IIE Transactions (2013) **45**, 658–669

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/0740817X.2012.757681

A batch-based run-to-run process control scheme for semiconductor manufacturing

KAIBO WANG* and KAI HAN

Department of Industrial Engineering, Tsinghua University, Beijing 100084, People's Republic of China
E-mail: kbwang@tsinghua.edu.cn

Received November 2011 and accepted September 2012

Run-to-Run (R2R) control has been widely used in semiconductor manufacturing to compensate for process disturbance and to improve quality. The traditional R2R controller only takes the process output in a previous run as its input and generates an optimal recipe for the next run. In a multistage semiconductor manufacturing process, variations in upstream stations are propagated to downstream stations. However, the information from upstream stations is not considered by existing controllers. In addition, most R2R processes have a limited capacity; the products must be processed in batches. Therefore, if the incoming materials could be grouped with small within-batch variations and large batch-to-batch variations and the recipes are customized for each batch to drive all batch averages toward the same target value, the output variation could be reduced and quality improved. A batch Exponentially Weighted Moving Average (EWMA) controller is proposed in this article. It employs a modified *K*-means algorithm to group the incoming materials into batches with a fixed and equal size while minimizing the within-batch variation. The controller then generates the control settings by taking both the batch information and the feedback quality information into account. Simulation studies show that the proposed controller could significantly reduce output variation and improve quality.

Keywords: Batch-to-batch variation, *K*-means clustering algorithm, run-to-run process control, semiconductor manufacturing, within-batch variation

1. Introduction

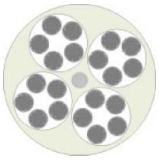
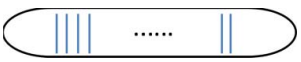
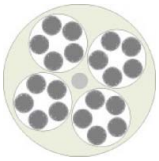
In semiconductor manufacturing processes, products are typically processed in the form of separated runs (batches). A run is defined as a series of operations on one workpiece or a batch of workpieces within an inseparable time interval. During processing, the products in the same batch undergo identical treatments and the treatments applied to products in different batches can be adjusted to compensate for process variations such as tool wear-out and dimensional changes. To reduce failures caused by potential process drifts or shifts between different batches, research on Run-to-Run (R2R) process control has attracted extensive attention in recent years (see, for example, Sachs *et al.* (1991), Sachs *et al.* (1995), Del Castillo and Hurwitz (1997), and Tseng *et al.* (2003)). Such control algorithms, typically called R2R controllers, update process model parameters in a recursive way when output from the previous run becomes available and help generate recipes for future runs.

A typical semiconductor manufacturing process has the following two features: first, multiple stages are required to produce a single product; second, the handling capacity of each stage is limited, thus forming *production batches*. In this research, we use a wafer preparation process as an example for illustration. More than 10 stages are needed to produce a wafer from a crystal silicon ingot. The inputs to five major stages (slicing, lapping, etching, Chemical Vapor Deposition (CVD), and polishing) are shown in Table 1. In the slicing stage, one, two, or four ingots can be sliced at the same time using wire saws; hundreds or thousands of wafers can be produced in one run. Sliced wafers are then arranged into smaller batches and mounted on a lapping plate to remove slicing cracks. The wafer quality is thereby improved. For example, thickness uniformity is improved after the lapping stage. After that, the wafers are put into holding baskets and etched. Then, wafers are removed from the baskets, reassembled, and placed into a quartz boat for CVD. Finally, the wafers are polished after being separated into batches according to the capacity of the polishing machine.

It is clear from the above illustration that *production batch* is a basic and important feature in wafer production. The batch size varies from stage to stage due to changes in the machine capacity. In a process we studied in a local factory, the slicing process generates batches of 300 to 400 wafers via simultaneous wire-saw cutting. Each lapping batch can

*Corresponding author

Table 1. Inputs for five major processing stages in wafer preparation

Input (schematic illustration)	*Input*	*Processing stage*
	Silicon ingot	Slicing
	Lapping plate with rings and slots; fixed capacity	Lapping
	Holding basket with fixed capacity	Etching
	Quartz boat with fixed capacity	CVD
	Polishing pad with fixed capacity	Polishing

only handle between 30 and 60 wafers, depending on the size of the wafers. In the final polishing stage, the batch size increases because the polishing plate is larger than the lapping plate. Therefore, in such a typical wafer fabrication process, there is a need to group wafers into batches before processing.

The main purpose of process control is to reduce output variation. Variation is also closely related to process yield and productivity. As Montgomery (2005, p. 4) pointed out, "quality is inversely proportional to variability." That is, the larger the variation is, the poorer the quality becomes. From the commonly used definition of the quality loss function, $\Sigma(y_t - \tau)^2$, where τ is the target of the process output and y_t is the measure of an individual output, we see the necessity to minimize output variability. Large variability in the process output may lead to a higher chance of producing nonconforming products, hence reducing yield and harming productivity. As ITRS (2009) reported, the yield and product maturity assumptions that are commonly used in some semiconductor manufacturing sections (microprocessor unit, DRAM, flash) is around 75–85%. Among others, random and systematic mechanisms are the major sources that limit the yield. A better understanding of the production system and a better coordination among incoming materials, machines, manpower, and procedures are expected to reduce variation, improve quality, and hence boost yield and productivity.

In semiconductor manufacturing, R2R process control is one main way to stabilize process output, reduce variation, and improve quality (see, for example, Del Castillo and Hurwitz (1997), Tsung and Shi (1999), Jin and Tsung (2009), and Lin and Wang (2012)). Most traditional R2R controllers are designed to take real-time or delayed process output information and continuously generate updated recipes to compensate for process shifts or drifts. Let y_t be the process output of run t. Then, a commonly assumed R2R process model is given by (see the review work by Del Castillo and Hurwitz (1997) and references therein):

$$y_t = \alpha + \beta u_t + d_t, \tag{1}$$

where α and β are the model intercept and slope, respectively; u_t represents the control settings (recipe) applied to run t; and d_t denotes the process disturbance. To control such a process, the Exponentially Weighted Moving Average (EWMA) R2R controller proposed by Sachs *et al.* (1995) is one of the most fundamental and popular algorithms. Suppose the initial estimates of parameters α and β in Equation (1) are a_0 and b, respectively. The EWMA controller updates the estimate of α repeatedly at the end of each run when a new output becomes available in the following way:

$$\alpha_t = \omega\,(y_t - bu_t) + (1 - \omega)\,a_{t-1},$$

and the control action for run t is set to $u_{t+1} = (\tau - a_t)/b$ so that the expected output is on target. The smoothing parameter ω here can be tuned to respond to fast or slow process dynamics. If the process disturbance d_t is a first-order integrated moving averages (IMA) time series, $d_i = d_{i-1} + \varepsilon_i - \theta\varepsilon_{i-1}$, where θ is the moving average coefficient, and the controller uses $\omega = 1 - \theta$, the EWMA controller is the optimal one to compensate for the disturbance series.

Following the initial work of Sachs *et al.* (1995), extensive studies have been carried out that have considered different disturbance models in the target process, such as the white noise model, the AutoRegressive Moving Average (ARMA) time series model, and the deterministic drifts model. Target to each particular type of assumed disturbance model, different controllers have been proposed. For example, the double-EWMA controller (Butler and Stefani, 1994) and the variable EWMA controller (Tseng *et al.*, 2007) can be used to compensate deterministic drift; the controller (Tsung and Shi, 1999) can compensate proportional–integral–derivative for the ARMA disturbance series. The self-tuning controller (Del Castillo and Hurwitz, 1997) is capable of recursively updating the estimates of all unknown parameters. The general harmonic rule controller (He *et al.*, 2009) is designed for a wide range of disturbances. The categorical controller (Wang and Tsung, 2007, 2010; Shang *et al.*, 2009; Lin and Wang, 2011) is designed to operate in the scenario when continuous observations are difficult to collect. In addition, the multivariate version of the EWMA and double-EWMA controller was introduced in Ingolfsson and Sachs

Table 2. Comparisons of R2R controllers

Controller	*Main features*
EWMA	Optimal for processes with an IMA(1,1) time series if properly tuned
Double-EWMA	Consideration of IMA(1,1) disturbance and a deterministic drift
Self-tuning	Separation of parameter estimation and process control; recursive least squares is used to estimate all unknown parameters
PID	Designed for ARMA(1,1) disturbance series
General harmonic rule	Designed for IMA(1,1), ARMA(1,1), or ARIMA(1,1) disturbance; robust to parameter estimation uncertainty
Categorical	Useful when categorical instead of continuous observations are available
Dead-band	Consideration of adjustment cost

(1993) and Tseng *et al.* (2002). The dead-band control strategy considers the adjustment cost (Lian *et al.*, 2006). Del Castillo *et al.* (2003) provided a unified view of a wide collection of controllers and formulated the R2R control problem in a linear quadratic Gaussian framework. Such a formulation can be generalized to handle more complicated process and disturbance models. The main features of these controllers are summarized in Table 2 for comparison.

However, all of these controllers work in a way that only process output deviations are taken into account to update model estimates and generate recipes; the coordination between upstream and downstream stages are not considered.

In a multistage production process, one major source of variation is the incoming material. That is, the output of an upstream process inevitably contains variation; such variations naturally propagate to downstream processes and are most likely enlarged. However, the above R2R controllers only use the feedback information to generate recipes; feedforward information from preceding stages, although important and also available in semiconductor manufacturing, is not considered by these controllers.

To depict the features of batch-based production more clearly, we discuss the lapping stage in this section in greater detail. Figure 1 shows a schematic plot of the process. Each lapping plate is equipped with rotating rings, and each ring has several wafer slots to hold the wafers for processing. Therefore, wafers that arrive at the lapping stage are first grouped into batches with a predetermined size. It is known that a lapping process often suffers from disturbances such as plate wear-out, slow temperature drift, and slow slurry changes. To improve the output quality (e.g., increase the thickness uniformity), recipes (which include settings such as the rotation speed, upper plate pressure, and lapping time) of the lapping process are constantly updated by an R2R controller. The wafer quality is usually characterized by parameters including thickness, total thickness variation, and total indicator reading, among others (Lin and Wang, 2011). If all of the batches have the same or similar quality, the traditional R2R control strategy, which only considers process disturbances and ignores the differences between incoming batches, is sufficient for such a process. An appropriate R2R controller can help reduce quality variation due to process drift or shift. Nonetheless, if the key measures of incoming batches differ significantly, ignoring such information may lead to serious quality deterioration.

As a simple illustration, suppose that wafer thickness is a highly important parameter for quality control and that lapping time is used to control the amount to be removed from the wafer to produce a consistent thickness. A traditional R2R controller would assume that the input thickness values of all batches are equal and calculate the lapping time based on the output thickness. While it is

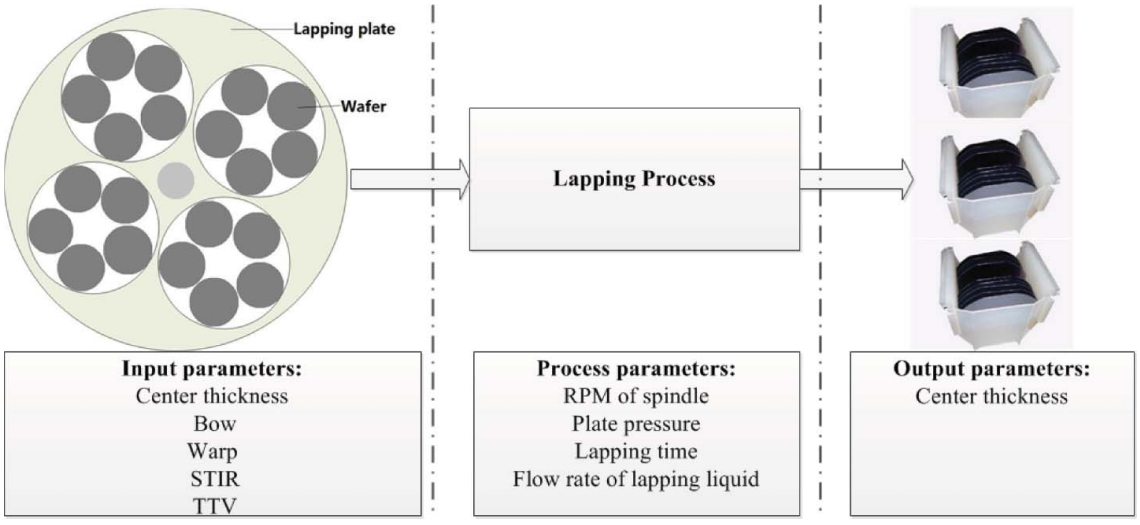


Fig. 1. Parameters and flow of the lapping stage (color figure provided online).

easy to understand that for some batches that are thicker or thinner than a normal batch, longer or shorter lapping times should be employed. Such incoming thickness information, together with the real output thickness information from the previous run, should be considered by the R2R controller when updating its recipes. In addition, the same process settings are applied to all samples in the same batch, and different settings could be applied to different batches. Therefore, we should try to make wafers in the same batch more similar but allow difference between batches; such differences should also be used to guide the generation of recipes for new runs.

Therefore, in this work, we aim to develop a new R2R control strategy in the following manner. To address the capacity constraint, a batch allocation algorithm is first designed to minimize variation within the batch by putting *similar* wafers into the same batch; an improved R2R feedback control algorithm is then incorporated to produce recipes that consider both feedback and incoming information.

The remainder of the article is organized as follows: Section 2 discusses process models and proposes a new control algorithm called batch-EWMA. Section 3 introduces a fixed-capacity K-means clustering algorithm to minimize batch variation. The performance of the newly proposed batch-EWMA controller is studied in Section 4. Finally, Section 5 concludes this work with topics for future research.

2. Process modeling and the batch-EWMA controller

Similar to the model in Equation (1), here we study a batch-based process and use the following equation to characterize the output of the process when both input and recipe information are considered:

$$y_{ij} = \alpha + \mathbf{a}^{\mathrm{T}}\mathbf{x}_{ij} + \mathbf{b}^{\mathrm{T}}\mathbf{u}_i + d_i, \tag{2}$$

where $y_{ij}, i = 1, \ldots, n, j = 1, \ldots, M$, is the measure of the jth product in batch i after processing; $\mathbf{x}_{ij}$ is the measure of the same product before processing; α is the intercept; and $\mathbf{u}_i$ is the recipe for batch i. 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 d_i is assumed to be a white noise or an IMA time series.

In the traditional EWMA control algorithm, the intercept parameter α is used to represent potential faults such as initial setup bias, parameter estimation bias, or process mean shift or drift. Therefore, this parameter is continuously updated after each production run based on a known process output. In Equation (2), large wafer differences are reflected in the variation of $\mathbf{x}_{ij}$. Therefore, recipe $\mathbf{u}_i$ could be optimized to compensate for changes in incoming materials if $\mathbf{x}_{ij}$ is known. In the lapping stage, such information could be obtained from an inspection machine.

In should be noted that the interactions among controllable factors and incoming variables are assumed to be negligible in Equation (2). For the lapping process we illustrate in this work, such an assumption is supported by an analysis of the physical mechanism and engineering knowledge. If the interactions among these factors are significant, Equation (2) should be extended to take these effects into consideration. In such cases, the following batch allocation scheme applies and the R2R control algorithm should be changed accordingly. However, the idea of improving quality through the coordination between upstream and downstream stations by using incoming information and R2R process control is still important and should be pursued.

To achieve a higher level of process capability and quality, the process recipes should be optimized such that the Mean Square Error (MSE) of the process output is minimized. Let τ be the target value of y_{ij}. Then, the MSE of n batches, each with a fixed batch size M, is given by

$$MSE = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ij} - \tau)^2 \tag{3}$$

or, equivalently,

$$MSE = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(y_{ij} - \mu_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \tau)^2, \tag{4}$$

where μ_i is the average output value of batch i.

The objective of R2R control is to minimize Equation (4). In Equation (4), the first term measures the mean square deviation of each wafer from the batch average, whereas the second term measures the mean square deviation of each batch from the target. That is, the total quality variation in a batch manufacturing process can be divided into two parts: within-batch variation and batch-to-batch variation.

Equation (2) shows that each output y_{ij} is determined by two factors: the incoming information $\mathbf{x}_{ij}$ and the recipe $\mathbf{u}_i$. An identical recipe is applied to all wafers in the same batch. Therefore, minimizing within-batch variation is equivalent to minimizing the variation in the incoming information $\mathbf{x}_{ij}$ of the same batch. That is, wafers allocated in the same batch should be *similar*. Once a batch allocation scheme is determined, the recipe should be designed based on the information of each batch. In this way, even if one batch is different from another, the *customized* recipe generated by an R2R controller could be used to move all batches toward the same quality target and thereby minimize the overall MSE. Because this procedure involves both batch allocation and R2R EWMA control, we name this strategy the batch-EWMA controller.

The structure of the proposed batch-EWMA controller is illustrated in Fig. 2. Different from the traditional structure shown in Del Castillo and Hurwitz (1997) and others, the input information $\mathbf{x}_{ij}$ also appears in the framework and is fed into the controller.

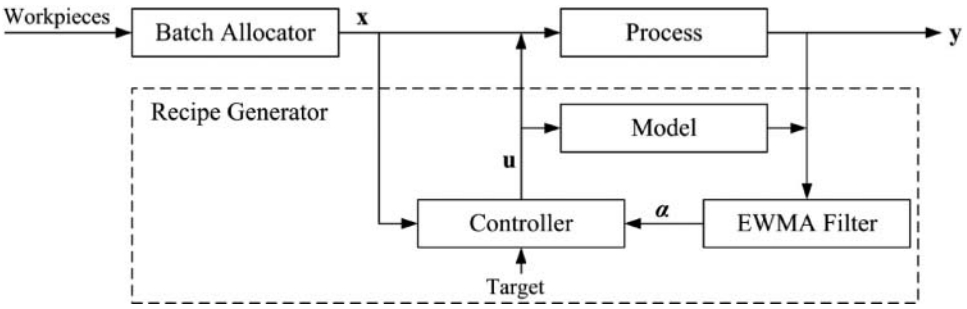


Fig. 2. Schematic structure of the batch-EWMA controller.

The batch-EWMA controller consists of two main components: a batch allocator and a recipe generator. First, the batch allocator groups similar wafers into batches with small wafer-to-wafer variations. Then, a control algorithm is put into place to generate control recipes for each batch. It should be noted that although each batch has its own level of incoming values, the batches share certain common information, such as intercept α and coefficient $\mathbf{b}$. These parameters are determined by the physical processing mechanism of the lapping stage, and all wafers should obey the same material removal mechanism.

In this article, we modify the widely used EWMA controller to generate recipes because this controller is capable of compensating for initial bias and non-stationary IMA disturbances. The modification is made to take incoming information into consideration. In the following section, we first introduce the recipe generation algorithm by assuming that wafers are already grouped into different batches; in addition, the means of all batches $\bar{\mathbf{x}}_i$ are known. The batch allocation algorithm is presented and its performance is studied in the subsequent section.

It is emphasized that the choice of the control algorithm heavily depends on the assumed process dynamics. Other types of controllers, including those introduced in the Introduction section, could be adapted to replace the EWMA controller used here if the true process has a totally different dynamics structure. In addition, Apley and Kim (2004) and Wang and Tsung (2007) emphasized that when parameter estimation uncertainty is being considered, the control action is usually more conservative than the usual case. A cautious controller may be employed if such uncertainty is significant.

To implement an R2R controller using Equation (2), initial estimates of model parameters need to be obtained first. In practice, these parameters could be estimated offline using design of experiments or regression analysis based on historical data. The R2R controller can then recursively update parameter estimates and help generate recipes more suitable for each run. However, initial values could potentially affect control stability (see, for example, Ingolfsson and Sachs (1993), Tseng *et al.* (2002), and Good and Qin (2006) for the stability conditions of single-input–single-output models with an EWMA controller and multi-input–multi-output models with an MEWMA controller).

Let α_{i-1} be the estimated intercept parameter before batch i. To compensate for initial estimation bias and potential process shifts or drifts, when the new output from batch i becomes available, the intercept parameter is first updated using an EWMA equation as follows:

$$\alpha_i = \omega(y_i - \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i - \mathbf{b}^{\mathrm{T}}\mathbf{u}_i) + (1-\omega)\alpha_{i-1}, \tag{5}$$

where ω is a turning parameter.

To minimize batch differences (the second term in Equation (4)), the output of each batch should target to τ. Based on the derivation in Ingolfsson and Sachs (1993), the recipe for the next run should be selected to satisfy:

$$\tau = \alpha_i + \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i + \mathbf{b}^{\mathrm{T}}\mathbf{u}_i. \tag{6}$$

The solution for run $(i+1)$ is then chosen as the projection of $\mathbf{u}_i$ onto the contour $\tau = \alpha_{i-1} + \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i + \mathbf{b}^{\mathrm{T}}\mathbf{u}$:

$$\mathbf{u}_{i+1} = \frac{\tau - \alpha_i - \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i}{\mathbf{b}^{\mathrm{T}}\mathbf{b}}\mathbf{b} + \left(\mathbf{I} - \frac{\mathbf{b}\mathbf{b}^{\mathrm{T}}}{\mathbf{b}^{\mathrm{T}}\mathbf{b}}\right)\mathbf{u}_i. \tag{7}$$

Solution (7) is an extension to the solution given by Ingolfsson and Sachs (1993). The only difference is the newly added term $-\mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i$, which is used to account for the variation in the incoming workpieces. As suggested by Ingolfsson and Sachs (1993), this choice achieves the objective stated in Equation (6) and minimizes recipe changes between runs, as measured by its Euclidian norm:

$$\|\mathbf{u}_{i+1} - \mathbf{u}_i\| = \sqrt{(\mathbf{u}_{i+1} - \mathbf{u}_i)^{\mathrm{T}}(\mathbf{u}_{i+1} - \mathbf{u}_i)}.$$

3. A fixed-capacity *K*-means clustering algorithm

As shown in Equation (4), the total output variation can be divided into two parts: within-batch variation and batch-to-batch variation. Batch-to-batch variation is expected to be reduced by the controller introduced in the previous section; the location of all batches is driven toward the same target value by the customized recipes. In this section, we

introduce a batch allocation algorithm to reduce within-batch variation. The batch allocator should form clusters by putting *similar* wafers in the same batch. Considering the capacity constraint of the machine for each processing stage, as illustrated in Table 1, the size of each cluster is restricted by each machine.

3.1. *A modified* K*-means clustering algorithm*

Considering the process model in Equation (2), we can rewrite the objective function in Equation (4) as follows:

$$MSE = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\alpha_i + \mathbf{a}^{\mathrm{T}}\mathbf{x}_{ij} + \mathbf{b}^{\mathrm{T}}\mathbf{u}_i - \mu_i)^2 + \frac{1}{n}\sum_{i=1}^{n}(\mu_i - \tau)^2.$$

The first term in the equation measures the deviation of each wafer from its batch mean. Using $\mu_i = \alpha_i + \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_i + \mathbf{b}^{\mathrm{T}}\mathbf{u}_i$, the first term becomes

$$M_1 = \frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{a}^{\mathrm{T}}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i))^2 \tag{8}$$

or, equivalently,

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

where

$$\Sigma^{-1} = \mathrm{diag}\left\{a_1^2, \cdots, a_d^2\right\}.$$

Equation (8) transforms the within-batch variation in the *process output* to the within-batch variation in the *process input*. After this transformation, the two terms, within-batch variation and between-batch variation, become independent. Therefore, the separate minimization of both terms would lead to the minimization of the overall MSE. The minimization of the between-batch variation is contributed by Equation (7), which produces a customized recipe for each batch based on their respective batch mean; the minimization of the within-batch variation can be expressed as follows:

$$\min\left(\frac{1}{nM}\sum_{i=1}^{n}\sum_{j=1}^{M}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)\Sigma^{-1}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)\right), \tag{9}$$

which could be minimized by a clustering algorithm.

K-means is a classic algorithm in data mining for clustering (see Sağlam *et al.* (2006) for a brief summary of the existing works on this basis; the method is also introduced in popular texts such as Kaufman and Rousseeuw (1990), Hastie *et al.* (2001), and Everitt *et al.* (2011)). The Euclidean or statistical distance between an observation and a cluster mean is used to measure similarity; samples are partitioned into clusters by minimizing an objective function analogous to Equation (9). In the conventional *K*-means algorithm, *k* cluster centers are first initialized and then each observation is assigned to a nearest cluster (or create a new cluster the point being far from all centers). After the assignment, all cluster centers are updated and then all points are reassigned to their nearest clusters (since the cluster centers may have changed after the assignment). In this process, new clusters may be created and existing clusters merged. The procedure is repeated until the assignment of all points remains unchanged. More details about the *K*-means algorithm can be found in Hastie *et al.* (2001).

The heuristic *K*-means algorithm does not always guarantee optimality (MacQueen, 1967). Selim and Ismail (1984) pointed out that the *K*-means algorithm is sensitive to the initial selection of cluster centers. They also provided a rigorous proof of the convergence of the *K*-means-type algorithm and suggested ways of obtaining local minima when it may fail. Phanendra Babu and Narasimha Murty (1993) suggested that repeating the algorithm several times with different starting points can give a more reliable clustering solution, although the global optimality is still not guaranteed. Due to its simplicity in implementation, the *K*-means algorithm is perhaps the most widely used algorithm in practice (Sağlam *et al.*, 2006).

However, the traditional *K*-means algorithm cannot satisfy the demand of grouping wafers in semiconductor manufacturing directly since the resulting cluster size is uneven. In real production, to make full utilization of machine capacity, the number of products in each batch should be equal and also fill the capacity. It is possible that the total number cannot be exactly divided by the batch number. In such cases, it is a common practice to fill all early batches but leave space in the last one.

Considering the batch size limitation in wafer batch allocation, we modify the conventional *K*-means algorithm slightly. Suppose that there are nM points to be assigned into n equal-sized clusters. The clustering procedure is illustrated as follows.

1. Randomly assign all points into the n equal-size D clusters; calculate the within-batch variation of each batch and sum them up.
2. For point $i = 1$ in batch 1, try to *switch* it with a point in a different batch and then calculate the overall within-batch variation. Then try to switch point i with another point, until all points that are not in batch 1 are tested. Identify the point where, if it is switched with point i, the largest reduction of within-batch variation is achieved. Finally, switch point i with the selected point.
3. Repeat Step 2b for all nM points.
4. Repeat Steps 2 and 3 until no further switch is needed. Then, the total within-batch variation is minimized.

The above *switch* operation guarantees that the number of clusters is unchanged, the number of points in each cluster is also fixed, and, in the meantime, the within-batch

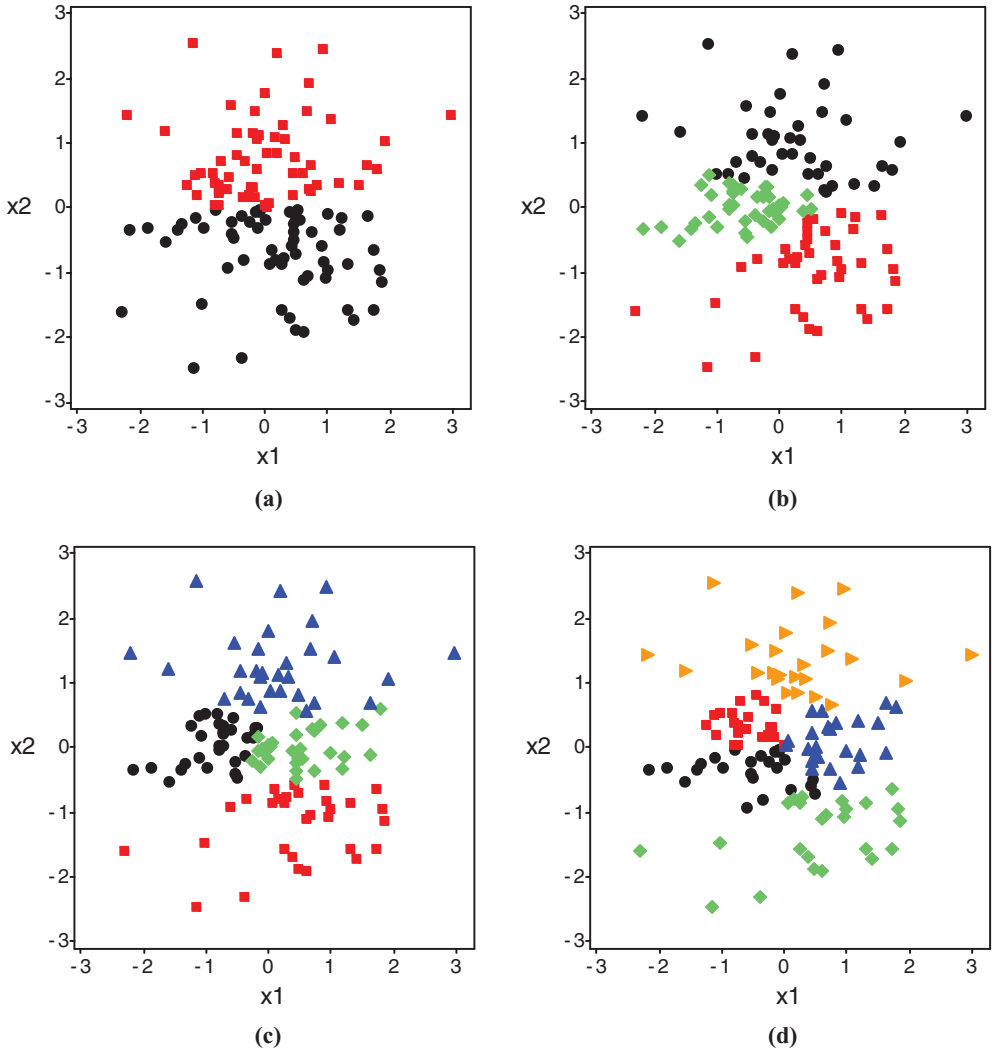


Fig. 3. Results of clustering a sample of 120 two-dimensional observations: (a) $K = 2$; (b) $K = 3$; (c) $K = 4$; and (d) $K = 5$ (color figure provided online).

variation is minimized. If unequal batch sizes are initially assigned, the procedure still helps keep all batch sizes unchanged. Therefore, the above method still works if the total number of products cannot be assigned to equal-sized batches. In the semiconductor manufacturing process we discussed, the above modified K-means algorithm is capable of allocating wafers in such a way that the practical engineering constraints are satisfied.

The modified K-means algorithm can only provide a local minimum. As suggested by one referee, repeating the algorithm with different initial assignments may generate a better solution. This has been confirmed by our simulation studies. Therefore, we incorporate this strategy in the following studies when minimizing the within-batch variation.

3.2. *Performance of the fixed-capacity clustering algorithm*

In this section, we study the performance of the proposed clustering algorithm via numerical simulations. Because it is difficult to graphically represent clustering results with more than three dimensions, we first simulate a two-dimensional process with each $\mathbf{x}_{ij}$ having two variables. A set of 120 randomly selected samples is generated by assuming $x_{ij(1)}, x_{ij(2)} \sim N(0, 1)$. The number of clusters to be generated is set to two, three, four, and five. Without loss of generality, let $\mathbf{a} = (1, 2)^{\mathrm{T}}$. Thus, we have $\Sigma^{-1} = \mathrm{diag}\{1, 4\}$ in the objective function of Equation (9).

The clustering results are shown in Figs. 3(a) to 3(d). It is clear that all of the points are partitioned into equally sized batches. The points in the same cluster are close to each other, indicating a small within-batch variation. Different batches have distinct location shifts; therefore, the batch-to-batch variation is large.

Table 3 compares the within-batch variations resulting from two methods: those from random grouping and those from clustering using the modified K-means algorithm. It is evident that if observations are clustered by the fixed-capacity K-means algorithm, the within-batch variation is

Table 3. Comparison of within-batch variations using different clustering schemes

Number of clusters	*Random assignment*	*Fixed-capacity* K-*means*
2	4.54	2.27
3	4.52	1.57
4	4.45	1.19
5	4.41	1.01

reduced significantly by between 50 and 77% compared with the variation produced by random grouping.

3.3. *Remarks and other algorithms for batch allocation*

The modified K-means algorithm introduced above serves as one way to minimize the within-batch variation, and hence, the overall variation in quality control. It should be noted that, similar to the conventional K-means algorithm that is limited to a local optimal solution (MacQueen, 1967; Sağlam *et al.*, 2006), the modified method can only find local minima. In addition, there are other clustering algorithms available for achieving the minimization of Equation (9).

As one referee suggested, batch allocation can be formulated as an integer programming problem and solved by existing algorithms. Sağlam *et al.* (2006) also introduced one way to formulate the clustering problem as a mixed-integer programing problem and developed a method to solve it. To solve Equation (9), we can define a *binary* decision variable $a_{k,i}$; $a_{k,i} = 1$ means that the kth point is assigned to batch i. We further force:

$$\sum_{i=1}^{n} a_{ki} = 1, \ \forall k, \ k = 1, \ldots, K,$$

so that each point is assigned to one batch only, and

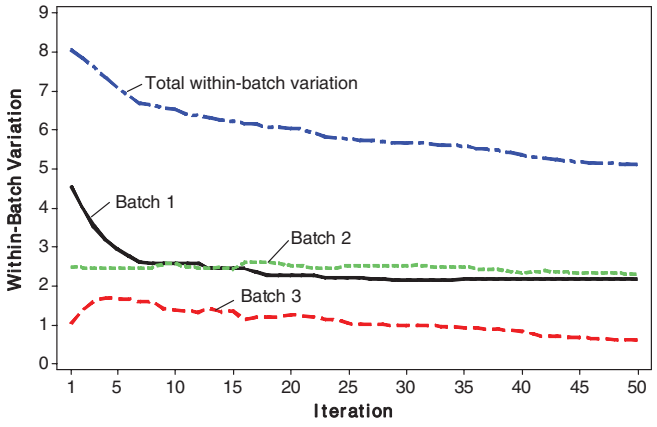
$$\sum_{k=1}^{K} a_{ki} = M, \ \forall i, \ i = 1, \ldots, n,$$

so that the batch contains exactly M points. The objective function in Equation (9) is cubic and can be solved by any Mixed-Integer Non-Linear Programming (MINLP) solver, such as the GAMS/DICOPT package (Grossmann *et al.*, 2002). Simulation results show that the solutions given by the integer programming and the modified K-means are either identical or very close. Similar to the K-means algorithm, the MINLP solver can only guarantee a local optimum.

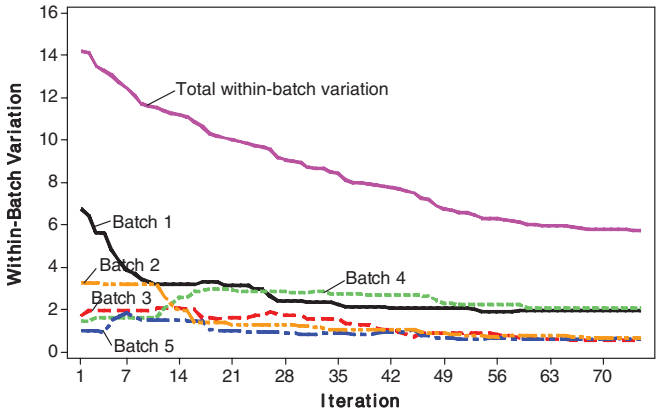
Compared with the modified K-means algorithm, the above MINLP formulation is straightforward and can be solved using existing commercial optimization packages. However, such a formulation may result in a large number of decision variables and thus increase the computational demand.

It should be noted that the modified K-means algorithm tries to minimize the overall within-batch variation, but it cannot guarantee that the variation of all clusters is minimized. Figure 4 shows the evolution of the within-batch variation of each batch and the summarization of all within-batch variations. It can be seen that as the algorithm runs, the summarization of the within-batch variation decreases, whereas the within-batch variations of certain batches may increase slightly. This means that switching a point from one batch to another batch does increase the within-batch variations, although the other one is decreased with a larger magnitude.

To use the batch allocation algorithm, there is an apparent need for data to characterize incoming products. In some existing applications, such information may not be readily available. This practical issue should be tackled from two aspects.



(a)



(b)

Fig. 4. Evolvement of the within-batch variations: (a) $K = 3$ and (b) $K = 5$ (color figure provided online).

1. In some processes, there is already a considerable amount of information available since advanced measurement machines can be used to collect continuous, multi-dimensional, and complex data. However, such information is unknown to the practitioners. By default, only a very rough fail/pass conclusion is used. In such cases, the quality practitioners should enquire about the capability of their measurement machine and make such information available.
2. If the coordination between the upstream and downstream stages is very important; intentional collection of needed information may be more cost-effective than just letting the defects happen.

4. Performance study of the batch-EWMA controller

To group incoming workpieces into different batches, we proposed to use the fixed-capacity K-means algorithm in the previous section to reduce within-batch variation. Since customized recipes are designed for each batch, all batches should be driven to a common target value. Therefore, the output quality after processing is expected to be improved. In this section, we apply the batch-EWMA controller to the lapping process and investigate its performance. The control performance is measured by the MSE given in Equation (3).

In this study, we first simulate an order of 100 ingots; each ingot generates 360 wafers after slicing. Batch allocation is required before the lapping stage for each ingot. The lapping batch is assumed to be 30. Therefore, each ingot is partitioned into 12 batches; 36 000 wafers in the form of 1200 batches are processed in total and the corresponding MSE is calculated. The disturbance in the lapping process is assumed to follow a normal distribution or have an IMA time series. After the simulation study, a real dataset is also analyzed to verify the controller's performance.

4.1. *Performance study under normal distributed disturbance*

Without loss of generality, suppose that the incoming quality is measured by five input variables; that is, the dimension of $\mathbf{x}_{ij}$ is five; there are four controllable variables, which means that $\mathbf{u}_i$ is a four-dimensional vector. The values of $\mathbf{x}_{ij}$ are generated from the standard normal distribution $N(0, 1)$. Suppose that the true values of the process parameters are given by

$$\alpha = 10, \quad \mathbf{a} = (1, 1, 1, 1, 1)^{\mathrm{T}}, \quad \mathbf{b} = (1, 1, 1, 1)^{\mathrm{T}}.$$

The process disturbance d_i in Equation (2) is assumed to follow the normal distribution $N(0, 1)$. The process target is set as 10. A dataset is generated using the above parameter settings; this dataset is assumed to be a historical one for initial parameter estimation.

First, we fit the linear model in Equation (2) with the presence of $\mathbf{x}_{ij}$ to the dataset; the estimates obtained are shown as follows:

$$\alpha_0 = 9.58, \ \mathbf{a} = (1.02, 0.93, 1.09, 1.06, 0.92)^{\mathrm{T}}, \\ \mathbf{b} = (1.24, 0.94, 0.87, 1.19)^{\mathrm{T}}.$$

If $\mathbf{x}_{ij}$ is excluded from the regression model, we have

$$\alpha_0 = 9.26, \quad \hat{\mathbf{b}} = (0.98, 1.02, 0.51, 1.03)^{T}.$$

The discount factor ω in Equation (5) is set to 0.3.

Four hypothetical scenarios are studied. The first scenario only uses the traditional EWMA controller. All workpieces are randomly grouped. In this situation, the input information $\mathbf{x}_{ij}$ is not utilized by the controller. At the end of each run, when an output becomes available, the recipe for the new run is determined based on the following equation (see Ingolfsson and Sachs (1993) for more details):

$$\mathbf{u}_{i+1} = \frac{\tau - \alpha_i}{\mathbf{b}^{\mathrm{T}}\mathbf{b}}\mathbf{b} + \left(\mathbf{I} - \frac{\mathbf{b}\mathbf{b}^{\mathrm{T}}}{\mathbf{b}^{\mathrm{T}}\mathbf{b}}\right)\mathbf{u}_i.$$

In the second scenario, $\mathbf{x}_{ij}$ is taken into account by the controller shown in Equation (7). However, the incoming wafers are still randomly clustered.

In the third scenario, incoming products are first clustered using the fixed-capacity K-means algorithm. However, the R2R controller ignores such information and still treats all clusters equally.

In the fourth scenario, the proposed batch-EWMA control strategy is applied. Wafers are first clustered using the fixed-capacity K-means algorithm and the within-batch variation is minimized; then, the EWMA controller in Equation (7) is applied to generate recipes for each batch. The control performance of the different scenarios is shown in Table 4.

It can be seen from Table 4 that scenario 2 performs slightly better than scenario 1. That is, the MSE is reduced if the input information is considered when generating recipes. However, since the clusters are formed randomly, the additional benefit by doing this procedure is not significant. Scenario 3 turns out to be the worst. That is, if the incoming products are grouped (then the between-batch

Table 4. Performance comparison under normally distributed disturbance

Scenario	*Clustering method*	*Utilization of input information*	*MSE*	*Std. dev. of MSE*
1	Random clustering	Not utilized	6.19	1.65
2	Random clustering	Utilized	6.17	1.59
3	Fixed-capacity K-means clustering	Not utilized	7.08	4.73
4	Fixed-capacity K-means clustering	Utilized	3.87	1.20

variation becomes large), but such grouping information is not considered by the controller, the control performance even deteriorates. This happens since the EWMA controller assumes that all batches have a common intercept parameter and it thus tries to update this parameter after each run when seeing a deviation between the true output and the target. However, since the output varies a lot due to the large batch difference, the estimated intercept parameter is wrongly changed repeatedly. Finally, if the fixed-capacity K-means algorithm is also implemented, compared to scenario 2, the MSE is lowered by 37%.

4.2. *Process with a disturbance following an IMA model*

As suggested in Montgomery *et al.* (2000), uncontrolled disturbances in certain industrial processes can be described by an IMA(1,1) series. Therefore, in this section, we also study the performance of the proposed controller when an IMA(1,1) model is presented in the process; the disturbance series d_i in Equation (2) takes the following form:

$$d_i = d_{i-1} + \varepsilon_i - \theta\varepsilon_{i-1},$$

where ε_i is a white noise series, $\varepsilon_i \sim N(0, 1)$, and the coefficient θ is set to 0.6. All other model parameters are the same as those given in Section 4.1.

Similar to Section 4.1, four different clustering/control strategies are applied to control 12 batches of simulated wafers. The control performance is shown in Table 5. The general performance pattern is similar to that shown in Table 4. Once again, compared to scenario 2, the overall MSE is reduced significantly by 38% when the proposed batch-EWMA control framework is utilized.

4.3. *Performance study using a real dataset*

In this section, we apply the proposed batch-EWMA controller to a real dataset collected before the lapping stage in semiconductor manufacturing. The dataset contains information on 360 wafers. Before entering the lapping stage, all wafers are measured by an automated machine; the status of each wafer is characterized by five variables, which is represented by $\mathbf{x}_{ij}$ in the model. In the lapping stage, four control variables are designed to be controllable; the settings of these variables are to be determined by the R2R controllers. One main target of the lapping stage is to achieve thickness uniformity. Therefore, the goal of our R2R controllers here is to achieve a minimum thickness variation of all wafers.

Table 5. Performance comparison under IMA disturbance

Scenario	*Clustering method*	*Utilization of input information*	*MSE*	*Std. dev. of MSE*
1	Random clustering	Not utilized	6.20	1.68
2	Random clustering	Utilized	6.22	1.59
3	Fixed-capacity K-means clustering	Not utilized	7.16	4.74
4	Fixed-capacity K-means clustering	Utilized	3.86	1.22

Table 6. Estimated parameter values of the process model

Parameters	*Initial model for the EWMA controller*	*Initial model for the batch-EWMA controller*
Intercept	550.439	584.060
a_1	—	−0.0574
a_2	—	−0.0238
a_3	—	0.9316
a_4	—	0.0603
a_5	—	0.0225
b_1	−0.0451	−0.0744
b_2	0.1417	0.1607
b_3	−0.0252	−0.0272
b_4	−0.0026	−0.0016

To understand the behavior of the lapping machine, designed experiments were conducted in practice and data were collected for model building. For performance comparison, we need to generate two linear models for initializing R2R control. One model, designed for the proposed batch-EWMA controller, has the input information $\mathbf{x}_{ij}$ in the formula; the other model, designed for the conventional EWMA controller, excludes $\mathbf{x}_{ij}$ from the formula since this controller does not use such information in recipe generation. Parameters of the fitted models are shown in Table 6. A normality test confirms that the residuals have a normal distribution.

The initial control variable is determined based on the mean of all historical observations:

$$\mathbf{u} = [21 \quad 25 \quad 66 \quad 215]^{\mathrm{T}}.$$

The capacity of the lapping machine is limited to 30, which leads to 12 batches for this order. All of these batches are fed into the process sequentially; the traditional EWMA controller and the proposed batch-EWMA controller are applied to the same dataset; common factors in both controllers are set to equal values for a fair comparison. The

Table 7. Performance comparison when applied to the real example

Controller	*Clustering method*	*Utilization of input information*	*MSE*
EWMA	Random clustering	Not utilized	8.15
Batch-EWMA	Fixed-capacity K-means clustering	Utilized	4.99

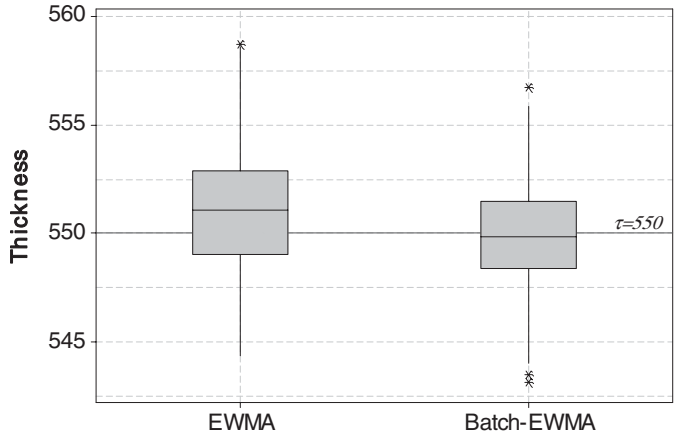


Fig. 5. Boxplot of process output under EWMA and batch-EWMA controllers.

targeted output thickness is 550 μm and the EWMA discount factor in Equation (5) is set to 0.3.

Figure 5 shows the output thickness of the individual wafers under different control scenarios. The thickness distribution under the batch-EWMA strategy is clearly more concentrated than that under the traditional R2R strategy. As shown in Table 7, the MSE of the simulated wafer thicknesses is reduced to 4.99, which is approximately 61.2% of the MSE produced by the EWMA controller.

5. Conclusions

In certain semiconductor manufacturing processes, workpieces have to be grouped into clusters before processing due to limited equipment capacity. The final output quality is affect by the within-batch variation and batch-to-batch variation.

In this article, we proposed a batch-EWMA control strategy based on a practical batch size constraint. A modified fixed-capacity K-means algorithm is first utilized to cluster incoming materials and minimize the within-batch variation. Cluster information is fed to the controller and a new recipe is generated by the batch-EWMA controller using the cluster information and the feedback information from the previous run. In this way, the recipes for different batches are customized and therefore the between-batch variation is minimized. Simulation studies reveal that the new framework can reduce output variation and improve the product quality significantly.

The control scheme proposed in this article is designed for a single manufacturing stage. Since a semiconductor manufacturing process and, in fact, many processes in other industries usually involve multiple stages, coordinating additional stages and their corresponding batch allocation could be interesting topics for future research.

Acknowledgements

We thank the editor and two anonymous referees for their valuable suggestions, which have helped to significantly improve this work. This work was supported by the National Natural Science Foundation of China under grant 71072012.

References

Apley, D.W. and Kim, J. (2004) Cautious control of industrial process variability with uncertain input and disturbance model parameters. *Technometrics*, **46**(2), 188–199.

Butler, S.W. and Stefani, J.A. (1994) Supervisory run-to-run control of polysilicon gate etch using in-situ ellipsometry. *IEEE Transactions on Semiconductor Manufacturing*, **7**(2), 193–201.

Del Castillo, E. and Hurwitz, A.M. (1997) Run-to-run process control: literature review and extensions. *Journal of Quality Technology*, **29**(2), 184–196.

Del Castillo, E., Pan, R., and Colosimo, B.M. (2003) A unifying view of some process adjustment methods. *Journal of Quality Technology*, **35**(3), 286–293.

Everitt, B.S., Landau, S., Leese, M., and Stahl, D. (2011) *Cluster Analysis*, John Wiley & Sons, West Sussex, UK.

Good, R.P. and Qin, S.J. (2006) On the stability of MIMO EWMA run-to-run controllers with metrology delay. *IEEE Transactions on Semiconductor Manufacturing*, **19**(1), 78–86.

Grossmann, I.E., Viswanathan, J., Vecchietti, A., Raman, R., and Kalvelagen, E. (2002) *GAMS/DICOPT: A Discrete Continuous Optimization Package*, GAMS Corporation Inc., Washington DC.

Hastie, T., Tibshirani, R., and Friedman, J. (2001) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY.

He, F., Wang, K., and Jiang, W. (2009) A general harmonic rule controller for run-to-run process control. *IEEE Transactions on Semiconductor Manufacturing*, **22**(2), 232–244.

Ingolfsson, A. and Sachs, E. (1993) Stability and sensitivity of an EWMA controller. *Journal of Quality Technology*, **25**(4), 271–287.

ITRS. (2009) International technology roadmap for semiconductors. Available at http://www.itrs.net/Links/2009ITRS/Home 2009.htm, accessed September 2012.

Jin, M. and Tsung, F. (2009) Smith–EWMA run-to-run control schemes for a process with measurement delay. *IIE Transactions*, **41**(4), 346–358.

Kaufman, L. and Rousseeuw, P.J. (1990) *Finding Groups in Data: An Introduction to Cluster Analysis*, John Wiley & Sons, Hoboken, NJ.

Lian, Z., Colosimo, B.M., and Del Castillo, E. (2006) Setup error adjustment: sensitivity analysis and a new MCMC control rule. *Quality and Reliability Engineering International*, **22**(4), 403–418.

Lin, J. and Wang, K. (2011) Online parameter estimation and run-to-run process adjustment using categorical observations. *International Journal of Production Research*, **49**(13), 4103–4117.

Lin, J. and Wang, K. (2012) A Bayesian framework for online parameters estimation and process adjustment using categorical observations. *IIE Transactions*, **44**, 291–300.

MacQueen, J. (1967) Some methods for classification and analysis of multivariate observations, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 281–297.

Montgomery, D.C. (2005) *Introduction to Statistical Quality Control*, John Wiley & Sons, Hoboken, NJ.

Montgomery, D.C., Keats, J.B., Yatskievitch, M., and Messina, W.S. (2000) Integrating statistical process monitoring with feedforward control. *Quality and Reliability Engineering International*, **16**(6), 515–525.

Phanendra Babu, G. and Narasimha Murty, M. (1993) A near-optimal initial seed value selection in *k*-means algorithm using a genetic algorithm. *Pattern Recognition Letters*, **14**(10), 763–769.

Sachs, E., Guo, R.S., Ha, S., and Hu, A. (1991) Process-control system for VLSI fabrication. *IEEE Transactions on Semiconductor Manufacturing*, **4**(2), 134–144.

Sachs, E., Hu, A., and Ingolfsson, A. (1995) Run by run process control - combining SPC and feedback-control. *IEEE Transactions on Semiconductor Manufacturing*, **8**(1), 26–43.

Sağlam, B., Salman, F.S., Sayın, S., and Türkay, M. (2006) A mixed-integer programming approach to the clustering problem with an application in customer segmentation. *European Journal of Operational Research*, **173**(3), 866–879.

Selim, S.Z. and Ismail, M.A. (1984) K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **6**(1), 81–87.

Shang, Y., Wang, K., and Tsung, F. (2009) An improved run-to-run process control scheme for categorical observations with misclassification errors. *Quality and Reliability Engineering International*, **25**, 397–407.

Tseng, S.T., Chou, R.J., and Lee, S.P. (2002) A study on a multivariate EWMA controller. *IIE Transactions*, **34**(6), 541–549.

Tseng, S.T., Tsung, F., and Liu, P.Y. (2007) Variable EWMA run-to-run controller for a drifted process. *IIE Transactions*, **39**, 291–301.

Tseng, S.T., Yeh, A.B., Tsung, F., and Chan, Y.Y. (2003) A study of variable EWMA controller. *IEEE Transactions on Semiconductor Manufacturing*, **16**(4), 633–643.

Tsung, F. and Shi, J.J. (1999) Integrated design of run-to-run PID controller and SPC monitoring for process disturbance rejection. *IIE Transactions*, **31**(6), 517–527.

Wang, K. and Tsung, F. (2007) Run-to-run process adjustment using categorical observations. *Journal of Quality Technology*, **39**(4), 312–325.

Wang, K. and Tsung, F. (2010) Recursive parameter estimation for categorical process control. *International Journal of Production Research*, **48**(5), 1381–1394.

Biographies

Kaibo Wang is an Associate Professor in the Department of Industrial Engineering, Tsinghua University, Beijing, China. He received his B.S. and M.S. degrees in Mechatronics from Xi'an Jiaotong University, Xi'an, China, and his Ph.D. in Industrial Engineering and Engineering Management from the Hong Kong University of Science and Technology, Hong Kong. He has published papers in journals such as *Journal of Quality Technology*, *IIE Transactions*, *Quality and Reliability Engineering International*, *International Journal of Production Research*, and others. His research is devoted to statistical quality control and data-driven complex system modeling, monitoring, diagnosis, and control, with a special emphasis on the integration of engineering knowledge and statistical theories for solving problems from real industry.

Kai Han received his B.S. and M.S. degrees in Industrial Engineering from Tsinghua University, Beijing, China, in 2009 and 2012, respectively. His research focuses on the modeling and control of semiconductor manufacturing processes.