Real-Time Statistical Process Control Using Tool Data

Costas J. Spanos, *Member, IEEE,* Hai-Fang Guo, Alan Miller, and Joanne Levine-Parrill

***Abstract*—During the last five years we have witnessed the widespread application of statistical process control in semiconductor manufacturing. As the requirements for process control grow, however, traditional statistical process control applications fall short of their goal. This happens because modern processes are more complex than they used to be. Further, because of the expanding use of the so called "cluster" tools, modern technologies are also less observable than before. Because of these difficulties, we can no longer afford to wait until a malfunction can be detected on a traditional control chart.**

Fortunately, modern semiconductor manufacturing tools can communicate to the outside world a number of their internal parameters, such as throttle valve positions, chamber pressures, temperatures, etc. It is intuitively obvious that equipment malfunctions will manifest themselves first in the values of these internal parameters and much later on the wafer properties. In this paper we describe a process monitoring scheme that takes advantage of such real-time information in order to generate malfunction alarms. This is accomplished with the application of time-series filtering and multivariate statistical process control. This scheme is capable of generating alarms on true real-time basis, while the wafer is still in the processing chamber. Several examples are presented with tool data collected from the SECSII port of single-wafer plasma etchers.

I. Introduction

As INTEGRATED CIRCUITS (ICs) become more complex, the semiconductor manufacturing community is focusing its resources on achieving tight process control over the critical process steps. Many tools and techniques are being used toward this end. Statistical Process Control (SPC) is prominent among them, as it can help in the timely detection of costly process shifts.

Historically, SPC has been used with process measurements in order to uncover equipment and process problems. Such problems are manifested by significant degradation in equipment operation and product quality. To discover this degradation, critical process parameters are monitored using various types of control charts. The measurements consist mainly of in-line readings collected from wafers after the completion of the process step in question.

Although this method is helpful in detecting process drifts, there is significant delay between the occurrence of a drift and the resulting control chart violation. As production volume increases, faster response to process drifts becomes necessary in order to assure high product quality and low cost. In addition, the proliferation of multi-chamber (cluster) tools, makes it even more difficult to collect the necessary in-line measurements. Under these circumstances we must use other types of information for quality control purposes.

Modern semiconductor manufacturing equipment can communicate internal sensor readings over standard RS232 ports using the SECSII protocol. This capability has been recognized as crucial for the diagnosis of equipment failures, and for the improvement of the overall product quality [1]. Unfortunately, in a high volume production facility the monitoring of multiple sensors results in an overload of information. Further, most of the popular SPC strategies cannot be applied to real-time readings, since these readings usually show non-stationary, auto-correlated and cross-correlated variation. A special type of SPC procedure is therefore needed to automate the processing of tool data.

This paper describes the development and the application of a novel SPC method that uses time-series filters [2] and multivariate statistics [3] to analyze internal machine parameters. These parameters are sampled several times per second, and the readings are filtered using a time-series model. The filtered readings are then combined into a single variable with well defined statistical properties [4]. This single statistical variable is calculated every few seconds, and is plotted against formally defined control limits. Real-time misprocessing alarms generated in this manner allow a controller to interrupt faulty runs and prevent any adverse effects on the equipment or the product. These alarms can be used for scheduling preventive maintenance. In the future, these alarms might also be used in conjunction with automated diagnosis routines [5].

This method has been applied on a Lam Research Rainbow single wafer plasma etcher, and on an Applied Materials Precision 5000 cluster tool. The results show that the filtered statistical parameter has successfully responded to several types of process faults, which were introduced in a controlled fashion. The faults included mismatched RF components, different loading factors, gas leaks, and miscalibrated equipment controls. It is noteworthy that none of these faults could have been easily detected by traditional wafer measurements.

Manuscript received January 21, 1992; revised March 26, 1992.

C. Spanos is with the Department of EECS, University of California at Berkeley, Berkeley, CA 94720.

H.-F. Guo was with the Department of EECS, University of California. She is presently with IBM Corporation, San Jose, CA.

A. Miller is with Lam Research, Fremont, CA.

J. Levine-Parrill is with IBM Corporation, East Fishkill, NY.

IEEE Log Number 9202883.

The rest of this paper is structured as follows: Section II presents a brief overview of traditional statistical process control. Section III describes the real-time, multivariate SPC approach, which includes the time series-model and the calculation of Hotelling's T^2 statistic. Experimental results are presented in Section IV along with a brief description of the equipment and the data acquisition tools. Finally, Section V contains a summary and some suggestions for future extensions of this work.

II. Traditional Statistical Process Control

The concept of statistical control of a production sequence was introduced in 1924 by Walter A. Shewhart of the Bell Telephone Laboratories [6]. Today, SPC is understood as a collection of methods whose objective is to improve the quality of a process by reducing the variability of its critical parameters.

A process is said to be in statistical control when, "*through the use of past experience, we can predict, at least within limits, how the process may be expected to vary in the future*" [7]. When a process is in statistical control, there is only natural variation or "background noise" because of mechanisms known as *chance causes*. Sometimes, however, a process can change due to *assignable* causes, such as significant environmental changes, miscalibrations, variability of raw material, or human error. Assignable causes make a process unpredictable and cause it to lose the state of control as defined above. The main purpose of SPC is to detect the presence of an assignable cause so that it can be corrected.

From a statistical point of view, SPC casts the decision-making process as a formal *hypothesis test*. In this context, the *null hypothesis* (H_0) states that the process under consideration is under statistical control, while the *alternative hypothesis* (H_a) states that the process is out of statistical control. To test these hypotheses, a random sample x is selected from the population of interest, and the suitable test statistic is calculated. Typically, we calculate the average of several readings of x, and the resulting statistical score is tested against the limits listed in (1). The range of values that leads to the rejection of a hypothesis is called the *critical region* or the *rejection region*. For the Shewhart $\bar{X}$ chart, the upper and lower (UCL and LCL) limits used to validate H_0 are given next:

$$\begin{aligned} \text{UCL} &= \mu + Z_{\alpha/2}\sigma_{\bar{x}} \\ \text{LCL} &= \mu - Z_{\alpha/2}\sigma_{\bar{x}} \end{aligned} \tag{1}$$

where x is distributed according to the $N(\mu, \sigma^2)$ normal distribution, $\bar{x}$ is the arithmetic average calculated from n samples of x, and $\sigma_{\bar{x}} \equiv \sigma/\sqrt{n}$. Also, $Z_{\alpha/2}$ is the standard normal score which excludes the $\alpha/2$ portion off the high tail of the standard normal distribution. According to this equation, the probability of rejecting H_0 by mistake, an occurrence known as a type I error, is equal to α. Alternatively, accepting H_0 by mistake is known as a type II error. The distribution that illustrates the nature of the $\bar{X}$ chart is shown in Fig. 1.

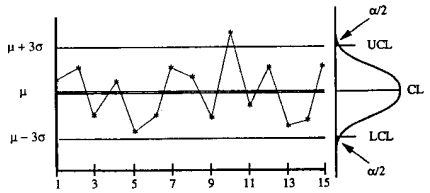

Fig. 1. An $\bar{X}$ control chart and its hypothesis-testing nature.

A popular set of rules developed by Western Electric in the 1950s and known as the *Western Electric Rules*, provides additional ways to generate alarms [8].

At this point, it is important to emphasize that the operation of the $\bar{X}$ chart is based on the model described by (1). This equation implies that all the "good" data must come from the same population, which must follow a normal distribution around a fixed value. In other words, the data must be *Identically, Independently and Normally Distributed*. This is known as the *IIND* assumption and is summarized below:[1]

$$\begin{aligned} x_t &= \mu + a_t \qquad t = 1, 2, \cdots \\ a_t &\sim N(0, \sigma^2). \end{aligned} \tag{2}$$

The *IIND* assumption is essential for the simple control chart. Without it, the chart and its limits would not truly reflect the process. Unfortunately, real time data often violate the *IIND* assumption. In the next chapter we focus on the statistical nature of such data.

III. Real-Time Statistical Process Control

As the volume of production increases, instantaneous detection of process drifts becomes necessary. Most modern equipment have some automated data acquisition capabilities. Unfortunately, traditional statistical process control methods cannot be applied directly on tool data, because most tool-generated data violate the *IIND* assumption. Indeed, in most cases, real-time data are nonstationary, and in addition they are auto-correlated and cross-correlated, even when they originate from a process that is under control.

To accommodate this situation, a novel SPC scheme is developed and applied to several test processes. This scheme employs time-series [2] multivariate statistics [3]. First, time-series models are needed to transform real-time sensor data into *IIND* signals; and a particular multivariate technique, known as the Hotelling's T^2 statistic, is used to combine the *IIND* signals into a single, well behaved statistical variable.

This scheme is capable of generating alarms on true real-time basis, *while the wafer is still in the processing chamber*. In this way, we are able to detect misprocessing before it impacts the product. In this section we describe in some detail this real-time SPC scheme.

[1]The expression $a_t \sim N(0, \sigma^2)$ means that the random variable a_t is distributed accordingly to a normal distribution with zero mean and a variance σ^2.

A. Time-Series Modeling

Readings collected sequentially are rarely independent. It is this lack of independence, for example, that allows the forecasting of daily temperature lows and highs from recent readings and from historical records. Often, the statistical behavior of a time-varying parameter can be described by *time-series* model. The purpose of a time-series model is to capture the dependencies among sequential readings of a variable. Time-series models are often used to *forecast* the value of a future reading from the values of several *past* observations [9], [11].

The statistical behavior of data collected from most modern semiconductor manufacturing equipment can be modelled with the help of a time-series model. The fact that process readings are statistically related to past values can be intuitively understood: Consider, for example, that modern equipment use feedback control on critical parameters, such as temperature or pressure. The sensors in the control loop record the deviation of the parameter from its target value, and, in the next instant, the controller tends to compensate the observed deviation. Thus, a reading higher than a target value is very likely to be followed by a low value and vice versa, leading to an apparent negative autocorrelation between consecutive readings. Conversely, at high sampling rates the monitored parameters are subject to "inertia," leading to an apparent positive autocorrelation between consecutive readings. In general, dependencies among readings collected over time can be described by the following equation:

$$x_t = f(x_{t-1}, x_{t-2}, \cdots, a_{t-1}, a_{t-2}, \cdots) + a_t$$
$$t = 1, 2, \cdots$$
$$a_t \sim N(0, \sigma^2) \tag{3}$$

where x is the signal and a is the *IIND* prediction error of the time series model. In this work, the main goal is to find suitable time-series models to filter real-time data used for statistical process control. The methods used to obtain the models are discussed next. Later we will see how the model can be applied within a practical real-time SPC technique. Next we give a very brief overview of time-series modeling. For an in-depth coverage, the reader should consult the extensive literature on the subject [2], [4], [9], [10], [13], [14].

B. Univariate Box–Jenkins Analysis

In this application we use the univariate *Box–Jenkins* time-series analysis [2]. The assumption behind the univariate analysis is that the time-series behavior of one parameter can be fully explained by using past observations of this parameter. A Box–Jenkins model is also called an ARIMA(p, d, q) model, and it consists of three linear components (or filters) as illustrated in Fig. 2. These components are the *auto-regressive* part of order p, the *integration* part of order d, and the *moving-average* part of order q [2].

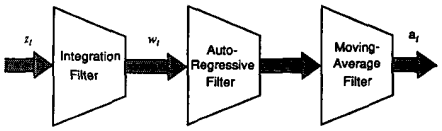

Fig. 2. The three components of the ARIMA model.

The general form of the ARIMA(p, d, q) model is given below:

$$\phi(B) w_t = \theta(B) a_t,$$
$$\phi(B) \equiv 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$$
$$\theta(B) \equiv 1 - \theta_1 B - \theta_2 B^2 - \cdots \theta_q B^q$$
$$w_t = \nabla^d z_t \qquad \text{where } d \geq 0 \tag{4}$$

Difference Operator:

$$\nabla z_t \equiv z_t - z_{t-1} \qquad \nabla^2 z_t \equiv \nabla(\nabla z_t) \cdots$$

Backward Shift Operator:

$$B z_t \equiv z_{t-1} \quad B^2 z_t \equiv z_{t-2} \cdots$$

where z_t is the original reading collected at time t, w_t is the respective differentiated signal, and a_t is the *IIND* residual. Below we explain the function of each of the three components of the ARIMA model.

The first part of the ARIMA model is the *integration* component. This part is necessary because a condition for fitting the autoregressive and moving-average parts of the model, is that the signal must be stationary. This means that the mean, variance and autocorrelation functions of the time-series must be time invariant. The *integration* component of the ARIMA model is used to convert a nonstationary signal to a stationary one. Simple or higher-order differentiation can be used to achieve a time-invariant mean.[2]

The second part of the ARIMA model is the *autoregressive* (AR) part, which is needed in order to describe the dependency of the current observation on previous observations. This is done through the autoregressive coefficients ϕ_i.

The third part of the ARIMA model is the *moving-average* (MA) part, which describes the dependency of the current observation on previous forecasting errors (also known as *random shocks*), by means of the moving-average coefficients θ_i.

Occasionally, the original data show seasonal periodic patterns. These patterns can be modeled by creating ARIMA models for the seasonal variation as well as for the individual samples. The composite model is known as a *Seasonal* ARIMA model or SARIMA(p, d, q) $\times$ (P, D, Q)$_s$, where p is the number of significant autocorrelations, d is the number of differentiations, q is the number of significant moving average terms within each season,

[2]Taking the log or the square root of the data might be necessary in order to produce a constant variance.

and P, D, Q are the autocorrelations, differentiations and moving average terms, taken across seasons of duration s [14]. The complete SARIMA$(p, d, q) \times (P, D, Q)_s$ model is expressed by (5):

$$\phi(B)\Phi(B^S)w_t = \theta(B)\Theta(B^S)a_t$$
$$w_t = \nabla_s^D(\nabla^d z_t) \tag{5}$$

A model can be obtained from the collected data when the process is under control; in this way the model describes the "good" process. Once a model has been developed, it can be used to *forecast* (or predict) each new value. The difference between the forecast value and the actual value is the forecasting error, or residual. The residual is by definition, an *IIND* variable:[3]

$$a_t = z_t - \hat{z}_t \sim N(0, \sigma^2) \tag{6}$$

C. Creating Box–Jenkins Models

To obtain a useful ARIMA model, Box and Jenkins proposed a three-step procedure [9]. This procedure is illustrated in Fig. 3. Two devices are used to select the ARIMA models: These are the *discrete autocorrelation function* (*acf*) and the *discrete partial autocorrelation function* (*pacf*). The *acf* and the *pacf* are calculated from the properly differentiated signal and are compared with the theoretical *acf* and *pacf* patterns from known model structures.

To further explain the *acf* we need to talk about the *autocorrelation* coefficient. The autocorrelation coefficient describes the statistical dependence between two readings collected at different times. The auto-correlation coefficient takes values in the range from -1 to $+1$. A zero value will be obtained when the observation of interest is independent from other observations, while a value of $+1$ indicates complete synergistic dependence. The value of -1 indicates complete antisynergistic dependence. The following equation defines the auto-correlation coefficient between all pairs of n readings that have been collected k observation time intervals apart from each other. The autocorrelation coefficient is calculated from n consecutive observations, by using the $(n - k)$ pairs of observations separated by k observation intervals. Expressed as a function of the integer k, the estimated *acf* is given by (7):

$$r_k = \frac{\sum_{t=1}^{n-k}(z_t - \bar{z})(z_{t+k} - \bar{z})}{\sum_{t=1}^{n}(z_t - \bar{z})^2} \qquad k = 1, 2, \cdots \tag{7}$$

The *partial autocorrelation function* (*pacf*) also gives a measure of dependence across pairs of readings, only now this dependence is given *after the dependence of the intervening readings has been accounted for*. This is accomplished by fitting the following regression equation:

$$z_{t+k} = \phi_{k1}z_{t+k-1} + \phi_{k2}z_{t+k-2} + \phi_{k3}z_{t+k-3} + \cdots + \phi_{kk}z_t + u_{t+1} \tag{8}$$

where this equation is fitted on the signal multiple times, with increasing value of k starting from $k = 1$. The *pacf* is the series $\phi_{11}, \phi_{22}, \cdots, \phi_{kk}$ which is usually displayed as a discrete function of k.

Both the *acf* and the *pacf* are needed in order to infer the structure of the best fitting ARIMA model. The inference of the best model structure is usually done by trial and error, using the *acf* and *pacf* of the original signal and its residuals for guidance [9]. After the structure of the model is inferred and its coefficients extracted, the *acf* and the *pacf* of the residuals are used to check the adequacy of the selected model. The process terminates when a satisfactory model is obtained. This interactive sequence is illustrated in Fig. 3. Attempts to automate this procedure have also been reported in the literature [10].

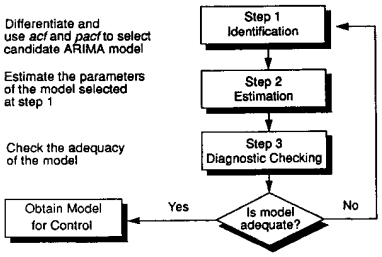

Fig. 3. The 3-step procedure for ARIMA modeling.

D. Hotelling's T^2 Statistic

A piece of equipment will, in general, be monitored through a number of sensor signals. Using the appropriate time-series model, each signal is filtered down to its IIND residual. Assuming that the time-series models have been properly built and that the machine is under control, each of these residuals will be an IIND random number.

This means that one could use a simple Shewhart control chart to monitor each residual. However, since the signals are originating from the same physical process, their residuals will probably be statistically correlated and using them in separate control charts can be misleading. In fact, it can be shown that as the number of correlated variables increases, the probability of generating false alarms from a control procedure that uses a large number of separate charts grows significantly [6]. This is because treating correlated signals separately leads to the underestimation of the probability of generating false alarms and the probability of not detecting a malfunction. Further, the information content of multiple, concurrent real-time control charts will undoubtedly overwhelm the human operator.

The function of Hotelling's T^2 statistic is to combine several cross-correlated variables into a single statistical

[3]The hat (ˆ) signifies a value predicted by the model.

score. This number is simply the square of the maximum possible univariate *student-t* score computed from any linear combination of the various outcome measures [3]. This score is calculated from the p correlated residuals as follows:[4]

$$T^2 = n(\bar{\mathbf{a}} - \mathbf{0})^T \mathbf{S}^{-1} (\bar{\mathbf{a}} - \mathbf{0})$$

where group mean $\bar{\mathbf{a}}^T = [\bar{a}_1 \cdots \bar{a}_p]$

nominal value of residuals $\mathbf{0}^T = [0 \cdots 0]$

$$\text{variance-covariance matrix } \mathbf{S} = \begin{bmatrix} s_1^2 & \cdots & s_{1p} \\ \cdots & \cdots & \cdots \\ s_{p1} & \cdots & s_p^2 \end{bmatrix} \tag{9}$$

where, in order to further ascertain that the entries in this formula are normally distributed, it is customary to use averages calculated over small, consecutive groups of size n for each residual. Some discussion is necessary concerning the estimation of the variance-covariance matrix $\mathbf{S}$. First, the diagonal elements in $\mathbf{S}$ are calculated as the average s value for each of the m groups of size n:

$$s_{jk}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{a}_{ijk} - \bar{\mathbf{a}}_{jk})^2 \qquad \begin{matrix} k = 1, 2, \cdots, m \\ j = 1, 2, \cdots, p \end{matrix} \tag{10}$$

The off-diagonal terms are estimators of the covariances and are calculated as follows:

$$s_{jhk}^2 = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{a}_{ijk} - \bar{\mathbf{a}}_{jk}) \cdot (\mathbf{a}_{ihk} - \bar{\mathbf{a}}_{hk}) \qquad \begin{matrix} k = 1, 2, \cdots, m \\ j = 1, 2, \cdots, p \qquad j \neq h. \\ h = 1, 2, \cdots, p \end{matrix} \tag{11}$$

Finally, the actual elements of the variance-covariance matrix $\mathbf{S}$ are calculated by averaging over the m groups the values found in (10) and (11):

$$s_j^2 = \frac{1}{m} \sum_{k=1}^{m} s_{jk}^2 \qquad j = 1, 2, \cdots, p$$

$$s_{jh} = \frac{1}{m} \sum_{k=1}^{m} s_{jhk}^2 \qquad j \neq h. \tag{12}$$

The T^2 score is sensitive to a shift in the mean value of one or more of the variables. This score can be used in a one-sided control chart, whose limit is set according to the number of variables, the sample size and the acceptable false alarm rate. The control limit of the T^2 statistic is related to the cumulative F distribution at level α:

$$T_{\alpha,p,m-1}^2 \sim \frac{p(m-1)}{m-p} F_{\alpha,p,m-p} \tag{13}$$

which, assuming that the number of measurements is high, can be approximated by a simple chi-square distribution with p degrees of freedom:

$$T_{\alpha,p,m-1}^2 \simeq \chi_{\alpha,p}^2. \tag{14}$$

Of course, the way the T^2 score has been defined here makes it the optimum statistic for controlling "unstructured" mean shifts, i.e., shifts that might happen in any direction within the p-dimensional space. This property is very useful in the context of our application, since shifts can indeed happen in any direction. When, however, particular, known directions are more susceptible to a shift, better statistics (such as the principal components [7] or the Z-scores [8]) might be utilized. In addition, although this property is not being investigated in this paper, the T^2 statistic can be extended to guard against a shift in the *variance* of the monitored parameter [7].

Another potential problem might arise from the fact that the T^2 statistic is not geared towards identifying a shift in the variance-covariance matrix and, in fact, will confound such a shift with a shift in the mean vector. Because of this the $\mathbf{S}$ matrix has to be re-calculated every time a new time-series model is calculated.

Other multivariate control methods are, of course, available. Most, however, suffer from the significant disadvantage of requiring the monitoring of multiple control charts. Such methods might prove advantageous for analyzing an alarm for diagnostic purposes and will most probably be the subject of future work by the authors. For routine monitoring applications, however, the simplicity of having to maintain a single control chart makes the T^2 statistic a very attractive proposition.

E. Implementation of the Real-Time SPC Scheme

In summary, the real-time SPC scheme takes multiple sensor data that are auto-correlated and cross-correlated, and then feeds them into individual time-series filters that produce multiple, cross-correlated IIND residuals. Hotelling's T^2 equations combine the cross-correlated residuals into a single real-time alarm signal. This sequence is illustrated in Fig. 4.

This alarm signal can be used either as a passive SPC alarm, or it can initiate a diagnostic procedure [5]. A software package has been developed to implement this real-time SPC scheme. It includes four modules: data manipulation, ARIMA filtering, Hotelling's T^2 calculation, and alarm generation. These operations were initially implemented in the commercial statistical packages SAS™ [16] and RS/1™ [17]. Recently, we have completed independent implementations for Unix and DOS environments. Coupled with a SECSII server, either of these implementations is capable of actual real-time operation. The most recent implementation imports ARIMA models that are

[4] In this paper we employ bold-faced symbols to represent non-scalar quantities such as arrays and matrices. All arrays are columns, unless used with the superscript (^{T}) which symbolizes transposition.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.