathbf{x}_2 = \boldsymbol{\alpha}_{1,2} + \mathbf{A}_{1,2}\mathbf{x}_0 + \mathbf{A}_2\mathbf{B}_1\mathbf{u}_1 + \mathbf{B}_2\mathbf{u}_2 + \boldsymbol{\varepsilon}_{1,2}, \tag{20}$$

where $\boldsymbol{\alpha}_{1,2} = \boldsymbol{\alpha}_2 + \mathbf{A}_2\boldsymbol{\alpha}_1$, $\mathbf{A}_{1,2} = \mathbf{A}_2\mathbf{A}_1$, and $\boldsymbol{\varepsilon}_{1,2} = \mathbf{A}_2\boldsymbol{\varepsilon}_1 + \boldsymbol{\varepsilon}_2$.

Consider a specific case with the following arbitrary settings:

$$\begin{cases} \mathbf{x}_1 = \begin{bmatrix} 5 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}\mathbf{x}_0 + \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix}\mathbf{u}_1 + \boldsymbol{\varepsilon}_1, \\ \mathbf{x}_2 = \begin{bmatrix} 10 \\ 8 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix}\mathbf{x}_1 + \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}\mathbf{u}_2 + \boldsymbol{\varepsilon}_2. \end{cases} \tag{21}$$

where $\varepsilon_1, \varepsilon_2 \sim N_2(\mathbf{0}, \mathbf{I}_2)$. The target of the two stages is set as

$$\boldsymbol{\tau}_1 = \begin{bmatrix} 15 \\ 9 \end{bmatrix}, \quad \boldsymbol{\tau}_2 = \begin{bmatrix} 7 \\ 5 \end{bmatrix}.$$

The initial input $\mathbf{x}_0$ satisfies

$$\mathbf{x}_0 \sim \mathbf{N}_2(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0), \quad \boldsymbol{\mu}_0 = \begin{bmatrix} 20 \\ 10 \end{bmatrix}, \quad \boldsymbol{\Sigma}_0 = \begin{bmatrix} 9 & 0 \\ 0 & 4 \end{bmatrix}. \tag{22}$$

The weight matrix is set as

$$\boldsymbol{\Sigma} = \begin{bmatrix} 3 & 0 \\ 0 & 4 \end{bmatrix}.$$

The evaluation of the control performance is then conducted by following the procedures (repeated for 100 times):

(1) Estimate the process parameters from historical data. The parameters to be estimated include $\boldsymbol{\alpha}_1$, $\mathbf{A}_1$, $\mathbf{B}_1$, $\boldsymbol{\alpha}_2$, $\mathbf{A}_2$, and $\mathbf{B}_2$;

Table 1
The MSEs of the outputs for all of the stages.

Controller	Stage	Target	Mean of all 36,000 observations	Covariance of all 36,000 observations	Average MSE
EWMA controller	Initial input	(20,10)	(20.01,9.98)	$\begin{bmatrix} 8.96 & 0.01 \\ 0.01 & 3.98 \end{bmatrix}$	36.70
	Stage 1	(15,9)	(14.17,8.45)	$\begin{bmatrix} 39.59 & 8.36 \\ 8.36 & 15.85 \end{bmatrix}$	278.56
	Stage 2	(7,5)	(0.42,1.62)	$\begin{bmatrix} 90.17 & 55.62 \\ 55.62 & 57.24 \end{bmatrix}$	384.44
Batch-MMP controller	Initial input	(20,10)	(19.99,9.98)	$\begin{bmatrix} 9.03 & 0.01 \\ 0.01 & 3.98 \end{bmatrix}$	43.75
	Stage 1	(15,9)	(14.17,8.45)	$\begin{bmatrix} 8.19 & -3.03 \\ -3.03 & 5.76 \end{bmatrix}$	54.31
	Stage 2	(7,5)	(7.63,5.50)	$\begin{bmatrix} 9.61 & -2.00 \\ -2.00 & 6.81 \end{bmatrix}$	68.54

use the means of the historical control actions to set the initial $\mathbf{u}_1, \mathbf{u}_2$.
(2) Generate 360 “workpieces” according to the distribution given in (22).
(3) Stage 1—cluster those 360 workpieces into batches by minimizing Eq. (18); the size of each batch is set to 30.
(4) Stage 1—process the workpieces batch by batch; in this process, determine the control actions for each coming batch, based on the batch mean, and update the intercept vectors $\boldsymbol{\alpha}_1$ after each batch.
(5) Stage 2—re-cluster the 360 workpieces that are processed by stage 1 into 20 batches.
(6) Stage 2—process the workpieces batch by batch; in this process, determine the control actions for each coming batch based on the batch mean, and update the intercept vectors $\boldsymbol{\alpha}_2$ after each batch.

In the above simulation study, both the Batch-MMP controller and the traditional single-stage EWMA controller were implemented to make a performance comparison. The Batch-MMP controller employs the batch-allocation algorithm to minimize within-batch variation, while the conventional EWMA controller groups workpieces into clusters randomly. The simulation run is repeated for 100 times; the average output and MSE are calculated for each stage. The results are shown in Table 1. It is evident that, with the EWMA controller, the MSE increased from 36.70 to 278.56 after stage 1 and increased to 384.44 after stage 2. In other words, the variation of the process is magnified stage by stage. Instead, if the process is controlled by the Batch-MMP controller, then the variation is maintained at a low level. As a result, compared to the EWMA controller, the Batch-MMP controller can reduce the process variation and can improve the quality greatly.

Fig. 5 shows the MSEs for all of the 100 replicates. The solid lines refer to the MSEs of the EWMA controller, while the dashed lines refer to those of the Batch-MMP controller. The figure shows clearly that the two dashed lines fall below the solid lines. Thus, we can arrive at the conclusion that the Batch-MMP controller performs well when reducing the output MSEs for an MMP.

5.2. Performance study under an IMA disturbance series

In the case above, the process disturbances are assumed to be white noise, which are given as $\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2 \sim N_2(\mathbf{0}, \mathbf{I}_2)$. In some cases, an IMA series could better illustrate a non-stationary process disturbance [7,43]. Therefore, in the following, we study the performance of the controller when both stages suffer from an IMA disturbance series. In such a situation, Eq. (21) is rewritten as follows:

$$\begin{cases} \mathbf{x}_1 = \begin{bmatrix} 5 \\ 2 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix} \mathbf{x}_0 + \begin{bmatrix} -2 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{u}_1 + \mathbf{d}_1, \\ \mathbf{x}_2 = \begin{bmatrix} 10 \\ 8 \end{bmatrix} + \begin{bmatrix} 2 & 1 \\ 1 & 3 \end{bmatrix} \mathbf{x}_1 + \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix} \mathbf{u}_2 + \mathbf{d}_2. \end{cases}$$

where both $\mathbf{d}_1$ and $\mathbf{d}_2$ take the form of

$$\mathbf{d}_t = \mathbf{d}_{t-1} + \boldsymbol{\varepsilon}_t - \boldsymbol{\theta}^T \boldsymbol{\varepsilon}_{t-1},$$

where ε_1, $\varepsilon_2 \sim N_2(\mathbf{0}, \mathbf{I}_2)$, and t is the index of “run”. Here, we set $\boldsymbol{\theta} = (0.6, 0.6)^T$ for all of the stages.

Following the same simulation procedure that was described earlier with 100 replicates, we obtain the results and show them

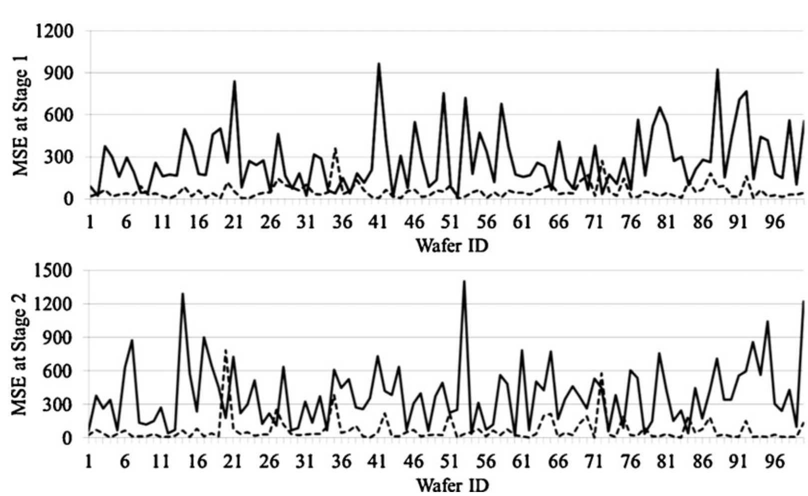

Fig. 5. The MSEs for all 100 replicates.

Table 2
The MSEs of the outputs for all of the stages.

Controller	Stage	Target	Mean of all 36,000 observations	Covariance of all 36,000 observations	Average MSE
EWMA controller	Initial input	(20,10)	(19.99,9.99)	$\begin{bmatrix} 9.01 & 0.05 \\ 0.05 & 4.02 \end{bmatrix}$	38.39
	Stage 1	(15,9)	(14.17,8.45)	$\begin{bmatrix} 40.02 & 8.63 \\ 8.63 & 16.04 \end{bmatrix}$	216.52
	Stage 2	(7,5)	(0.42,1.62)	$\begin{bmatrix} 90.53 & 56.14 \\ 56.14 & 57.97 \end{bmatrix}$	340.82
Batch-MMP controller	Initial input	(20,10)	(19.99,9.98)	$\begin{bmatrix} 9.01 & -0.03 \\ -0.03 & 4.02 \end{bmatrix}$	40.76
	Stage 1	(15,9)	(14.17,8.45)	$\begin{bmatrix} 8.29 & -3.26 \\ -3.26 & 5.86 \end{bmatrix}$	46.24
	Stage 2	(7,5)	(7.63,5.50)	$\begin{bmatrix} 9.43 & -2.21 \\ -2.21 & 6.98 \end{bmatrix}$	46.69

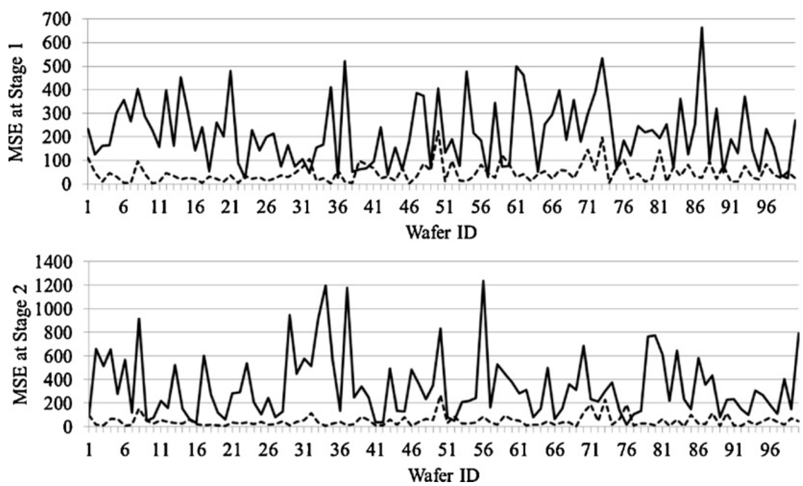

Fig. 6. The MSEs for all 100 replications.

in Table 2 and Fig. 6. For the traditional EWMA controller, the MSE is found to be enlarged stage by stage. However, the Batch-MMP successfully maintains the MSE on a relatively stable level. The final MSE achieved by the Batch-MMP controller is 46.69, and only 14% of the MSE caused by the EWMA controller.

Tables 1 and 2 show that if the process is controlled by the EWMA controller without considering the coordination of multiple stages, the resulting MSE increases quickly from the initial input to stage 1 and stage 2. The proposed MMP controller instead shows a much smaller increasing trend in the MSE. The reason is that the propagation of variation across the stages is corrected at each stage. Fig. 7 further shows a simulated process with 60 two-dimensional samples, and the size of each batch is 10. Using the same parameters and disturbance models as those studied above (refer to

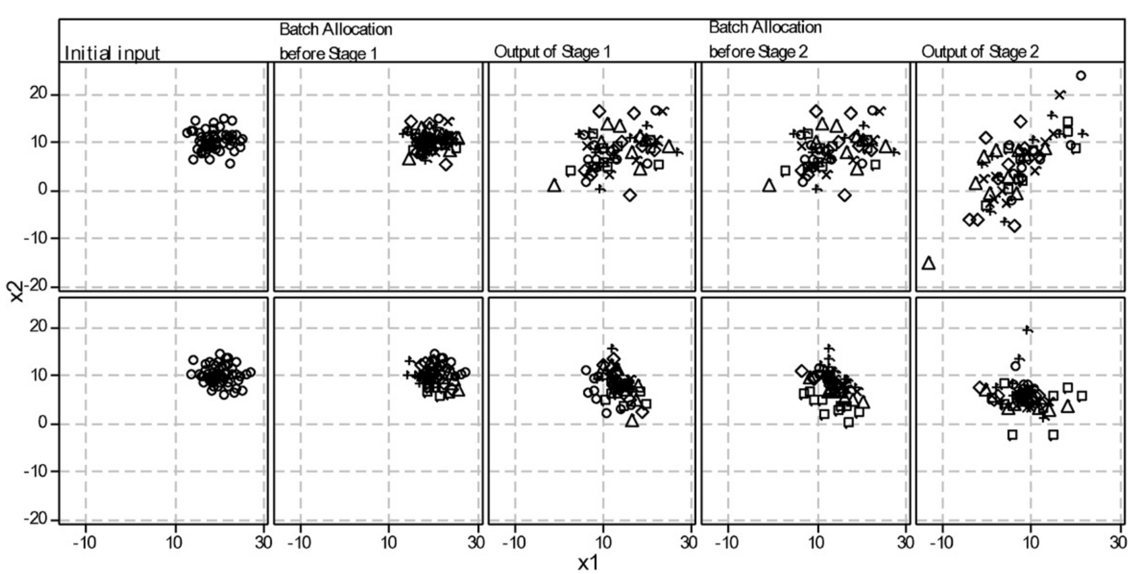

Fig. 7. The locations of the workpieces in each stage.

Tables 1 and 2), the graphs show the distribution of the samples at each stage.

In Fig. 7, the first row shows the quality states of the workpieces when the traditional EWMA controller is applied, and the second row shows those when the Batch-MMP controller is used. As shown in the figure, for the EWMA controller, because no batch allocation is applied, all of the clusters are formed randomly and overlap with each other. All of the clusters, therefore, share the same recipe. Because of the influence of the process noise, the distribution of the points is enlarged stage by stage.

For the Batch-MMP controller, corrective actions are taken after each stage. For example, stage 1 increases the variation of the points. However, before entering stage 2, the points are re-clustered; the recipes of stage 2 are then customized for each batch. Even if one cluster could deviate far from the target, the customized recipe would bring it back to the target. Therefore, the variation after stage 2 is still relatively small. Thus, the coordination of multiple stages is helpful for improving the reduction in variation and for improving the quality.

6. Conclusions

In this paper, we proposed a Batch-MMP controller for MIMO multistage manufacturing processes. In batch-based semiconductor manufacturing processes, the variation of the final output can be decomposed into within-batch variation and batch-to-batch variation. A framework that consists of a Batch-Allocator and an R2R controller is then proposed. In this framework, the Batch-Allocation is designed to make input batches have small differences within the batches and have large differences between the batches; the R2R controller is designed to generate customized recipes to compensate for process disturbances and to move all of the batches to the same target.

The essence of the new proposal is to obtain better coordination among the stages of a complex manufacturing system. Without effective coordination, a disturbance at each stage propagates to downstream stages and accumulates until the end of the process. The batch-allocation operation breaks the accumulation; the R2R controller eliminates accumulated deviations and brings all of the batches back to the target. Simulation results show that the Batch-MMP effectively reduces the variation.

In this study, we employed the modified K-Means algorithm for clustering and the EWMA controller for feedback control. For other types of data structures and process dynamics, a different clustering method or controller could be considered.

It should be noted that in a multistage process, estimation of parameters for the model in Eq. (1) is critical. Inaccurate estimates may affect the controller's stability and robustness. Therefore, modeling building and parameter estimation for such processes are important topics for future research.

To the practitioners, the batch-allocation operation may be time-consuming. However, if the saving of quality loss is profound, the implementation of this operation should be pursued. In addition, the widely used high-speed product sorting equipment may be one way to an automatic implementation of this operation in practice.

In a multistage process, the proper use of tolerance is critical for preventing defect propagation and for reducing the overall cost. When the Batch-MMP controller is applied, it inevitably affects the tolerance of the middle stages. It is possible that products that were originally classified as no-go defects using the old standard are corrected by downstream stages and still satisfy the final quality requirements. Therefore, the tolerance synthesis for a multistage manufacturing process that is equipped with the Batch-MMP controller is also an important topic that deserves future research efforts.

Acknowledgements

The authors would like to thank the editor and the referees for their valuable comments and suggestions, which have helped improve this work greatly. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 71072012 and Tsinghua University Initiative Scientific Research Program.

Appendix. The modified K-Means algorithm for batch allocation

(1) The algorithm starts with a user specified value of k. Takes the first k points in the sample as the initial means and form k groups. Also, one parameter should be specified: the capacity of each group M. Usually we assume the sample has a size of $N \times M$. Thus, we will finally obtain N full groups.

(2) **Initial assignment**. Assign each point to its nearest group which is not full. After each new point is added, calculate the average within-group variation (AWGV), which is defined as follows:

$$\text{AWGV} = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^T \boldsymbol{\Sigma}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i),$$

where $\mathbf{x}_{ij}$ is the quality vector of the jth workpiece in group i, and $\boldsymbol{\Sigma}$ is weight matrix measuring respective weights of components of the quality vector.

(3) **Optimization**. Choose one point $\mathbf{x}_{ij}(1 \le i \le n, 1 \le j \le M)$. Try to exchange this point with another one $\mathbf{x}_{i_1 j}(i_1 \neq i, \quad 1 \le j \le M)$, i.e., assign $\mathbf{x}_{ij}$ this point to Group i_1 and assign $\mathbf{x}_{i_1 j}$ the other point to Group i. Calculate the reduction of the AWGV. Enumerate all points $\mathbf{x}_{i_1 j}(i_1 \neq i, \quad 1 \le j \le M)$, determine the greatest reduction of the AWGV for $\mathbf{x}_{ij}$ and exchange the group assignments.

(4) Step (3) is iterated until all $\mathbf{x}_{ij}$, $i = 1, \ldots, n$, $j = 1, \ldots, M$, are visited.

References

[1] Del Castillo E, Hurwitz AM. Run-to-run process control: literature review and extensions. Journal of Quality Technology 1997;29(2):184–96.
[2] Sachs E, Guo RS, Ha S, Hu A. Process control system for VLSI fabrication. IEEE Transactions on Semiconductor Manufacturing 1991;4(2):134–44.
[3] Shi J. Stream of variation modeling and analysis for multistage manufacturing processes. CRC; 2006.
[4] Desrochers A. A comparison of adaptive control strategies for hot steel strip rolling mills. Journal of Manufacturing Systems 1982;1(2):183–94.
[5] Ingolfsson A, Sachs E. Stability and sensitivity of an EWMA controller. Journal of Quality Technology 1993;25(4):271–87.
[6] Butler SW, Stefani JA. Supervisory run-to-run control of polysilicon gate etch using in-situ ellipsometry. IEEE Transactions on Semiconductor Manufacturing 1994;7(2):193–201.
[7] Wang K, Tsung F. Run-to-run process adjustment using categorical observations. Journal of Quality Technology 2007;39(4):312–25.
[8] Shang Y, Wang K, Tsung F. An improved run-to-run process control scheme for categorical observations with misclassification errors. Quality and Reliability Engineering International 2009;25:397–407.
[9] Wang K, Tsung F. Recursive parameter estimation for categorical process control. International Journal of Production Research 2010;48(5):1381–94.
[10] Lin J, Wang K. A Bayesian framework for online parameters estimation and process adjustment using categorical observations. IIE Transactions 2012;44:291–300.

[11] Tseng ST, Chou RJ, Lee SP. A study on a multivariate EWMA controller. IIE Transactions 2002;34(6):541–9.
[12] Del Castillo E, Rajagopal R. A multivariate double EWMA process adjustment scheme for drifting processes. IIE Transactions 2002;34(12):1055–68.
[13] Fan SKS, Wang CY. On-line tuning system of multivariate DEWMA control based on a neural network approach. International Journal of Production Research 2008;46(13):3459–84.
[14] Su CT, Hsu CC. On-line tuning of a single EWMA controller based on the neural technique. International Journal of Production Research 2004;42(11):2163–78.
[15] Jen CH, Jiang BC, Fan SKS. General run-to-run (R2R) control framework using self-tuning control for multiple-input multiple-output (MIMO) processes. International Journal of Production Research 2004;42(20):4249–70.
[16] He F, Wang K, Jiang W. A general harmonic rule controller for run-to-run process control. IEEE Transactions on Semiconductor Manufacturing 2009;22(2):232–44.
[17] Campbell W, Firth S, Toprac A, Edgar T. A comparison of run-to-run control algorithms. Proceedings of the American Control Conference 2002;3:2150–5.
[18] Wang J, He QP, Qin SJ, Bode CA, Purdy MA. Recursive least squares estimation for run-to-run control with metrology delay and its application to STI etch process. IEEE Transactions on Semiconductor Manufacturing 2005;18(2):309–19.
[19] Chen L, Ma M, Jang SS, Wang DSH, Wang S. Performance assessment of run-to-run control in semiconductor manufacturing based on IMC framework. International Journal of Production Research 2009;47(15):4173–99.
[20] Good RP, Qin SJ. On the stability of MIMO EWMA run-to-run controllers with metrology delay. IEEE Transactions on Semiconductor Manufacturing 2006;19(1):78–86.
[21] Tseng ST, Song W, Chang YC. An initial intercept iteratively adjusted (IIA) controller: an enhanced double EWMA feedback control scheme. IEEE Transactions on Semiconductor Manufacturing 2005;18(3):448–57.
[22] Tseng ST, Tsung F, Liu PY. Variable EWMA run-to-run controller for drifted processes. IIE Transactions 2007;39(3):291–301.
[23] Tseng ST, Jou BY, Liao CH. Adaptive variable EWMA controller for drifted processes. IIE Transactions 2010;42(4):247–59.
[24] Joseph VR. Robust parameter design with feed-forward control. Technometrics 2003;45(4):284–92.
[25] Leang S, Spanos CJ. Application of feed-forward and feedback control to a photolithography sequence; 1992. p. 143–7.
[26] Jutan A, Uppal A. Combined feedforward–feedback servo control scheme for an exothermic batch reactor. Industrial & Engineering Chemistry Process Design and Development 1984;23(3):597–602.
[27] Pao LY, Butterworth JA, Abramovitch DY. Combined feedforward/feedback control of atomic force microscopes; 2007. p. 3509–15.
[28] Del Castillo E. Statistical process adjustment: a brief retrospective, current status, and some opportunities for further work. Statistica Neerlandica 2006;60(3):309–26.
[29] Mantripragada R, Whitney D. Modeling and controlling variation propagation in mechanical assemblies using state transition models. IEEE Transactions on Robotics and Automation 1999;15(1).
[30] Jin J, Shi J. State space modeling of sheet metal assembly for dimensional control. Journal of Manufacturing Science and Engineering 1999;121:756.
[31] Ding Y, Ceglarek D, Shi J. Modeling and diagnosis of multistage manufacturing processes. Part I: State space model. In: Japan/USA Symposium on Flexible Automation. 2000. p. 23–6.
[32] Huang Q, Shi J. Stream of variation modeling and analysis of multistage systems serial-parallel manufacturing. Journal of Manufacturing Science and Engineering: Transactions of the ASME 2004;126(3):611–8.
[33] Ding Y, Ceglarek D, Shi J. Fault diagnosis of multistage manufacturing processes by using state space approach. Journal of Manufacturing Science and Engineering: Transactions of the ASME 2002;124(2):313–22.
[34] Ding Y, Shi J, Ceglarek D. Diagnosability analysis of multi-station manufacturing processes. Journal of Dynamic Systems, Measurement, and Control 2002;124(1):1–13.
[35] Zhou SY, Ding Y, Chen Y, Shi JJ. Diagnosability study of multistage manufacturing processes based on linear mixed-effects models. Technometrics 2003;45(4):312–25.
[36] Huang Q, Shi JJ. Variation transmission analysis and diagnosis of multi-operational machining processes. IIE Transactions 2004;36(9):807–15.
[37] Li J, Jin J, Shi J. Causation-based t2 decomposition for multivariate process monitoring and diagnosis. Journal of Quality Technology 2008;40(1):46–58.
[38] Tsung F, Xiang L. Multistage process control and monitoring via group EWMA schemes. In: Proceedings of the 2004 IEEE International Conference on Networking, Sensing & Control. 2004. p. 1087–92.
[39] Wang K, Han K. A batch-based run-to-run process control scheme for semiconductor manufacturing. IIE Transactions, http://dx.doi.org/10.1080/0740817X.2012.757681. in press.
[40] Sağlam B, Salman FS, Sayın S, Türkay M. A mixed-integer programming approach to the clustering problem with an application in customer segmentation. European Journal of Operational Research 2006;173(3):866–79.
[41] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. New York: Springer; 2001.
[42] Del Castillo E, Pan R, Colosimo BM. A unifying view of some process adjustment methods. Journal of Quality Technology 2003;35(3):286–93.
[43] Tseng ST, Yeh AB, Tsung F, Chan YY. A study of variable EWMA controller. IEEE Transactions on Semiconductor Manufacturing 2003;16(4):633–43.