Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Context-based State Estimation in Semiconductor Manufacturing: Reference Path Based State Transformation Approach

An-Jhih Su*, Cheng-Ching Yu*¶, Jyh-Cheng Jeng, Hsiao-Ping Huang*, Cheng-Jer Yang***, Hung-Wen Chiou***, and Shu-Ching Yang*****

*National Taiwan University, Taipei, 10617 Taiwan
¶ (Tel: 886-2-3366-3037; e-mail: ccyu@ ntu.edu.tw)
**National Taipei University of Technology, Taipei, 10608 Taiwan
***ProMOS Technologies Inc., Hsinchu, 30078, Taiwan

Abstract: There are many possible factors in semiconductor manufacturing processes such as metrology tool bias, product type, or chamber that may induce disturbance to the process output. To account for all these types, a context-based model is often used. The most important feature of a context-based system is rank deficiency, and therefore we propose a method that unbiasedly estimates the relative status of each context and process output by state transformation. The transformed states are straightforward and physically meaningful. Furthermore, a solution of planning paths with a guarantee of output performance is also investigated. The other application of particle count estimation for data from a real fab is also demonstrated.

1. INTRODUCTION

In semiconductor manufacturing, there are hundreds of process steps, and the overall process can be treated as a sequence of batch processes. In a given process step, wafers may have come from one of several alternative equipment tools. The multi-tool and sequential process features result in a path-relative problem. Usually, the path is determined based on the experience of engineers or operators instead of theoretical analysis. Furthermore, there are many product types (especially in a foundry fab), sub-paths inside one piece of process equipment (e.g. chambers), and many metrology tools. These features make the path relative problem complicated as shown as Fig. 1.

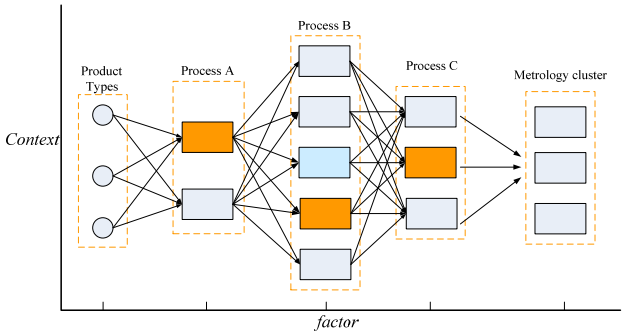

Fig. 1. A sketch of a context-based system.

Fig. 1 also indicates an important feature, which is that the measurement of quality data is usually taken only after several sequential process steps are completed. From the viewpoint of quality control, it is not necessary to measure every lot (or wafer). Besides, some types of metrology are very time-consuming (e.g. measuring the number of particles).

To form a path problem, there are two axes which are the factor axis and context axis. The factor is a categorical collection of components that will contribute to the system output. In semiconductor manufacturing, the factor could be product type, layer, equipment tool, materials, chamber, or sub-stage inside a piece of equipment. For a factor, there are several instances that can be chosen for processing, and these members refer to contexts (or conditions, levels, treatments). It must be noted that only one context can be applied for each factor.

Estimating the path-relative output quality for virtual metrology or feeding information to the next process step allows for feedforward control at the next process step. Edward et. al. (2001) not only use path information but also tool raw data, time, operator, etc. to successfully estimate electric parameters using neural network. Edgar et. al. (2004) introduced the concept of contexts, and a Kalman filter based state estimator for multi-product and multi-tool system is proposed (Pasadyn and Edgar, 2005). Firth (2002) proposed just-in-time adaptive disturbance estimation (JADE) which is a recursive, context-based state estimation. JADE is an efficient method for sharing information compared to thread based methods, and Wang et. al. (2007) points out JADE is actually equivalent to the recursive least squares method but with a fixed covariance matrix. Wu et. al. (2006) propose a context-based analysis-of-covariance (ANCOVA) model including time series models for each context to capture the process dynamics and improve output prediction capability.

The most important feature of a context-based problem is rank deficiency (Firth, 2002; Pasadyn et. al. 2005; Edgar et. al., 2004; Wang et. al., 2007). Besides estimation of process output, our motivation is: What other information could be obtained in a rank deficient system? The rest of this paper is organized as follows: In Section 2, we introduce the

procedure of state transformation, and show that the transformed states are straightforward and physically meaningful. The compact state space model of a context system is given in Section 3, such that a general state estimation technique (e.g. Kalman filtering) can be applied to it. In Section 4, we propose a strategy for path decision to optimize the process output. For the other application we consider the particle count estimation as a context-based problem. Data from a real fab are provided and used for validation. Conclusions follow in Section 5.

2. PROBLEM DEFINITION AND FORMULATION

Model selection and identification is not discussed in this work. We assume that the contexts to be used have been validated via a statistical significance test such as analysis of variance (ANOVA). The system is assumed to be linear in the sense that output is a linear combination of contexts in which only one context is chosen from each factor.

Define a system which has f factors, and n_i (i=1,2,…,f) contexts for each of the i factors. Use a vector, **n,** to stand for the set of n_i:

$$\mathbf{n} = \begin{bmatrix} n_1 & n_2 & \cdots & n_f \end{bmatrix} \in I^f \tag{1}$$

where I is the set of integers, and n_i>1. Consequently, the total number of contexts, N, in the system is simply obtained as:

$$N = \sum_{i=1}^{f} n_i \tag{2}$$

The notation $x_{i,j}$ means the state of the j^{th} context (j=1,2,…,n_i) in the i^{th} factor. A set of contexts within the i^{th} factor is

$$\mathbf{x}_i = \begin{bmatrix} x_{i,1} & x_{i,2} & \cdots & x_{i,n_i} \end{bmatrix}^T \in R^{n_i} \tag{3}$$

and the set of all contexts in the system is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T & \cdots & \mathbf{x}_f^T \end{bmatrix}^T \in R^N \tag{4}$$

A path indicates which contexts are used for each factor. Based on this, the *path description vector* of the k^{th} run is defined as

$$\mathbf{p}(k) = \begin{bmatrix} p_1(k) & p_2(k) & \cdots & p_f(k) \end{bmatrix} \in I^f$$
$$p_i(k) \in \{1, 2, ..., n_i\} \tag{5}$$

where $p_i(k)$ indicates which context is used in the i^{th} factor. An observation of system output associated with a given path $\mathbf{p}(k)$ can be obtained:

$$y(k) = \sum_{i=1}^{f} x_{i,p_i(k)} + v(k) \tag{6}$$

where v is the white noise with variance R.

2.1 System Matrix and Characteristics

To obtain the system matrix, unfold $\mathbf{p}(k)$ into a *context-based path vector*:

$$\mathbf{a}(k) = \begin{bmatrix} \mathbf{a}_1(k) & \mathbf{a}_2(k) & \cdots & \mathbf{a}_f(k) \end{bmatrix} \in B^N \tag{7}$$

$$\mathbf{a}_i(k) = \begin{bmatrix} a_{i,1}(k) & a_{i,2}(k) & \cdots & a_{i,n_i}(k) \end{bmatrix} \in B^{n_i} \tag{8}$$

$$a_{i,j}(k) = \begin{cases} 1, & j = p_i(k) \\ 0, & j \neq p_i(k) \end{cases} \tag{9}$$

where B stands for Boolean number. $\mathbf{a}(k)$ indicates whether each context is used or not, and it always has the same dimensions as $\mathbf{x}$. Thus, (6) can also be represented as

$$y(k) = \mathbf{a}(k)\mathbf{x}. \tag{10}$$

Furthermore, with M observations Eg. (10) becomes the linear equation:

$$\mathbf{A}\mathbf{x} = \mathbf{y} \tag{11}$$

where $\mathbf{A} = \begin{bmatrix} \mathbf{a}^T(1) & \mathbf{a}^T(2) & \cdots & \mathbf{a}^T(1) \end{bmatrix}^T_{M \times N}$ and $\mathbf{y} = \begin{bmatrix} y(1) & y(2) & \cdots & y(M) \end{bmatrix}^T$.

One may try to solve $\mathbf{x}$ in (11) by the least squares method if adequate data is available. However, Wang et. al. (2007) point out that this linear equation suffers rank deficiency:

$$\text{rank}(\mathbf{A}) \leq N - f + 1 \tag{12}$$

Since $\mathbf{A}$ is always rank deficient, $\mathbf{x}$ cannot be unbiasedly and uniquely identified. Although one can predict the system output with adequate data, there are still two issues of interest. First, the status of contexts cannot be obtained due to the non-unique solution of $\mathbf{x}$. The other issue is: How should a rank deficient model be updated when the solution of $\mathbf{x}$ is non-unique? A very inefficient method is to keep all path data as a growing matrix and invert it every time prediction is needed. To solve these two problems, the following method based on dimensionality reduction is proposed.

2.2 Reference Path Based State Transformation

From (13), the maximum rank of the system is analytically known, and the rank should be equal to N-f+1 for a space including all possible paths. The rank number provides information about how many *independent* relationships exist. Clearly, the number of new states after transformation should be equal to the rank number, and these new states should be independent of each other. Now, the question is how to transfer original context states, $\mathbf{x}$, into new states, $\mathbf{z}$?

The first step of our proposed method is to select a specific path as the *reference path*, and treat its system output as reference output, or to a degree, *nominal* output. The reference path is constructed by selecting one reference context in each factor. This path is defined as $\mathbf{p}^{ref}$ which is an specific instance of $\mathbf{p}(k)$ in (5). The reference output is defined as the first new states:

$$z_{ref} = \sum_{i=1}^{f} x_{i,p_i^{ref}} = x_{1,p_1^{ref}} + x_{2,p_2^{ref}} + \cdots + x_{f,p_f^{ref}} = \mathbf{a}^{ref}\mathbf{x} \quad (13)$$

where $\mathbf{a}^{ref}$ is a context-based path vector associated with $\mathbf{p}^{ref}$. Next, to have information on status of each context, the deviation from the reference path is considered. For each context, deviation from the reference path define the remaining new states:

$$z_{i,j} = x_{i,j} - x_{i,p_i^{ref}} \ (i=1,2,\ldots,f \text{ and } j=1,2,\ldots,n_i) \quad (14)$$

Note that $z_{i,j}=0$ when $j=p_i^{ref}$ because there is no deviation for a reference context from itself. Since the state, $z_{i,p_i^{ref}}$, is redundant, the set of deviation states in the i^{th} factor with $z_{i,p_i^{ref}}$ removed is:

$$\mathbf{z}_i = \begin{cases} \begin{bmatrix} z_{i,1} & \cdots & z_{i,p_i^{ref}-1} & z_{i,p_i^{ref}+1} & \cdots & z_{i,n_i} \end{bmatrix}^T, & 1 < p_i^{ref} < n_i \\ \begin{bmatrix} z_{i,2} & z_{i,3} & \cdots & z_{i,n_i} \end{bmatrix}^T, & p_i^{ref} = 1 \\ \begin{bmatrix} z_{i,1} & z_{i,2} & \cdots & z_{i,n_i-1} \end{bmatrix}^T, & p_i^{ref} = n_i \end{cases} \quad (15)$$

where $\mathbf{z}_i \in R^{n_i-1}$. Since $\mathbf{z}_i$ is only a linear combination of $\mathbf{x}_i$, we use a matrix, $\mathbf{T}_i$, to indicate the relationship of state transformation for the i^{th} factor as $\mathbf{z}_i=\mathbf{T}_i\mathbf{x}_i$. From the definition of $\mathbf{z}_i$, the entries of $\mathbf{T}_i$ are given as:

$$\mathbf{T}_i(r, p_i^{ref}) = -1, \qquad r = 1,2,\ldots,n_i - 1 \quad (16)$$

$$\mathbf{T}_i^{(p_i^{ref})} = \mathbf{I}_{n_i-1} \quad (17)$$

(16) means the entries in the column p_i^{ref} are equal to -1, and (17) indicates that the *minor matrix* of $\mathbf{T}_i$ with column p_i^{ref} removed becomes an identity matrix with size n_i-1. Furthermore, application of (15) to all factors gives a new set of states:

$$\mathbf{z} = \begin{bmatrix} z_{ref} & \mathbf{z}_1^T & \mathbf{z}_2^T & \cdots & \mathbf{z}_f^T \end{bmatrix}^T \in R^{N-f+1} \quad (18)$$

The number of total new states is, N-f+1, the rank of a space including all possible paths. Thus, the relationship between the original and transformed states has been described. The relationship between $\mathbf{z}$ and $\mathbf{x}$ is:

$$\begin{bmatrix} z_{ref} \\ \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_f \end{bmatrix}_{(N-f+1)\times 1} = \begin{bmatrix} \mathbf{a}^{ref} \\ \mathbf{T} \end{bmatrix}_{(N-f+1)\times N} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_f \end{bmatrix}_{N\times 1} \quad (19)$$

where $\mathbf{T}$ is a block-diagonal matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{T}_f \end{bmatrix}_{(N-f)\times N} \quad (20)$$

For simplicity of notation, define

$$\tilde{\mathbf{T}} = \begin{bmatrix} \mathbf{a}^{ref} \\ \mathbf{T} \end{bmatrix} \quad (21)$$

and (19) becomes to

$$\mathbf{z} = \tilde{\mathbf{T}}\mathbf{x} \quad (22)$$

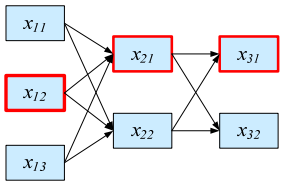

Fig. 2 Sketch of a 3-2-2 system in Example 1

Example 1 A system (Fig. 2) with 3 factors, 3-2-2, is used to illustrate the state transformation procedure. For this case, $\mathbf{n}$=[3 2 2]. If the reference path is selected as $\mathbf{p}^{ref}$=[2 1 1], the context-based path $\mathbf{a}^{ref}$=[0 1 0 1 0 1 0] can be obtained from (7)-(9). Next, (16) and (17) can be used to determine the transformation matrix, $\mathbf{T}_i$, for each factor: $\mathbf{T}_1$=[1 -1 0; 0 -1 1], $\mathbf{T}_2$=[-1 1], and $\mathbf{T}_3$=[-1 0]. Application of (19) to this example, gives:

$$\begin{bmatrix} z_{ref} \\ z_{11} \\ z_{13} \\ z_{22} \\ z_{32} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{13} \\ x_{21} \\ x_{22} \\ x_{31} \\ x_{32} \end{bmatrix} \quad (23)$$

After the transformation of $\mathbf{x}$ into $\mathbf{z}$, the linear system of (11), using new states but with the same output, becomes:

$$\tilde{\mathbf{A}}\mathbf{z} = \mathbf{b} \quad (24)$$

where $\tilde{\mathbf{A}} = \mathbf{A}\mathbf{T}^+$. Since $\tilde{\mathbf{A}}$ is full rank if adequate data is available, the standard method can be applied to (24) to get the solution of $\mathbf{z}$.

3. IMPLEMENTATION OF STATE ESTIMATION

3.1 State Space Model

If an *unknown* disturbance enters into a context, the status of a context will have dynamic behavior as shown as Fig. 3. The state $x_{i,j}$, to a degree, describes the unknown disturbance $\delta_{i,j}$ behavior as

$$x_{i,j}(t) = \overline{x}_{i,j} + \delta_{i,j}(t) \tag{25}$$

where t is the discrete time index, and $\overline{x}_{i,j}$ stands for the nominal value of $x_{i,j}$. By taking the difference between $x_{i,j}(t)$ and $x_{i,j}(t+1)$, the following relation can be obtained:

$$x_{i,j}(t+1) = x_{i,j}(t) + \Delta\delta_{i,j}(t) \tag{26}$$

where $\Delta\delta_{i,j}(t) = \delta_{i,j}(t+1) - \delta_{i,j}(t)$.

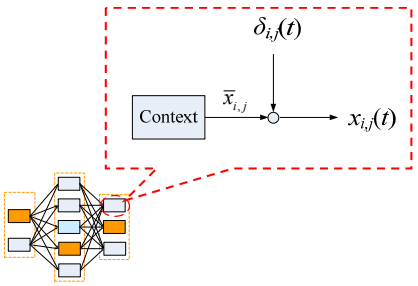

Fig. 3 Dynamics of a context

The run index, k, is used to count the number of runs that have occurred so far. However, a run will have several alternative contexts for processing, such that the amount of time used for a context will not be the same as the run number. It also must be noted that a context may not be used for a run, and consequently this context should just be kept at its previous value. Thus, we introduce an index, $k_{ij}(k)$, to count the number of times that the context $x_{i,j}$ is prior to run k. The mathematical relation for $k_{i,j}$ can be expressed as

$$k_{i,j}(k) = \sum_{l=1}^{k} a_{i,j}(l) \tag{27}$$

With the relation between the run index and context index, we rewrite (26) in the k domain:

$$x_{i,j}(k+1) = x_{i,j}(k) + a_{i,j}(k) \cdot d_{i,j}(k) \tag{28}$$

where $d_{i,j}(k) = \Delta\delta_{i,j}\left(k_{i,j}(k)\right)$

At run k from Eq(28), an unused context means $a_{i,j}(k)=0$ and $x_{i,j}$ will be just kept its previous run value.

The dynamics of each context can now be synchronized to the run domain k, and consequently the following state space model of a time-varying system can be obtained:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{x}(k) + \mathbf{D}(k) \cdot \mathbf{d}(k) \\ y(k) &= \mathbf{a}(k) \cdot \mathbf{x}(k) + v(k) \end{aligned} \tag{29}$$

where $\mathbf{d}$ is disturbance set vector defined as

$$\mathbf{d}(k) = \left[\mathbf{d}_1^T(k) \quad \mathbf{d}_2^T(k) \quad \cdots \quad \mathbf{d}_c^T(k)\right]^T \in R^N \tag{30}$$

$$\mathbf{d}_i(k) = \left[d_{i,1}(k) \quad d_{i,2}(k) \quad \cdots \quad d_{i,n_i}(k)\right]^T \in R^{n_i} \tag{31}$$

and $\mathbf{D}(k)$ is a diagonal matrix in which diagonal entries are given by $\mathbf{a}(k)$ (in short: $\mathbf{D}(k)=\mathrm{diag}(\mathbf{a}(k))$. Since $\mathbf{x}$ is used to describe unknown disturbances, we assume dynamics of all contexts can be modelled as random processes (or integrated white noise). Thus, $\mathbf{d}(k)$ will be a white noise sequence with covariance matrix $\mathbf{Q}$. Note that the state space in (29) is an unobservable linear time-varying system. Application of the state transformation matrix to (29) gives:

$$\begin{aligned} \mathbf{z}(k+1) &= \mathbf{z}(k) + \tilde{\mathbf{D}}(k) \cdot \tilde{\mathbf{d}}(k) \\ y(k) &= \tilde{\mathbf{a}}(k) \cdot \mathbf{z}(k) + v(k) \end{aligned} \tag{32}$$

where $\tilde{\mathbf{d}} = \mathbf{T}\mathbf{d}$ with covariance $\tilde{\mathbf{Q}} = \mathbf{T}\mathbf{Q}\mathbf{T}^T$, and $\tilde{\mathbf{D}}(k) = \mathrm{diag}\left(\tilde{\mathbf{a}}(k)\right)$.

Since a compact state space model is obtained, we apply a Kalman filter to (32) for state estimation. Thus, the recursive update equations are simplified to

$$\mathbf{K}(k) = \mathbf{P}(k)\tilde{\mathbf{a}}^T(k)\left(R(k) + \tilde{\mathbf{a}}(k)\mathbf{P}^-(k)\tilde{\mathbf{a}}^T(k)\right)^{-1} \tag{33}$$

$$\hat{\mathbf{z}}(k) = \hat{\mathbf{z}}(k-1) + \mathbf{K}(k)\left(y(k) - \tilde{\mathbf{a}}(k)\hat{\mathbf{z}}(k-1)\right) \tag{34}$$

$$\mathbf{P}(k+1) = \left(\mathbf{I} - \mathbf{K}(k)\tilde{\mathbf{a}}(k)\right)\mathbf{P}(k) + \tilde{\mathbf{D}}(k)\tilde{\mathbf{Q}}(k)\tilde{\mathbf{D}}^T(k) \tag{35}$$

where $\mathbf{K}$ is the blending factor, and $\mathbf{P}$ is the covariance matrix of the estimation error

3.2 Reference Path Selection

No matter what reference path is selected, the transformed states from different reference paths are in the same subspace and can be unbiasedly estimated. Since the reference path is a basis for comparison, it is suggested that the reference path be selected from less variant contexts or frequently used contexts. Usually, both criteria suggest the same context. The reference run path is usually steady such that deviation from it has greater physical meaning.

Some contexts may fade in or out, especially product or equipment type contexts. When a context that is a member of the reference path is not used for some time, the estimation of the reference output is not reliable. The solution to deal with this problem is to switch the reference path. Define $\mathbf{T}_{\mathrm{I}}$ and $\mathbf{T}_{\mathrm{II}}$ as transformation matrices associated with the original reference and the new reference path, respectively. Thus the relationship between original estimated states, $\hat{\mathbf{z}}_I$, and new states, $\hat{\mathbf{z}}_{II}$, is given as

$$\hat{\mathbf{z}}_{II} = \mathbf{T}_{II}\mathbf{T}_I^+\hat{\mathbf{z}}_I \tag{36}$$

and the original covariance matrix, $\mathbf{P}_{\mathrm{I}}$, and new one, $\mathbf{P}_{\mathrm{II}}$, have the relation:

$$\mathbf{P}_{II} = (\mathbf{T}_{II}\mathbf{T}_I^+)^T\mathbf{P}_I(\mathbf{T}_{II}\mathbf{T}_I^+) \tag{37}$$

By applying the above two equations, one can switch to the new reference path and then continue the recursion with the Kalman filter.

4. APPLICATIONS

4.1 Path Planning

Although the most important objective of path planning is maximizing the throughput, here we discuss how to arrange paths with a guarantee of output quality. For example, one can compensate for a tool in bad condition by feeding wafers processes by that tool to a tool in better condition in a subsequent process step and thereby achieve the average quality requirement. Since the relative condition of each context can be solved, it is possible to plan a path before starting a run. For simplicity, we consider a wafer processing route problem, such that all contexts stand for process tools. The system may have h runs fed into the system at one time, and these h runs are in parallel routes (no conflict of using the same tool). Furthermore, h cannot exceed the maximum capacity, h_{max}, of the system.

The objective is to find h paths that give overall optimal output performance:

$$\min_{\mathbf{A}} \left\| \hat{\mathbf{y}}(k) - y^{Target} \right\| = \min_{\mathbf{A}} \left\| \mathbf{A}\mathbf{T}^{+}\hat{\mathbf{z}} - y^{Target} \right\| \qquad (38)$$

where $\mathbf{A} = \left[\mathbf{a}^T(1) \quad \cdots \quad \mathbf{a}^T(h) \right]^T$

subject to:

$$a_{i,j}(k) \in \{0,1\} \qquad \forall i \in I, j \in J, k \in K \qquad (39)$$

$$\sum_{j=1}^{n_i} a_{i,j}(k) = 1, \qquad \forall i \in I, k \in K \qquad (40)$$

$$\sum_{k=1}^{h} a_{i,j}(k) \le 1, \qquad \forall i \in I, j \in J \qquad (41)$$

where $I = \{1,2,\ldots,f\}$, $J = \{1,2,\ldots,n_i\}$, and $K = \{1,2,\ldots,h\}$.

This optimization is a mixed integer non-linear programming (MINLP) problem.

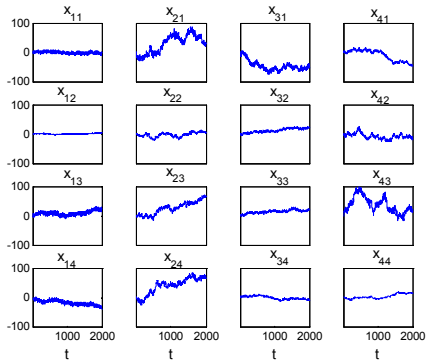

Fig. 4 Disturbance sequences of all contexts in Example 2

Example 2 Consider a process with 4 factors and 4 contexts for each factor where all contexts are assigned IMA behavior, the output target is $y^{target} = 0$ and the maximum run capacity is $h_{max} = 4$. A snapshot of all context behavior is shown in Fig. 4. Fig. 5 shows the results of h=4. The result shows significant improvement on performance. However, output becomes worse after run 1000. By continuously using all tools, some tools may drift far away from their nominal values such that seeking suitable combinations of tools to closely match the target may be impossible. Note that if all tools drift in the same direction (compared to nominal values), path planning may not significantly improve process output.

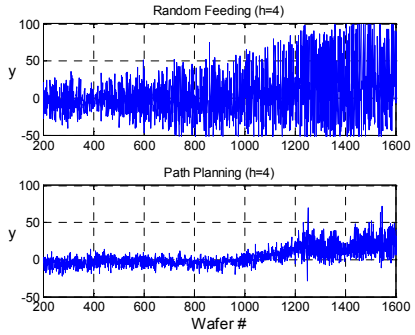

Fig 5. Path planning result of Example 2

4.2 Particle Count Estimation

Particle is an important cause of killer defects (Quirkm and Serda,2001). The particle number on the wafer is counted by optical scanning, and the metrology tool is very time consuming and expensive. We assume that particles on a wafer are accumulated from each process tool. A tool in worse condition will generate more particles on wafers; for example, a leaking pipe may generate many particles on wafers. Thus, with this assumption the context process method can be applied to the process. However, behavior such as particle burst (e.g. sudden, short duration events) (Borden, 1990) are outside the scope of this assumption and consequently cannot be modelled by this method.

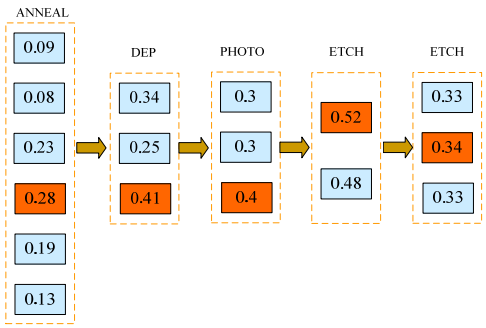

Fig. 6 Process flowchart and tool used frequency

Data are collected during a half year from a real fab in Taiwan which fabricates 300mm wafers. In this study, there are five processes before the particle count measurement: annealing, deposition, photolithography, and two etchings. The process scheme and tool use frequencies during a half year is shown in Fig. 6. The data are collected from the same product constituted over 90% of the process line. There are two types of particle measurement: regular inspection and

random inspection. For regular inspection, 30% of the lots will be sent to metrology tools, and 5 wafers from each lot will be measured. Random inspection is usually undertaken by engineers or operators for diagnosis or other purposes, and the number of measured wafers is also irregular. In order to have a consistent data format, we drop the data from random inspection. Data observations which are outliers or missing values are also neglected.

We consider each lot to be on run (as defined in this work), and use the average particle count of 5 wafers as output for each lot. About 1800 lots are collected, and the first 200 lots are used as a training set. The most frequently used tools from each process are selected as members of the reference path, $\mathbf{p}^{ref}$=[4 3 3 1 2]. Fig. 7 shows the result of one-step-ahead prediction. The trend follows the real measurement closely.

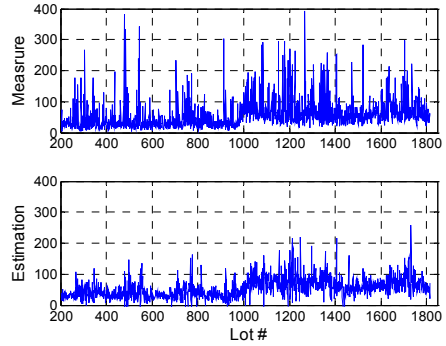

Fig. 7 One-step-ahead estimation of the average particle number

Furthermore, we consider the fact that each etching equipment tool has two chambers, and wafers in a lot will be sent to one or the other with the same probability. Thus, there are total of 5 etching tools with 10 processing chambers, and the number of all contexts becomes 22. For this case, the reference path is selected as $\mathbf{p}^{ref}$=[4 3 3 2 3]. Due to the large number samples, only a part of result is shown in Fig. 8. The result of one-step-ahead prediction also follows the spikes seen in the measurements

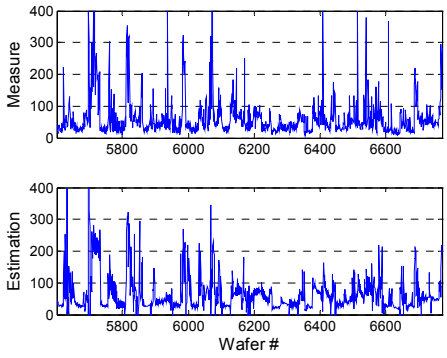

Fig. 8 One-step-ahead estimation of the particle number of wafers

Estimation of the particle number provides the capability for monitoring. Furthermore, the relative status of each tool is also available, and this can be used as a health index of each tool. In Fig. 7, the average particle number rises to a higher value after run 1000. Although analysis on behavior of transformed sates can be made, there is no fault record for all tools in the system data base during the time period for validating our result.

5. CONCLUSIONS

There are many factors that have significant influence on process output in semiconductor manufacturing. The proposed method of context-based state estimation can both successfully estimate relative context status and process output. Although the context system is not observable, some transformed states still can be unbiasedly estimated. The transformed states are straightforward and physically meaningful: a real path is used as reference and other contexts are expressed as differential values relative to the reference path. We also investigated a method for path planning with a guarantee of output performance. However, the path planning problem is more complicated in reality. For example, there may be a different process time for each process or equipment, or tool down time due to maintenance. To deal with these issues, some event scheduling techniques (e.g. Petri net) should be considered.

A context-based process can be treated as a fab-wide framework that can integrate information of product types, tools, etc. A future task will be to integrate context information down to tool-level run-to-run control.

REFERENCES

Borden, P. (1990), The nature of particles generation in vacuum process tools, *IEEE Transaction on Semiconductor Manufacturing*, **3**, No.4.

Edgar, T. F., S. Firth, C. Bode, and V. Martinez (2004) ,Multiproduct Run to Run Control for High Mix Fabs, *2nd AEC/APC Symposium Asia*, Hsunchu, Taiwan

Edward A. Rietman, S. A. Whitlocj, M. Beachy, A. Roy, and T. L. Willingham (2001), A system model for feedback control and analysis of yield: a multistep process model of effective gate length, poly line width, and IV parameters, *IEEE Tran. on Semi. Manuf.*, **14**, No. 1.

Firth, S. (2002), *Just-in-time adaptive disturbance estimation for run-to-run control in semiconductor processes*, PhD Thesis, University of Texas at Austin,

Pasadyn, A. J. and T. F. Edgar (2005), Observability and state estimation for multiple product control in semiconductor manufacturing, *IEEE Tran. on Semi. Manuf*, **18**, No. 4.

Quirkm M. and J. Serda (2001), *Semiconductor manufacturing technology*, Prentice Hall, New Jersey.

Wang, J., Q. P. He, and T. F. Edgar (2007), A General Framework for State Estimation in High-Mix Semiconductor Manufacturing, 2007 *American Control Conference*, New York, July .

Wu, M.F., S. S. Jang, and D. S. H. Wong (2006), Mixed Products Run-to-Run Process Control -An ANCOVA Model Based Control Approach, *IFAC Workshop on Advanced Process Control for Semiconductor Manufacturing*, Singapore, Dec.