AUTOMATIC TIME-SERIES MODEL GENERATION FOR REAL-TIME STATISTICAL PROCESS CONTROL

by

Hao-Cheng Liu

Memorandum No. UCB/ERL M93/45

8 June 1993

COVER PAGE

AUTOMATIC TIME-SERIES MODEL GENERATION FOR REAL-TIME STATISTICAL PROCESS CONTROL

by

Hao-Cheng Liu

Memorandum No. UCB/ERL M93/45

8 June 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TITLE PAGE

AUTOMATIC TIME-SERIES MODEL GENERATION FOR REAL-TIME STATISTICAL PROCESS CONTROL

by

Hao-Cheng Liu

Memorandum No. UCB/ERL M93/45

8 June 1993

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

To my father, my mother and Stephanie
for their love and support.

Acknowledgment

I would like to express my deep appreciation and gratitude to my research advisor, Dr. Costas J. Spanos, for his support and guidance through my graduate studies and the latter portion of my undergraduate studies. I also thank Dr. John A. Rice for being a part of the project report committee and for his valuable insights in time-series modeling.

Special thanks are due to two colleagues in the BCAM group: Mr. Eddie Wen, for his work in developing the BCAM Real-Time SPC interface, and Ms. Sherry Lee, for sharing her knowledge and expertise and providing insightful feedbacks and discussions. Their help and friendship are greatly appreciated.

I would also like to extend my gratitude to the rest of the BCAM group for making my graduate experience an enjoyable one. They are Mr. Eric Boskin, Mr. Eric Braun, Mr. Raymond Chen, Mr. Sean Cunningham, Ms. Zeina Daoud, Mr. Kwan Kim, Mr. Sovarong Leang, Ms. Pamela Tsai, and Mr. Crid Yu. Special thanks are also extended to past BCAM members Mr. Bart Bombay, Ms. Haifang Guo, Ms. Lauren Massa-Lochridge, Mr. Tom Luan, Dr. Gary May, and Mr. John Thompson.

I am also grateful to the Semiconductor Research Corporation and International Business Machines Corporation for sponsoring this work.

AUTOMATIC TIME-SERIES MODEL GENERATION FOR REAL-TIME STATISTICAL PROCESS CONTROL

by

Hao-Cheng Liu

ABSTRACT

As integrated circuit designs become more complex, in compliance with *Moore's Law*, assuring the production quality of these complex integrated circuits becomes increasingly difficult. Consequently, semiconductor manufacturers must focus on achieving tighter real-time process control in order to obtain justifiable production yields as well as sustain profitability in an increasingly competitive marketplace.

Traditionally, equipment and process faults are being discovered by "in-line" measurements done between process steps. However, due to an increased pressure to produce of a highly diverse product mixture in shorter cycle times, equipment and process faults must be detected in real-time. However, because real-time process control requires the analysis of real-time equipment sensor data, traditional statistical process control (SPC) techniques [1] cannot be readily applied to the sensor data due to their non-stationary, auto-correlated and cross-correlated characteristics.

The Berkeley Computer-Aided Manufacturing (BCAM) Real-Time SPC system utilizes econometric time-series models [2] in order to filter real-time readings of any existing autocorrelations. In addition, multivariate statistics, in particular, the Hotelling's T^2 statistic [3], are then used in order to combine the various cross-correlated signals into a single statistical score. This T^2 statistic is monitored with a single-sided control chart for real-time SPC [4].

The objective of this project was to develop and implement an algorithm for automating the time-series model generation process for real-time SPC. Furthermore, modifications must be made to the real-time SPC scheme in order to accomodate the batch nature of single-wafer processing operations. As a result, the BCAM Real-Time SPC scheme has been modified, and an automatic time-series model generator has been developed. The model generator has demonstrated success in generating useful time-series models for real-time sensor data filtering. Furthermore, the modified SPC scheme, which involves generating separate T^2 statistics for detecting *within-wafer* and *wafer-to-wafer* faults, has shown to be superior in detecting processing faults than the originally proposed methodology.

Signature: ____________________

Professor Costas J. Spanos
Committee Chairman

Table of Contents

Chapter 1 Introduction

1.1 Background

As technology continues to improve, integrated circuit designs become increasingly complex. Consequently, semiconductor manufacturers must focus on achieving tighter real-time control over critical manufacturing steps in order to obtain justifiable production yields and sustain profitability. Furthermore, process control becomes even more critical as the quality assurance testing of these complex integrated circuits becomes increasingly difficult.

Currently, various statistical process control (SPC) techniques are being used in the industry in order to detect equipment or process faults that might be detrimental to the product. The most common SPC approach is to use in-line data measured at the end of each process step and to place this data on a traditional Shewhart Control Chart. This of course assumes that the data is identically, independently and normally distributed (IIND) around a constant mean μ with a constant standard deviation σ [1].

However, as semiconductor manufacturing processes become increasingly complex and as production volume increases, it becomes imperative that equipment and process faults be detected online (during the wafer run) instead of in-line (after the wafer run). Therefore equipment manufacturers begin to realize the importance of building machines that are capable of monitoring sensor data on a real-time basis. These real-time sensor readings, however, cannot be placed directly on a traditional control chart because of their non-stationary, auto-correlated and cross-correlated characteristics.

Time-series models can be used to filter non-stationary and auto-correlated sensor signals. The residuals coming out of these filters should be IIND. If the residuals are uncorrelated, then each can be placed on a traditional Shewhart Control Chart.

The Berkeley Computer-Aided Manufacturing (BCAM) Real-Time SPC module utilizes Seasonal Autoregressive Integrated Moving Average (SARIMA) time-series models [2] for the filtration of the real-time sensor data. Because the filtered sensor data tend to be highly cross-correlated, multivariate statistics, in particular, the Hotelling's T^2 statistic [3], is used in order to combine the various cross-correlated signal residuals into a single statistical score. This T^2 statistic is then placed on a single-sided control chart for real-time SPC [4].

1.2 Motivation

Currently, ARIMA time-series models for the BCAM Real-Time SPC module, as well as for other applications, are generated interactively using standard statistical analysis tools. This procedure is time-consuming and requires specialized skills in time-series statistics. An automated time-series model generator will make the BCAM Real-Time SPC module, as well as several other advanced computer-aided manufacturing applications, more robust and practical.

Furthermore, there is a realization that SARIMA models, although used successfully, may not be ideal, or proper, for modeling semiconductor equipment sensor signals. These signals usually have only a small time segment of useful information from each wafer for SPC purposes. When one concatenates these small segments of data, the resulting time series will no longer be continuous. However, the ARIMA time-series model can be modified so that separate models can be generated for detecting within-wafer as well as wafer-to-wafer time-series patterns, thus accommodating for the lack of continuity in the concatenated sensor data. This modification must be built in to the proposed time-series model generator in order to make the BCAM Real-Time SPC complete.

A modified real-time SPC scheme with automatic time-series model generation has been developed and applied on the Lam Research Rainbow single-wafer plasma etcher.

The model generation algorithm has demonstrated success in generating useful time-series models for filtering the real-time sensor data. Furthermore, the modified real-time SPC scheme, which involves generating separate T^2 statistics for detecting *within-wafer* and *wafer-to-wafer* faults, has proven to be superior in detecting processing faults than the originally proposed methodology.

1.3 Organization

The BCAM Real-Time SPC scheme will be reviewed in Chapter 2. Chapter 3 will introduce the readers to ARIMA time-series modeling, as well as to the theory and algorithm for generating these models automatically. Modifications to the BCAM Real-Time SPC system will be discussed in Chapter 4, along with the necessary modifications to the automatic time-series model generation algorithm. Chapter 5 contains details in regards to the implementation of the new BCAM Real-Time SPC system. Some experimental results will then be presented in Chapter 6. The report will conclude in Chapter 7 with a discussion about the effectiveness of the scheme and a proposal of future work.

Chapter 2 The BCAM Real-Time SPC

2.1 Overview

Although the most popular methods for SPC involve the use of the traditional Shewhart control chart or the cumulative sum (CUSUM) chart, these methods cannot be readily applied to rapid, real-time readings. This is because these methods assume that the data being applied to these control charts are identically, independently and normally distributed (IIND), which typically is not the case with rapidly collected, real-time equipment sensor data.

The Berkeley Computer-Aided Manufacturing (BCAM) Real-Time SPC scheme [4] attempts to apply seasonal econometric time-series models, in particular, the seasonal ARIMA time-series models, as filters for real-time equipment sensor data. This eliminates any non-stationarity and autocorrelations from the machine data. Figure 1 shows how real-time readings can be filtered to IIND residuals for some selected sensor signals from the Lam Research Rainbow plasma etcher.

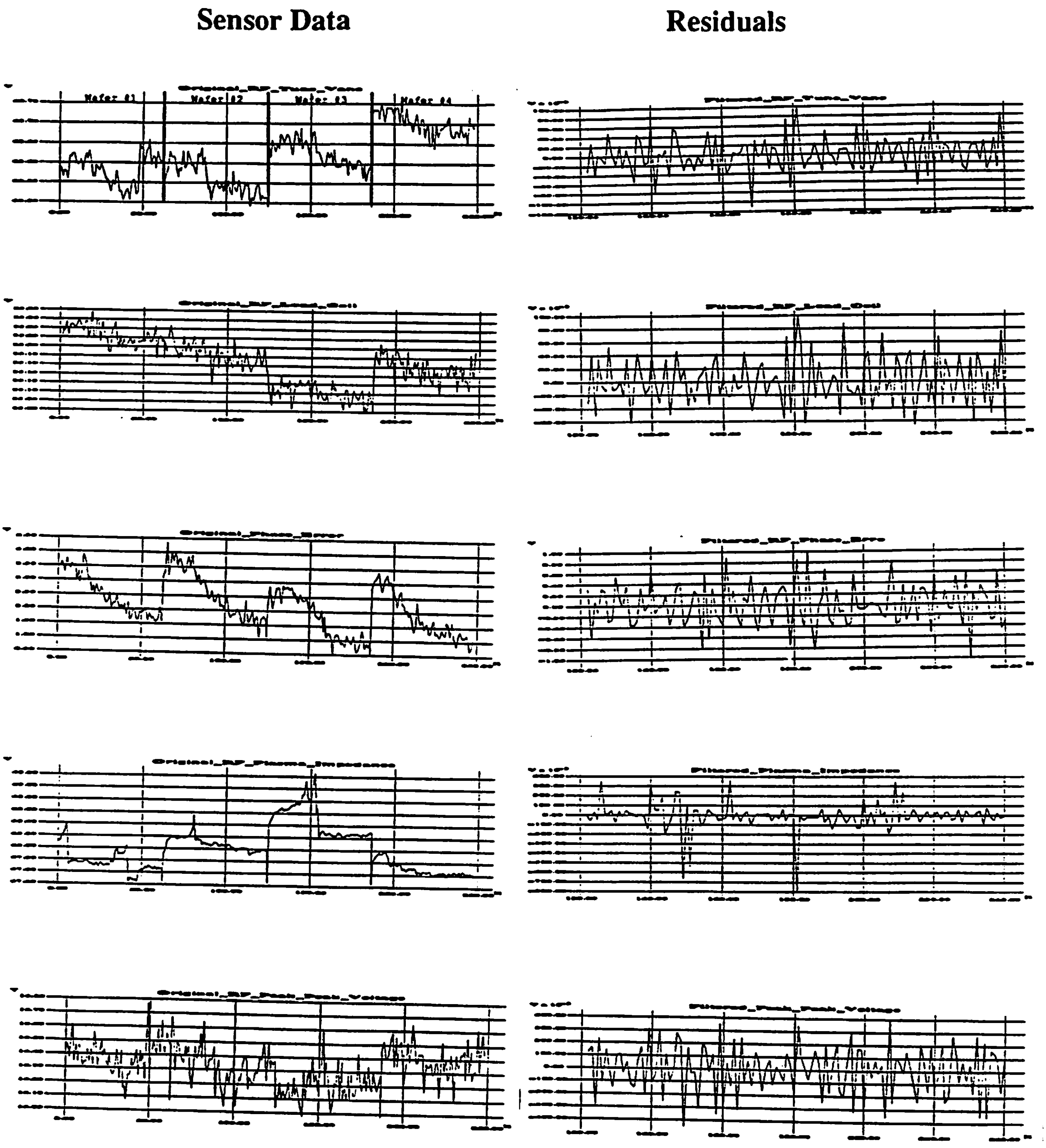


Figure 1. Selected real-time sensor data and corresponding IIND residuals from the Lam Research Rainbow plasma etcher [4].

In the case of multivariate control, there is a high likelihood of cross-correlations existing among the various parameters being monitored. The BCAM Real-Time SPC scheme utilizes multivariate statistics, in particular, the Hotelling's T^2 statistic [3], in order to combine the multiple IIND residuals into a single, well-behaved statistical score, thus accounting for any cross-correlations that might exist. Figure 2 shows the cross-correlation that exists between two selected sensor signals from the Lam Research Rainbow.

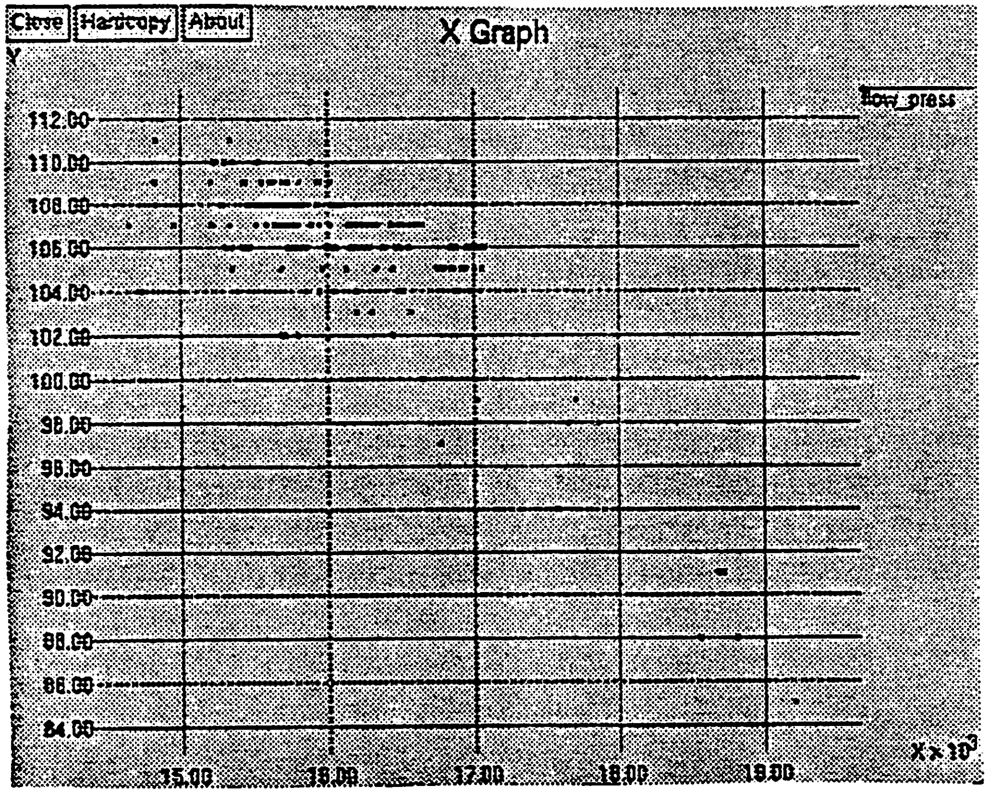


Figure 2. Sample cross-correlated data: chamber pressure plotted against Helium gas flow [4].

The T^2 statistic is then plotted in on a single-sided control chart for real-time control purposes. A T^2 control chart is shown in Figure 3.

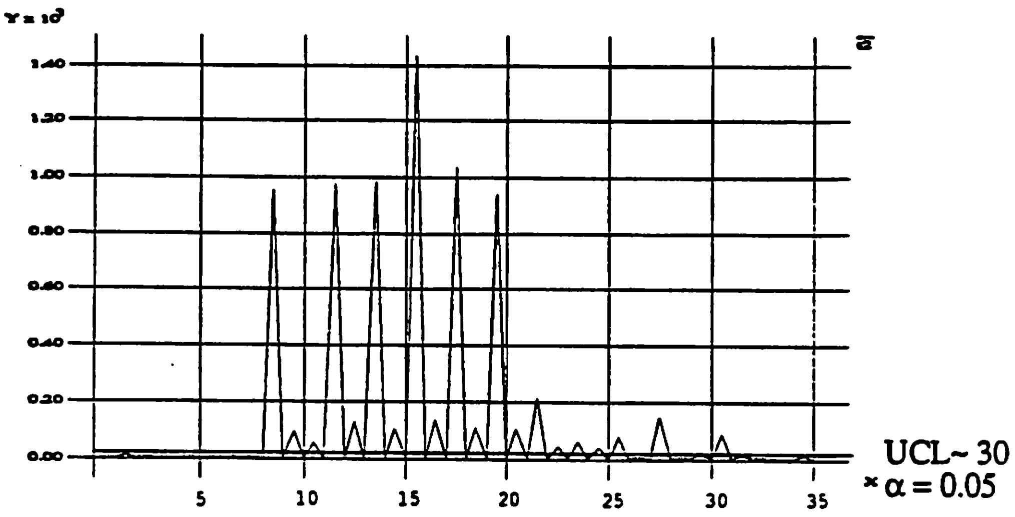


Figure 3. Sample T^2 control chart [4].

The BCAM Real-Time SPC methodology can be summarized by Figure 4.

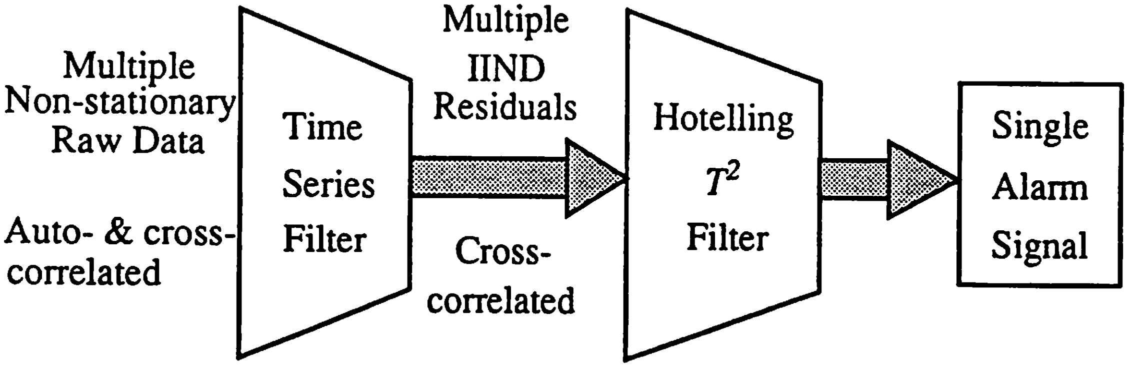


Figure 4. Summary of the BCAM Real-Time SPC scheme [4].

2.2 Problems

A couple of problems exist in regards to the real-time SPC scheme discussed in section 2.1. First, one cannot readily apply seasonal econometric time-series models to the real-time equipment sensor data due to its lack of continuity; and second, an algorithm for automating the time-series model generation process must be developed so that filters can generated real time in a multi-product production environment.

The BCAM Real-Time SPC system processes the real-time equipment sensor data by monitoring only a fixed *step length* of the data from the critical step of each wafer processed with an appropriate *delay* applied. The delay is necessary due to the instability of the sensor data at the beginning of each processing step. An illustration of the real-time equipment sensor data monitoring and delay analysis process is shown in Figure 5.

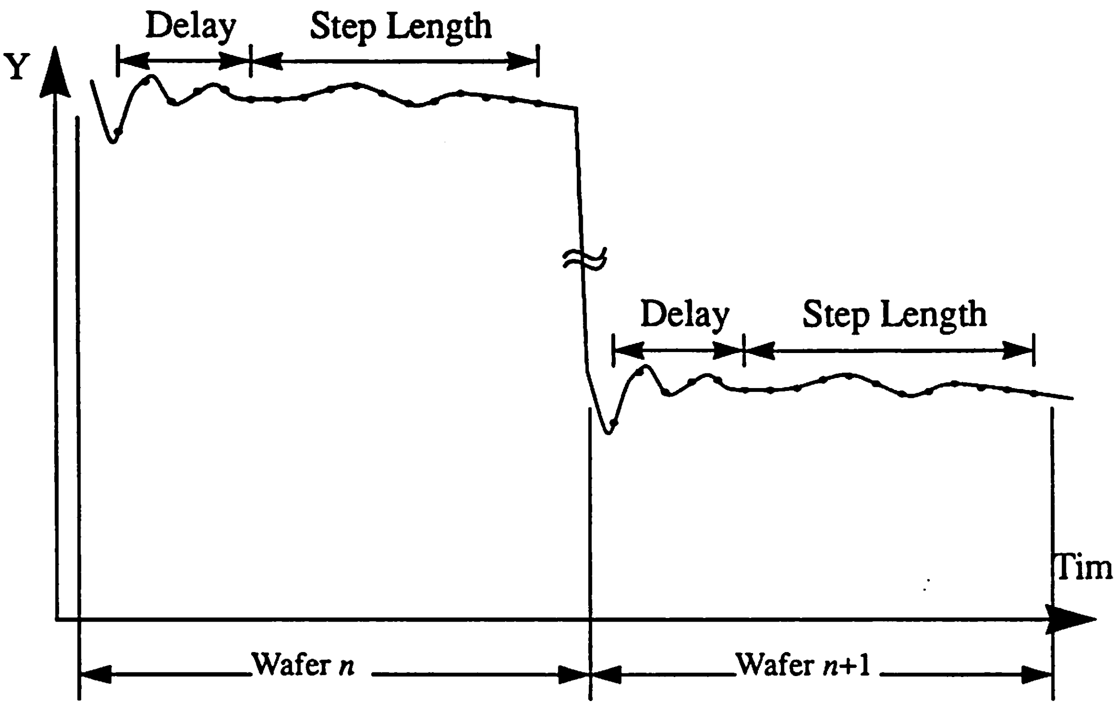


Figure 5. Illustration of the real-time equipment sensor data monitoring and delay analysis process

It is obvious that the real-time equipment sensor data cannot be modelled with a SARIMA time-series model, since the last monitored point for wafer *n* and the first point for wafer *n*+1 are separated by a long unmonitored idle period. Because of this, the real-time sensor data fails the assumption of continuous seasons or periods. Therefore, the data series used in the BCAM Real-Time SPC scheme lack the continuity needed for SARIMA modeling.

In contrast, the one-year periods in seasonal economic data need not be fixed to the monthly data from within a calendar year; it can be any set of twelve continuous monthly data points regardless of whether all twelve points fall within one particular calendar year. Furthermore, the points in an economic data series tend to be continuous through time. Therefore, although it may be ideal to use SARIMA time-series models for forecasting economic data, it must be modified for application to the BCAM Real-Time SPC module. See Figure 6 for a comparison of a continuous economic time-series data and a concatenated non-continuous equipment sensor time-series data.

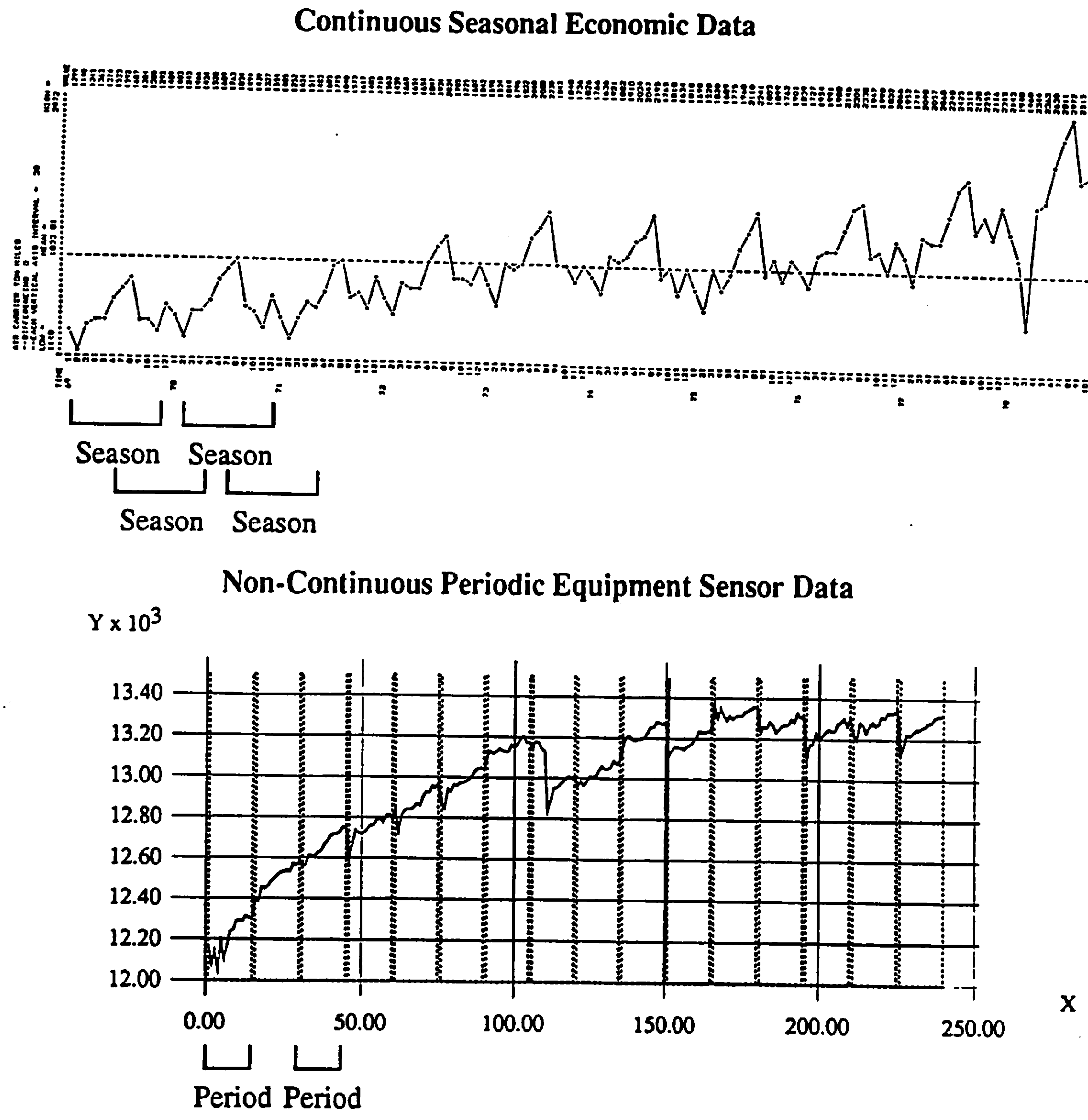


Figure 6. A continuous economic time-series data vs. a non-continuous equipment sensor time-series data [2].

In addition, the modified time-series models must be generated automatically for practical use in the BCAM Real-Time SPC. This is because in a production environment, a variety of products are run through each piece of equipment in any given time. This change in product lines might result in shifts and changes in the machine sensor readings, thus requiring that new models be generated in between runs. Furthermore, the current method of generating time-series models is time-consuming and requires specialized skills in time-series statistics, as the models are generated interactively using standard statistical analysis tools. Thus a methodology for automating the model generation process must be sought. Such a methodology has been developed and is discussed in the next chapter.

Chapter 3 Automatic Time-Series Model Generation

3.1 Background

Linear models can be, and have been, used in order to "forecast" future readings of a time series as a function of past readings and past forecast errors. This method of using linear models for time-series forecasting is illustrated in Figure 7, where observation w_t is forecasted based on past observation w_{t-i} with a forecast error of α_t using a simple autoregressive model.

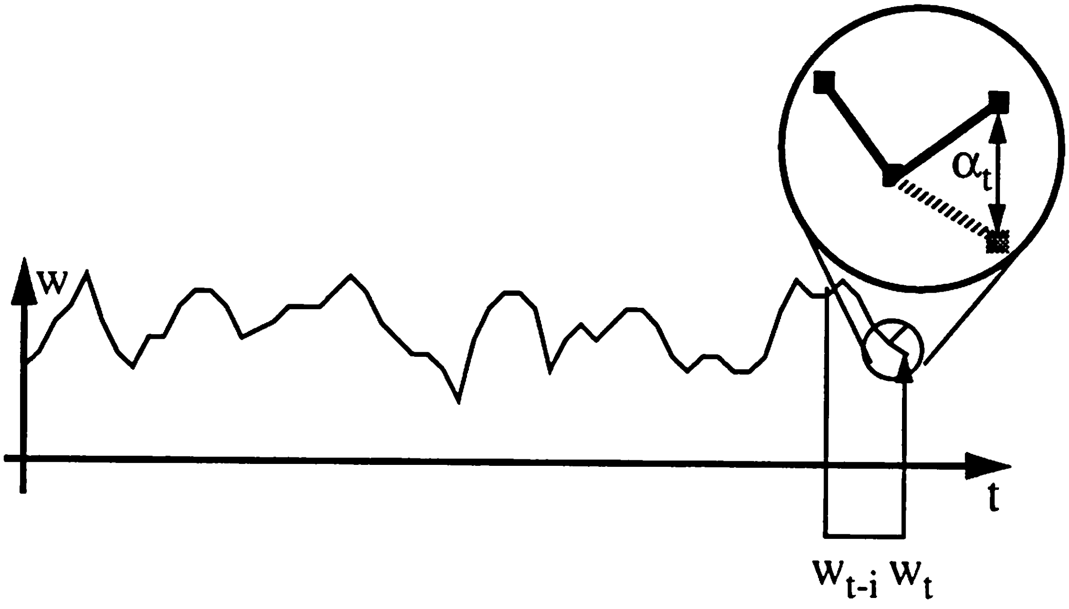


Figure 7. Illustration of time-series modeling.

This method of time-series forecasting applies only to "stationary" data series. A stationary time series has a mean, variance, and autocorrelation "structure" that are essentially constant through time. (The autocorrelation function, which is a way of measuring how the observations within a single data series are related to each other, will be used to determine the autocorrelation structure of a time series.) Non-stationary sequences are often differenced in order to achieve approximate stationarity.

An Autoregressive Integrated Moving Average (ARIMA) model is an algebraic statement showing how a time-series variable (z_t) is related to its own past values (z_{t-1}, z_{t-2}, z_{t-3}, ...), i.e. autoregression, and its past forecast errors (a_{t-1}, a_{t-2}, a_{t-3}, ...), i.e. moving average [2]:

$$\hat{w}_t = \sum_{i=1}^{p} \phi_i w_{t-i} - \sum_{j=0}^{q} \theta_j a_{t-j}$$

$$\text{where} \quad w_t = \nabla^d z_t \tag{1}$$

∇^d : *d*th order of differencing

$$\text{where} \qquad \nabla^1 z_t = z_t - z_{t-1}, \nabla^2 z_t = \nabla(\nabla z_t), \ldots$$

This is the form of the ARIMA model that is used for Real-Time SPC.

One typically generates ARIMA time-series models interactively by first identifying appropriate models, estimating the model parameters, and checking the models for adequacy. This is time-consuming and requires significant skills in using standard statistical tools.

The ARIMA model generation process can be automated if the “integration”, “autoregression” and “moving average” components can be separated and solved for separately. This is typically done by first determining the differencing order necessary to ensure that the data series is stationary, then determining the autoregressive order and parameters using what is known as the *modified Yule-Walker equations*, and finally applying some appropriate technique to find the moving average order and parameters.

3.2 Stationarity and Time-Series Integration

ARMA modeling applies only for a stationary time series. By definition, a series is stationary if all the statistical moments of the series are constant through time. However, if the data series is not stationary, its statistics, and perhaps its mean will shift through time. Therefore, it would be expected that the estimated ACF for this series will drop slowly toward zero.

In order to determine whether a time series is stationary and if differencing is necessary, one looks at the estimated autocorrelation functions (ACFs) of the time series. An autocorrelation function, again, is a way of measuring how the observations within a single data series are related as a function of the time elapsed between the readings. The autocorrelation function at lag k is defined as follows [2]:

$$r_k = \frac{\sum_{t=1}^{n-k} \tilde{z}_t \tilde{z}_{t+k}}{\sum_{t=1}^{n} (\tilde{z}_t)^2} \qquad (2)$$

where $\tilde{z}_t = z_t - \bar{z}$

and n = data count

An example of the autocorrelation function (ACF) plot is shown below in Figure 8.

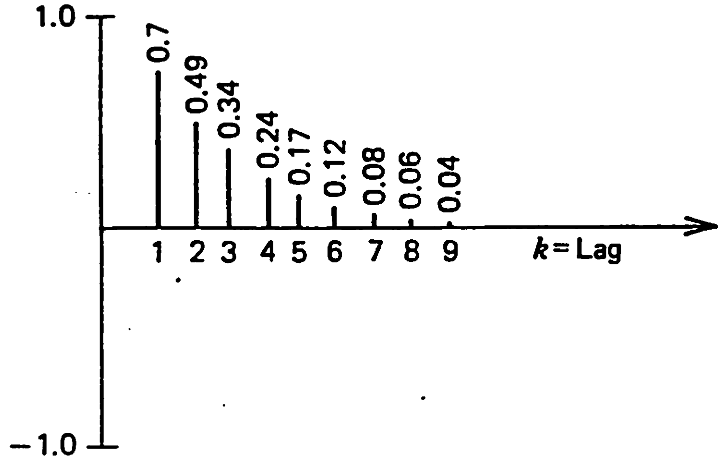
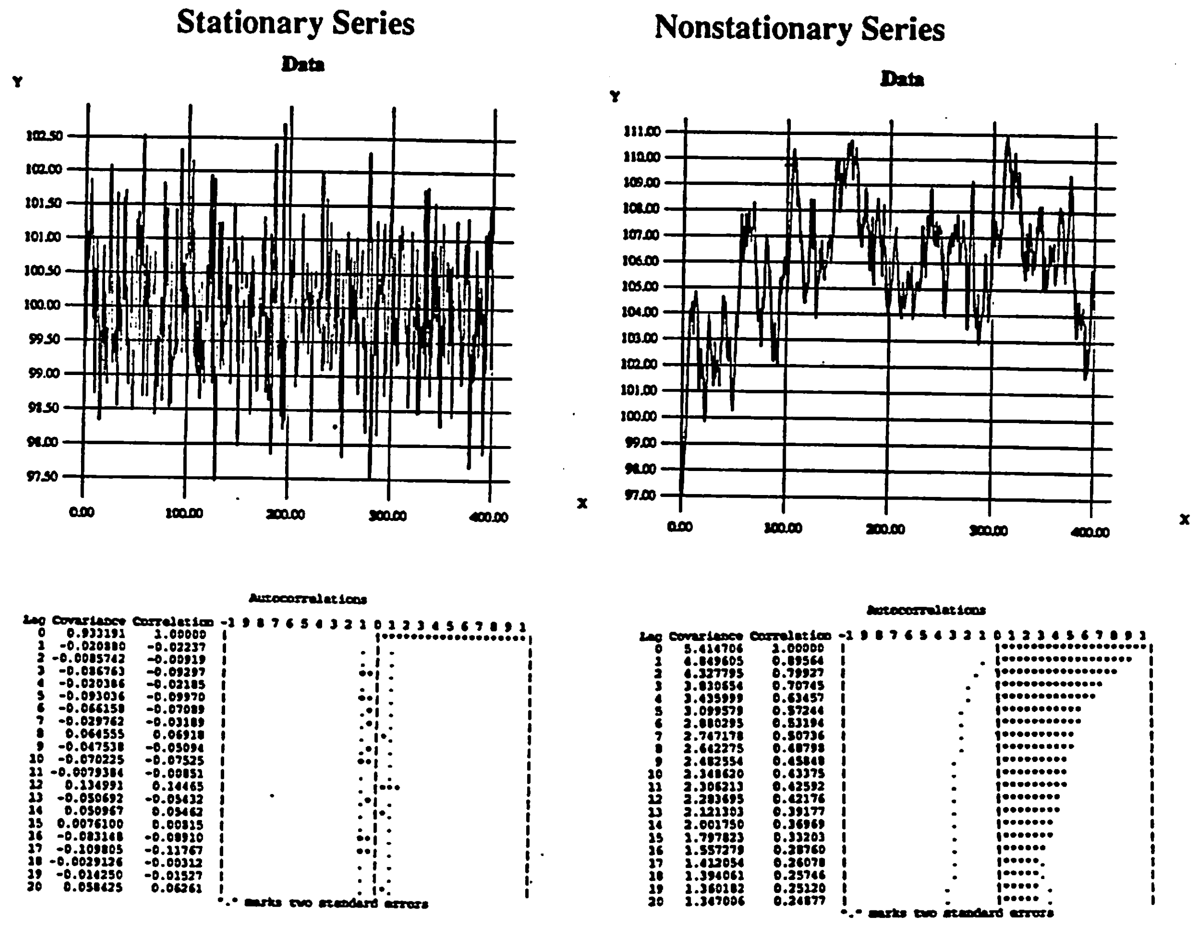


Figure 8. An example of the autocorrelation function (ACF) plot.

If the estimated autocorrelations have absolute t-values of greater than roughly 1.6 (for a 90% confidence in estimation) for the first five to seven lags, this is an indication that the series may have a nonstationary mean and may need to be differenced [2]. An example comparing the ACF plot of a stationary series and that of a nonstationary series is shown in Figure 9.



```
                                   Autocorrelations

Lag Covariance Correlation -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1
  0   0.933191    1.00000  |                    |********************|
  1  -0.020880   -0.02237  |                  . | .                  |
  2 -0.0085742   -0.00919  |                  . | .                  |
  3  -0.086763   -0.09297  |                  **| .                  |
  4  -0.020386   -0.02185  |                  . | .                  |
  5  -0.093036   -0.09970  |                  **| .                  |
  6  -0.066158   -0.07089  |                  .*| .                  |
  7  -0.029762   -0.03189  |                  .*| .                  |
  8   0.064555    0.06918  |                  . |*.                  |
  9  -0.047538   -0.05094  |                  .*| .                  |
 10  -0.070225   -0.07525  |                  **| .                  |
 11 -0.0079384   -0.00851  |                  . | .                  |
 12   0.134991    0.14465  |                  . |***                 |
 13  -0.050692   -0.05432  |                  .*| .                  |
 14   0.050967    0.05462  |                  . |*.                  |
 15  0.0076100    0.00815  |                  . | .                  |
 16  -0.083148   -0.08920  |                  **| .                  |
 17  -0.109805   -0.11767  |                  **| .                  |
 18 -0.0029126   -0.00312  |                  . | .                  |
 19  -0.014250   -0.01527  |                  . | .                  |
 20   0.058425    0.06261  |                  . |*.                  |
                           "." marks two standard errors
```

```
                                   Autocorrelations

Lag Covariance Correlation -1 9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9 1
  0   5.414706    1.00000  |                    |********************|
  1   4.849605    0.89564  |                  . |******************  |
  2   4.327795    0.79927  |                 .  |****************    |
  3   3.830654    0.70745  |                .   |**************      |
  4   3.435999    0.63457  |                .   |*************       |
  5   3.099579    0.57244  |               .    |***********         |
  6   2.880295    0.53194  |               .    |***********         |
  7   2.747178    0.50736  |               .    |**********          |
  8   2.642275    0.48798  |               .    |**********          |
  9   2.482554    0.45848  |              .     |*********           |
 10   2.348620    0.43375  |              .     |*********           |
 11   2.306213    0.42592  |              .     |*********           |
 12   2.283695    0.42176  |              .     |********            |
 13   2.121303    0.39177  |              .     |********            |
 14   2.001750    0.36969  |              .     |*******             |
 15   1.797823    0.33203  |              .     |*******             |
 16   1.557279    0.28760  |              .     |******              |
 17   1.412054    0.26078  |              .     |*****.              |
 18   1.394061    0.25746  |              .     |*****.              |
 19   1.360182    0.25120  |             .      |***** .             |
 20   1.347006    0.24877  |             .      |***** .             |
                           "." marks two standard errors
```

Figure 9. Comparison of the ACF plots of stationary and nonstationary data.

In order to determine the estimated t-values and test the significance of the autocorrelation coefficients, one must first estimate the standard error of the ACFs. M.S. Barlett [5] has derived an approximate expression for the standard error of the sampling distribution of the autoregressive coefficient r_k. This estimated standard error, designated $s(r_k)$, is calculated as follows:

$$s(r_k) = \left(1 + 2\sum_{j=1}^{k-1} r_j^2\right)^{1/2} n^{-1/2} \tag{3}$$

This expression is appropriate for processes with normally distributed random shocks where the true MA order of the process is k-1.

Now one can use the estimated standard errors to test the null hypothesis $\mathbf{H_0}$: $\rho_k = 0$ for $k = 1, 2, 3, ...$ This hypothesis is tested by finding out how far away the sample statistic r_k is from the hypothesized value $\rho_k = 0$. This distance is expressed as a t-statistic equal to the equivalent number of estimated standard errors. Thus one can approximate the t-statistic in the following fashion [2]:

$$t_{rk} = \frac{r_k - \rho_k}{s(r_k)} \tag{4}$$

3.3 The Yule-Walker Equations and AR Modeling

The Yule-Walker equations are used for determining the AR model for a known AR process. These equations describe the linear relationship between the AR parameters and the autocorrelation function. The solution of these equations is provided by the computationally efficient Levinson-Durbin algorithm [6].

A relationship between the AR parameters and the autocovariance function R_{xx} of w_t is presented. This relationship is known as the Yule-Walker equation [7]. The derivation of the Yule-Walker equation proceeds as follows [6]:

$$R_{xx}(k) \equiv E(w_{t+k}w_t)$$

$$= E\left[w_t \left(\sum_{i=1}^{p} \phi_i w_{t+k-i} - a_{t+k} \right) \right] \tag{5}$$

$$= \sum_{i=1}^{p} \phi_i R_{xx}(k-i) - E(a_{t+k} w_t)$$

where w_t is the observation from a stationary time series, a_t is the forecast error or noise, and E[] implies the expected value.

We will define the following:

$$R_{nx}(k) \equiv E(a_{t+k} w_t) \tag{6}$$

But $R_{nx}(k) = 0$ for $k > 0$ since a future input to a causal, stable filter cannot affect the present output and a_t is "white" noise. In other words, since a_t is a white excitation, it is uncorrelated with those w_t occurring prior to t. Therefore, Expression (5) can be further simplified as:

$$R_{xx}(k) = \begin{cases} \sum_{i=1}^{p} \phi_i R_{xx}(k-i), k > 0 \\ \sum_{i=1}^{p} \phi_i R_{xx}(k-i) + \sigma^2, k = 0 \end{cases} \tag{7}$$

Expression (7) is known as the Yule-Walker equations. To determine the AR parameters, one need only choose the first p equations from Expression (7) for $k > 0$, solve for $(\phi_1, \phi_2, ..., \phi_p)$, and then find σ^2 from Expression (7) for $k = 0$. The set of equations which require the fewest lags of the autocovariance function is the selection $k = 1, 2, ..., p$. They can be expressed in matrix form as [6]:

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(-1) & \dots R_{xx}(-(p-1)) \\ R_{xx}(1) & R_{xx}(0) & \dots R_{xx}(-(p-2)) \\ \vdots & \vdots & \ddots \quad \vdots \\ R_{xx}(p-1) & R_{xx}(p-2) & \dots \quad R_{xx}(0) \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} R_{xx}(1) \\ R_{xx}(2) \\ \vdots \\ R_{xx}(p) \end{bmatrix} \tag{8}$$

It should be noted that Expression (8) can also be augmented to incorporate the σ^2 equation, yielding

$$\begin{bmatrix} R_{xx}(0) & R_{xx}(-1) & \dots \quad R_{xx}(-p) \\ R_{xx}(1) & R_{xx}(0) & \dots R_{xx}(-(p-1)) \\ \vdots & \vdots & \ddots \quad \vdots \\ R_{xx}(p) & R_{xx}(p-1) & \dots \quad R_{xx}(0) \end{bmatrix} \begin{bmatrix} -1 \\ \phi_1 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} -\sigma^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{9}$$

which follows from Expression (7).

The Levinson-Durbin algorithm provides an efficient solution for Expression (9). The algorithm proceeds recursively to compute the parameter set $(\phi_{11}, \sigma_1{}^2)$, $(\phi_{21}, \phi_{22}, \sigma_2{}^2)$, ..., $(\phi_{p1}, \phi_{p2}, ..., \phi_{pp}, \sigma_p{}^2)$. Note that an additional subscript, p, has been added to the AR coefficients to denote the order of each sequence. The final set at order p is the desired solution [6]. In particular, the recursive algorithm is initialized by setting:

$$\phi_{11} = \frac{R_{xx}(1)}{R_{xx}(0)} \tag{10}$$

$$\sigma_1^2 = (1 - |\phi_{11}|^2) R_{xx}(0) \tag{11}$$

with the recursion for $i = 2, 3, ..., p$ given by

$$\phi_{kk} = \frac{R_{xx}(k) + \sum_{i=1}^{k-1} \phi_{k-1,i} R_{xx}(k-i)}{\sigma_{k-1}^2} \tag{12}$$

$$\phi_{kj} = \phi_{k-1,j} + \phi_{kk}\phi_{k-1,k-j} \tag{13}$$

$$\sigma_k^2 = (1 - |\phi_{kk}|^2)\,\sigma_{k-1}^2 \tag{14}$$

It is important to note that $(\phi_{k1}, \phi_{k2}, ..., \phi_{kk}, \sigma_k^2)$, as obtained above, is the same as would be obtained by using Expression (9) for $p = k$. Thus the Levinson-Durbin algorithm also provides the AR parameters for all the lower order AR model fits to the data. This is a useful property when one does not know *a priori* the correct model order, since one can use Expression (9) to generate successively higher order models until the modeling error σ_k^2 is reduced to the desired value.

In particular, if a process is actually an AR process of order p, then $\phi_{p+1,k} = \phi_{pk}$ for $k = 1, 2, ..., p$ and hence $\phi_{p+1,p+1} = 0$. In general for an AR order p process, $\phi_{kk} = 0$ and $\sigma_k^2 = \sigma_p^2$ for $k > p$. Hence, the variance of the excitation noise is a constant for a model order equal to or greater than the correct order. Thus, in theory, the point at which σ_k^2 does not change would appear to be a good indicator of the correct model order. This means that σ_k^2 first reaches its minimum at the correct model order [6].

3.4 The Modified Yule-Walker Equations and ARMA Modeling

The Yule-Walker equations can be modified in order to generate ARMA models. To illustrate this, let w_t be a stationary time series generated by the following ARMA equation:

$$w_t = \sum_{i=1}^{p} \phi_i w_{t-i} + \sum_{j=0}^{q} \theta_j a_{t-j} \tag{15}$$

Multiplying both sides of Equation (15) by w_{t-k} and taking the expectations, we obtain:

$$R_{xx}(k) = \sum_{i=1}^{p} \phi_i R_{xx}(k-i) + \sum_{j=0}^{q} \theta_j R_{nx}(k-j) \tag{16}$$

where, once again,

$$R_{xx}(k) \equiv E(w_{t+k} w_t) \tag{17}$$

$$R_{nx}(k) \equiv E(a_{t+k} w_t) \tag{18}$$

However, as pointed out in Section 3.2.3, one can assume that $R_{nx}(k) = 0$ for $k > 0$. Therefore,

$$R_{xx}(k) = \begin{cases} \sum_{i=1}^{p} \phi_i R_{xx}(k-i) + \sum_{j=0}^{q} \theta_j R_{nx}(k-j), i = 0, \ldots, q \\ \sum_{i=1}^{p} \phi_i R_{xx}(k-i), i = q+1, q+2, \ldots \end{cases} \tag{19}$$

Thus the AR parameters can be estimated *independently* of the MA parameters if one uses the Yule-Walker equations as given by Expression (19) [8].

A popular approach for determining the ARMA model and estimating its parameters is to use $i > q$ to find the AR parameters (ϕ_1, ϕ_2, ..., ϕ_p) and then apply some appropriate technique to find the MA parameters (θ_1, θ_2, ..., θ_q) or an equivalent parameter set. For example, to find the AR parameters, using Expression (19) and $i = q+1, q+2, ..., q+p$, we solve the following matrix expression:

$$\underbrace{\begin{bmatrix} R_{xx}(q) & R_{xx}(q-1) & \cdots & R_{xx}(q-p+1) \\ R_{xx}(q+1) & R_{xx}(q) & \cdots & R_{xx}(q-p+2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(q+p-1) & R_{xx}(q+p-2) & \cdots & R_{xx}(q) \end{bmatrix}}_{|\mathbf{R}_{xx}|} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} R_{xx}(q+1) \\ R_{xx}(q+2) \\ \vdots \\ R_{xx}(q+p) \end{bmatrix} \qquad (20)$$

These equations have been called the *extended*, or *modified*, Yule-Walker equations.

The AR order can be determined by testing the singularity of the correlation matrix $|\mathbf{R}_{xx}|$. Therefore, in order to choose an appropriate model order p for the AR portion of the ARMA model, the property

$$\det|\mathbf{R}_{xx}| = 0 \qquad (21)$$

for dimension of $|\mathbf{R}_{xx}|$ greater than the AR order p can be used. The AR coefficients (ϕ_1, ϕ_2, ..., ϕ_p) can then be solved using the linear system of equations in Expression (20).

Expression (19) can be used to determine the MA order q, since q is seen to be the largest integer k for which

$$R_{xx}(k) - \sum_{i=1}^{p} \phi_i R_{xx}(k-i) \neq 0 \qquad (22)$$

This process can be repeated with the new estimate of the MA order q. Furthermore, the MA coefficients $(\theta_1, \theta_2, ..., \theta_q)$ can be determined using appropriate iterative optimization techniques [8].

3.5 Modifications to Time-Series Model Generation Algorithm for Noise Compensation

An inherent problem with automatic time-series model generation for the BCAM SPC scheme is that the real-time sensor signals tend to be very noisy. This makes it very difficult to determine the AR order using the matrix form of the modified Yule-Walker equations:

$$\underbrace{\begin{bmatrix} R_{xx}(q) & R_{xx}(q-1) & \dots R_{xx}(q-p+1) \\ R_{xx}(q+1) & R_{xx}(q) & \dots R_{xx}(q-p+2) \\ \vdots & \vdots & \ddots \quad \vdots \\ R_{xx}(q+p-1) & R_{xx}(q+p-2) & \dots \quad R_{xx}(q) \end{bmatrix}}_{|R_{xx}|} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} R_{xx}(q+1) \\ R_{xx}(q+2) \\ \vdots \\ R_{xx}(q+p) \end{bmatrix} \quad (23)$$

where

$$\det|R_{xx}| = 0 \quad (24)$$

for dimension of $|R_{xx}|$ greater than the AR order p. This is because it is very difficult to test whether the determinant of the correlation matrix $|R_{xx}|$ has reached zero.

However, Expression (19) can be used to determine the AR order as well as the AR coefficients $(\phi_1, \phi_2, ..., \phi_p)$ using simple linear regression. Expression (19) is repeated here for convenience:

$$R_{xx}(k) = \begin{cases} \sum\limits_{i=1}^{p} \phi_i R_{xx}(k-i) + \sum\limits_{j=0}^{q} \theta_j R_{nx}(k-j), i = 0, \ldots, q \\ \sum\limits_{i=1}^{p} \phi_i R_{xx}(k-i), i = q+1, q+2, \ldots \end{cases} \tag{25}$$

A linear regression can be fitted to the time series using Expression (19) for $i > q$ with an initial high AR order. The significance of the AR coefficients is then tested with the insignificant highest-order AR coefficient omitted, and the regression repeated with a lower AR order. This process is repeated until all AR coefficients determined using the linear regression methodology are tested to be significant. (The significance testing of the AR coefficients is typically done using an appropriate limit on the t-value of the coefficients.)

This methodology has been tested using real-time sensor data and has shown to be efficient in helping determine the AR order and coefficients of an ARMA process despite noisy samples. This will be demonstrated in the results shown in Chapter 5.

3.6 Summary

The process of automatically generating ARIMA time-series models essentially boils down to determining an optimal model order structure as represented by a point in a three-dimensional space as shown in Figure 10.

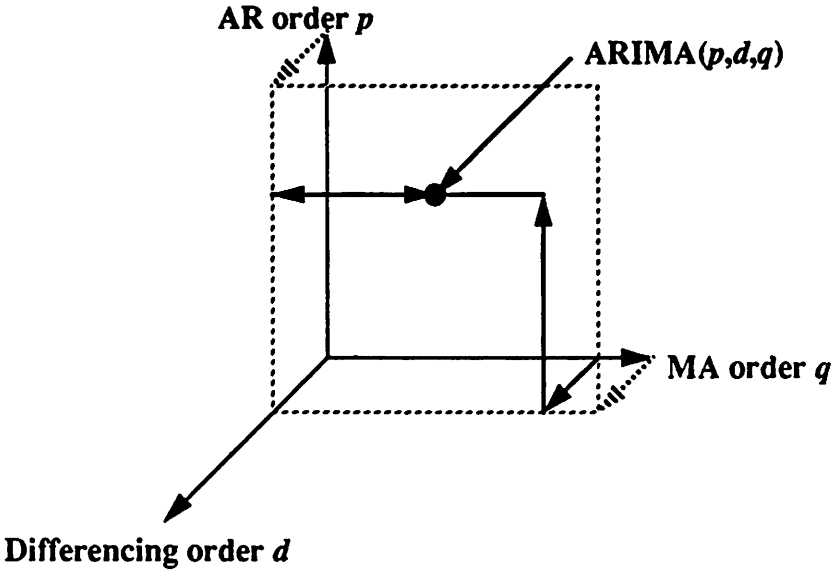


Figure 10. Algorithm for determining the structure of an ARIMA model.

The model generation process starts by determining the appropriate differencing order needed in order to derive a "stationary" time series, thus reducing the problem down to a search for an optimal point in a two-dimensional space. The *modified* Yule-Walker equations are then used with a high initial guess for the MA order q in order to determine the AR order p and the AR coefficients $(\phi_1, \phi_2, ..., \phi_p)$. The same modified Yule-Walker equations can then be used to estimate the MA order q as shown using Expression (22). This process is then repeated with the new estimate of the MA order q until the process converges. The MA coefficients $(\theta_1, \theta_2, ..., \theta_q)$ are then solved using iterative optimization techniques.

Chapter 4 Modified Real-Time SPC with Automatic Time-Series Model Generation

4.1 Overview of Modifications

As explained in section 2.2, one cannot use the seasonal ARIMA (SARIMA) model in order to model equipment sensor data for real-time SPC purposes. However, one can modify the method in which ARIMA models are generated in order to develop satisfactory filters for real-time SPC. This is done by decomposing the original sensor signal into two components, the *within-wafer* and *wafer-to-wafer* components, and by developing two separate ARIMA models: one for modeling the characteristics of the data variation from within the critical step of each wafer and one for modeling the wafer-to-wafer variation of the real-time sensor data. These models can be generated automatically using the algorithm described in Chapter 3.

The within-wafer and wafer-to-wafer residuals can then be combined into two separate T^2 statistics for SPC purposes. One will be used for detecting within-wafer processing faults while the other will be used for detecting wafer-to-wafer faults. An example of the signal decomposition and filtering process is shown in Figure 11.

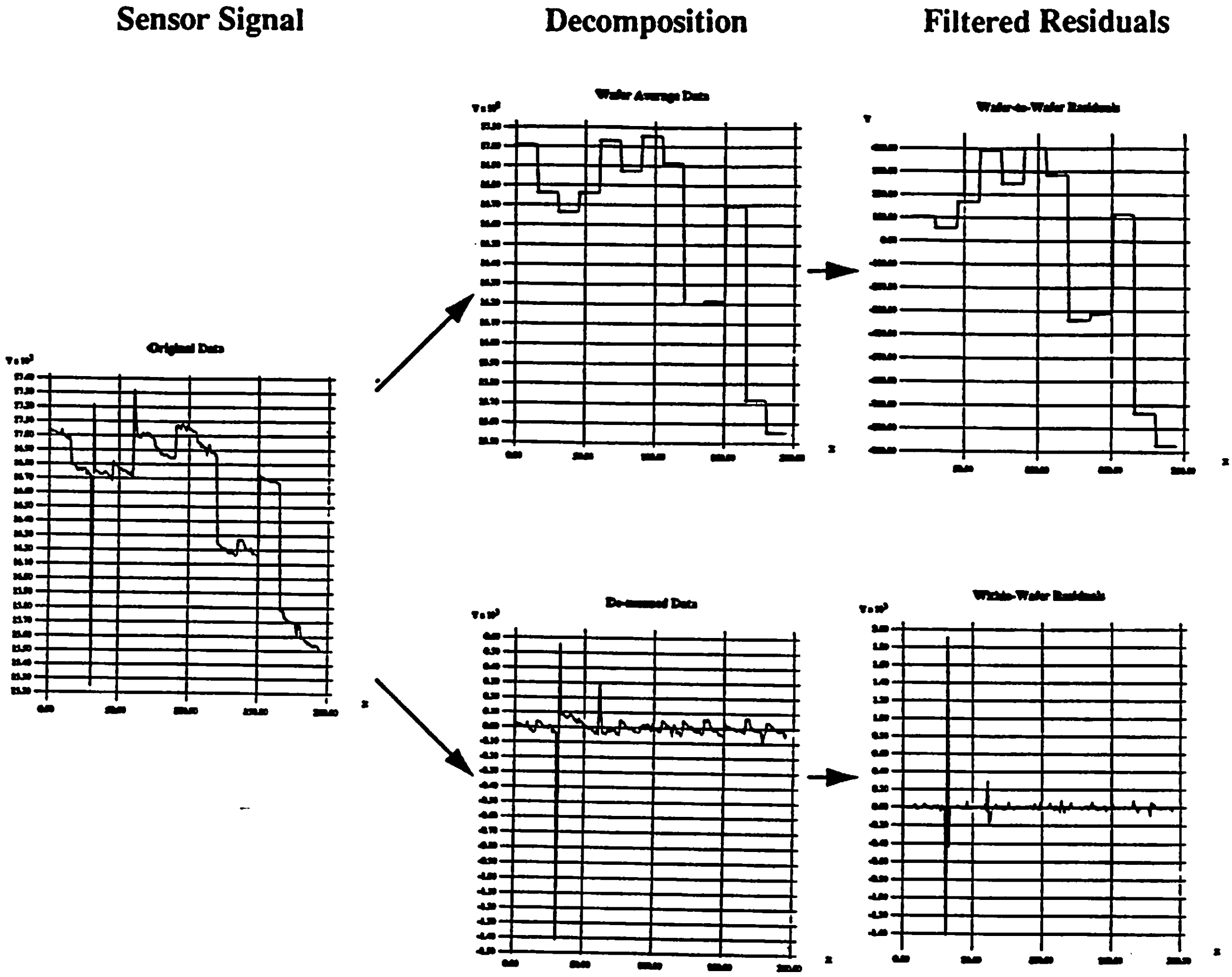


Figure 11. Example of the signal decomposition and filtering process.

4.2 Within-Wafer Data Modeling and Filtering

After carefully analyzing the equipment sensor data, it can be seen that the sensor data display a distinctive auto-correlated pattern during each wafer processing step. This pattern tends to repeat itself with every wafer processed. An ARIMA model can be built in order to model the characteristics of these within-wafer patterns by selectively looking at only the time-series autocorrelations within each wafer. The within-wafer time-series models and filtered residuals can be used to detect slight problems in the wafer processing step. Shown in Figure 12 is an example of this repeated pattern for select sensor signals from the Lam Research Rainbow plasma etcher and the IIND residuals after filtering the data with the automatically generated ARIMA time-series models.

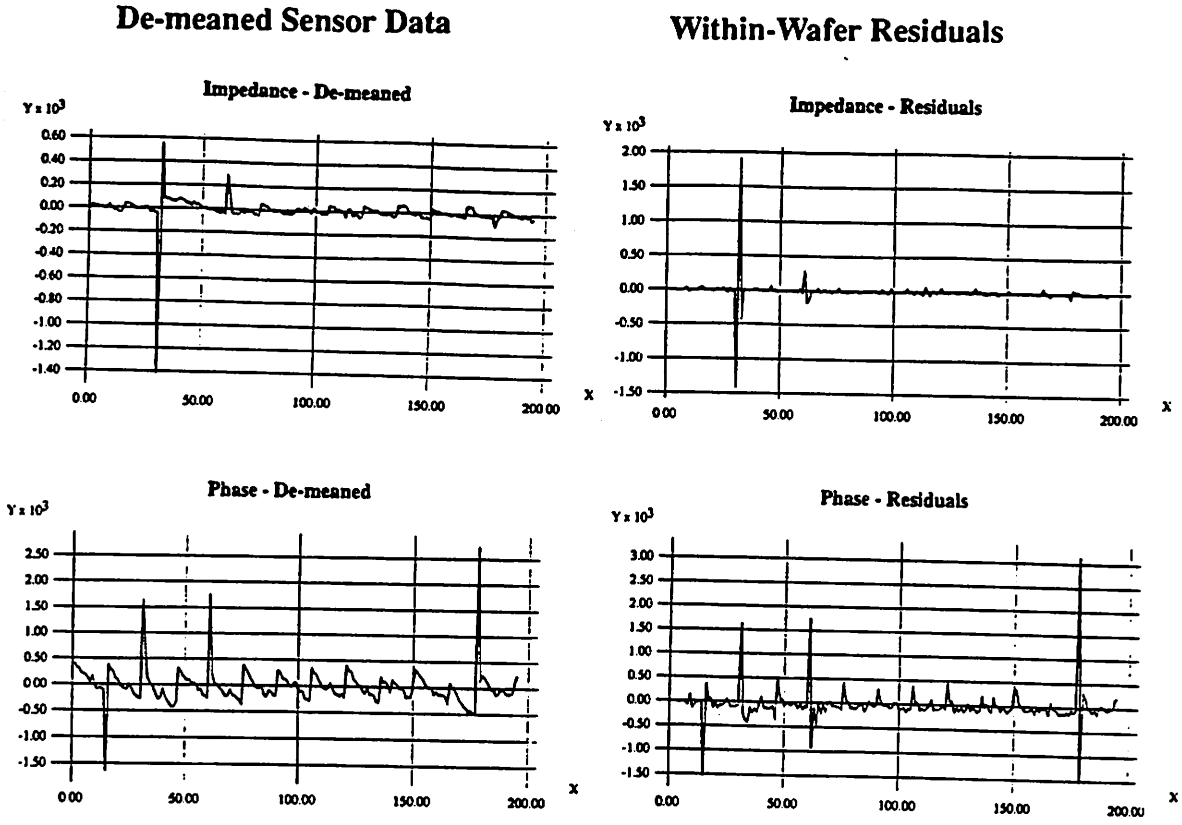


Figure 12. Selected equipment sensor data from the Lam Research Rainbow plasma etcher showing the repeated auto-correlated pattern for each wafer processed and their corresponding IIND residuals after filtering. Sensor data for each wafer have been de-meaned so that the within-wafer time-series pattern can be seen more easily.

Modifications to the ARIMA model generation algorithm must be made in order to generate an appropriate model for the de-meaned within-wafer data. The modifications simply involve calculating the time-series statistics (i.e. the autocorrelation functions) using selective samples that embody only the within-wafer time-series characteristics.

This means eliminating samples involving uncorrelated datapoints across wafers. An illustration of this selective sampling process is shown in Figure 13.

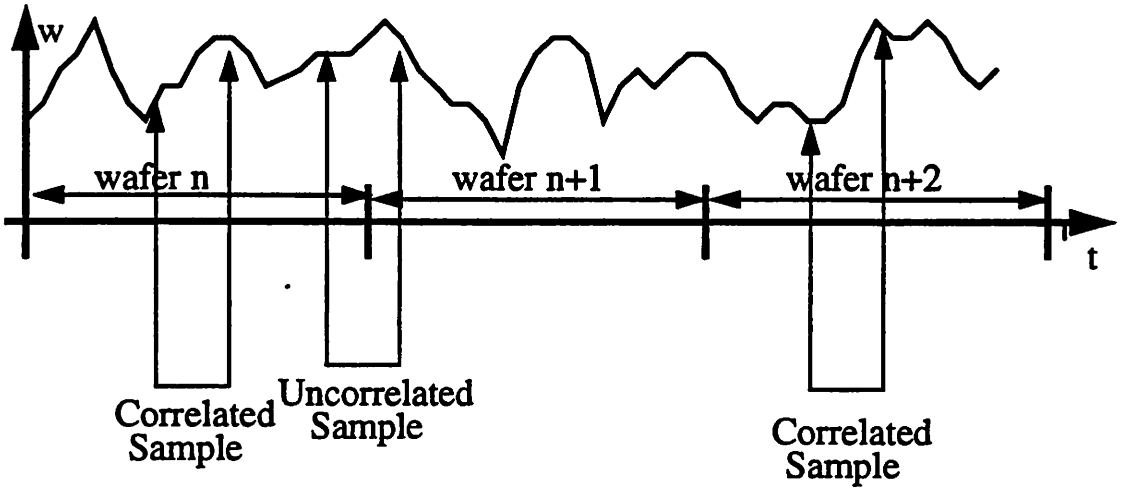


Figure 13. An illustration of the selective sampling process for determining within-wafer ARIMA time-series models.

4.3 Wafer-to-Wafer Data Modeling and Filtering

Although most sensor data show little autocorrelation across wafers, there are certain signals that have significant autocorrelation from wafer to wafer. This wafer-to-wafer correlation must be filtered with an appropriate time-series model. By looking at only the correlations between the wafer averages, one can build time-series models that will be able to filter these wafer-to-wafer correlations. The wafer-to-wafer time-series models and filtered residuals can be used to detect catastrophic problems in the wafer processing step (i.e. a significant shift in the real-time sensor signal). Selected original equipment sensor data, wafer averages of the original data, and their corresponding IIND wafer-to-wafer residuals are shown in Figure 14.

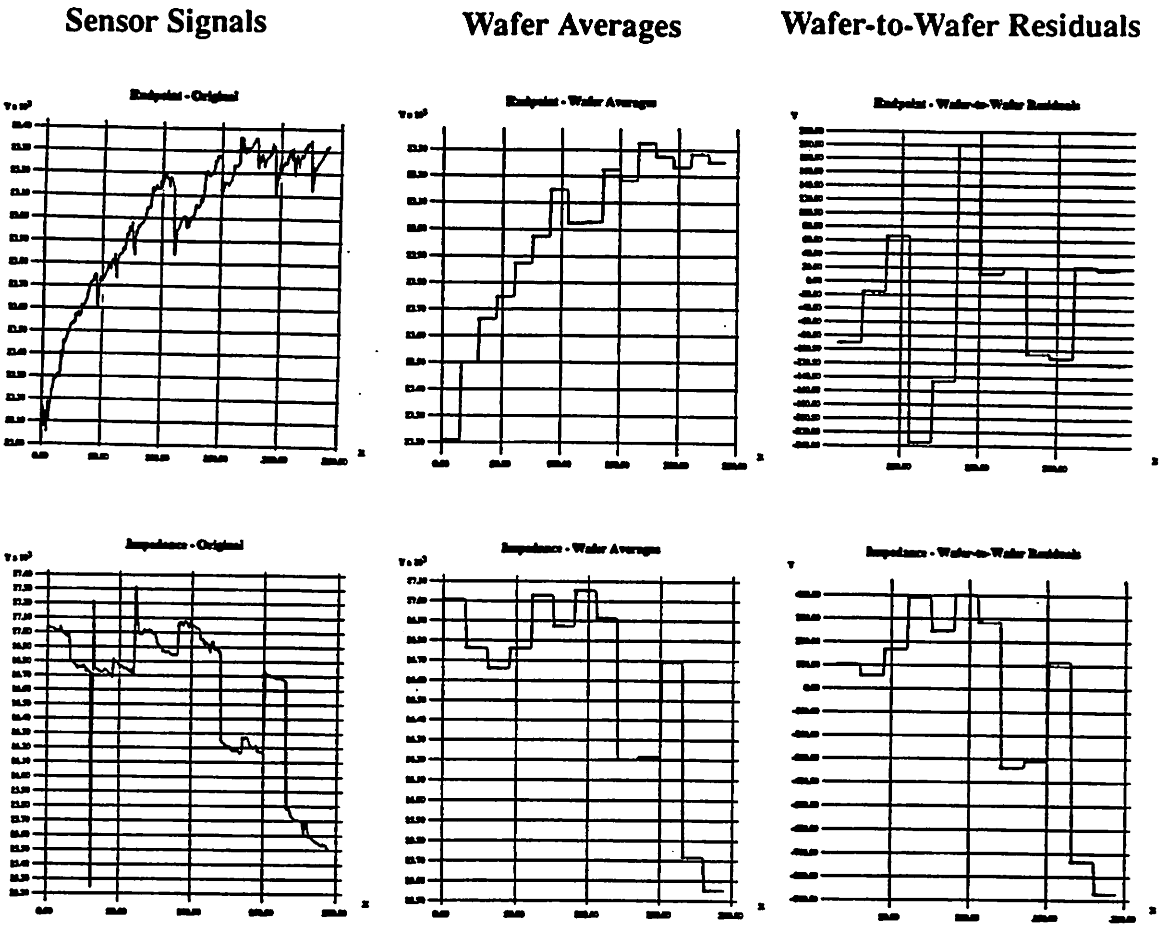


Figure 14. Selected equipment sensor data, wafer averages of the original data, and their corresponding IIND wafer-to-wafer residuals.

4.4 The Double-T^2 Control Chart

The Hotelling's T^2 statistic can be calculated for both the IIND within-wafer residuals and the IIND wafer-to-wafer residual means. The T^2 statistic is a well-defined variable that represents a combined score for many cross-correlated variables, and is calculated by grouping *n* readings from each of *p* cross-correlated parameters [3]:

$$
\begin{aligned}
&T^2 = n(\bar{\mathbf{X}}-\tilde{\mathbf{X}})^T S^{-1}(\bar{\mathbf{X}}-\tilde{\mathbf{X}}) \\
&\text{where group mean} \quad \mathbf{X}^T = [\bar{x}_1 \ldots \bar{x}_p] \\
&\text{nominal value} \quad \tilde{\mathbf{X}}^T = [\tilde{x}_1 \ldots \tilde{x}_p] \\
&\text{variance-covariance matrix} \quad \mathbf{S} = \begin{bmatrix} s_1^2 & \cdots & s_{1p} \\ \vdots & \ddots & \vdots \\ s_{p1} & \cdots & s_p^2 \end{bmatrix}
\end{aligned}
\qquad (26)
$$

The distribution of the T^2 statistic is related to the F-distribution as follows:

$$T^2_{\alpha,\,p,\,n-1} = \frac{p\,(n-1)}{n-p} F_{\alpha,\,p,\,n-p}$$

The T^2 statistic defined above is optimal for detecting mean shifts under the assumption of multivariate normality. Furthermore, it can be extended to guard against shifts in the variance of the monitored data. However, it is not geared towards identifying a shift in the variance-covariance matrix and will confound such a shift with a shift in the mean vector.

This statistic takes a low value when the average values of the cross-correlated variables are small. The T^2 score is very sensitive to any change in the mean of one or more of the combined variables. This score can be used in conjunction with a one-sided control chart whose limit is determined according to the number of variables, the sample (or *group*) size and the acceptable percentage of false alarms.

As noted above and in Expression (26), readings are grouped according to a specified *group size* n and a T^2 statistic is calculated for each group. This grouping is necessary in order to compensate for the occasional noise in the real-time data. Thus an occasional noisy spike in the data will not cause a large T^2 alarm given that the fault is minor.

The Double-T^2 Control Chart is a one-sided control chart displaying both the within-wafer and the wafer-to-wafer T^2 statistics in order to determine if a process is in a state of control. If a process goes out of control, one can determine whether the alarm was caused by a within-wafer or wafer-to-wafer fault. An example of the Double-T^2 Control Chart is shown in Figure 15.

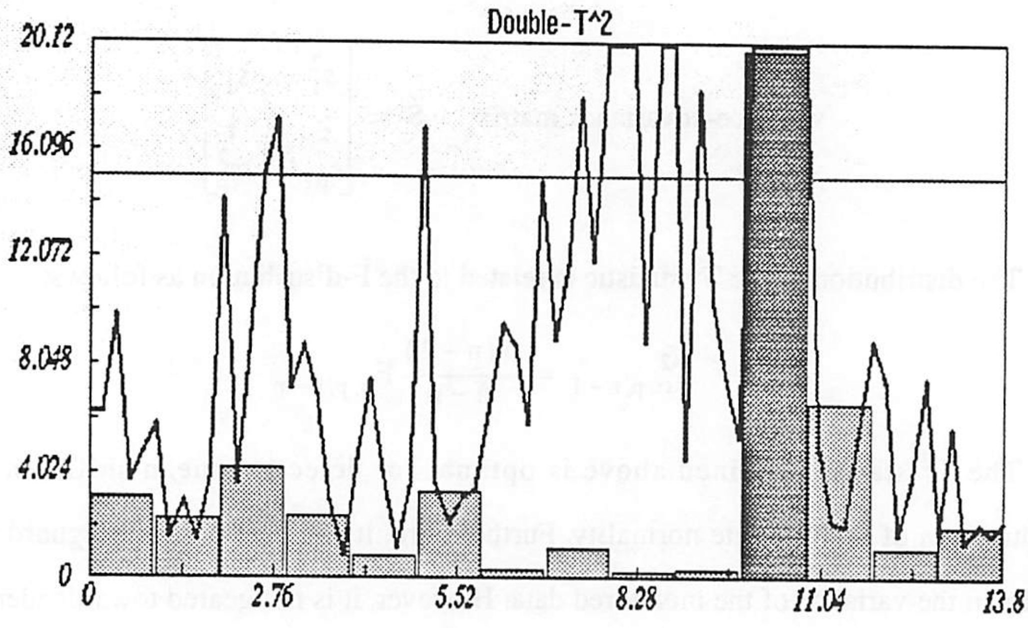


Figure 15. An example of a Double-T^2 Control Chart. The line graph plots the within-wafer T^2 statistic with a specified group size. The bar graph shows the wafer-to-wafer T^2 statistic for each wafer processed. (NOTE: The two T^2 statistics have been scaled so as to have the same control limit.)

4.5 Summary

In conclusion, an automatic ARIMA time-series model generator has been added to the BCAM Real-Time SPC module in order to make the application more practical and robust. Furthermore, the system has been modified so that two time-series models are generated for each signal in order to filter the within-wafer and wafer-to-wafer variation separately. This in terms implies the generation of two separate T^2 statistics for real-time SPC: one for signalling within-wafer faults and the other for signalling wafer-to-wafer faults. These two T^2 statistics are scaled and plotted together on what is called a Double-T^2 Control Chart. This scheme has been implemented in software and hardware, and is discussed in the following chapter.

Chapter 5 Implementation and Experimental Results

5.1 Implementation

5.1.1 The BCAM Real-Time SPC System

As discussed in the preceding chapter, the BCAM Real-Time Statistical Process Control system first decomposes the real-time sensor signal into its within-wafer and wafer-to-wafer components and filters them separately. The filtered residuals are then combined into two separate T^2 statistical scores that are then scaled and placed on a Double-T^2 Control Chart in order to detect abnormal within-wafer and wafer-to-wafer variations.

A top-level description of the dataflow for the system is shown below in Figure 16.

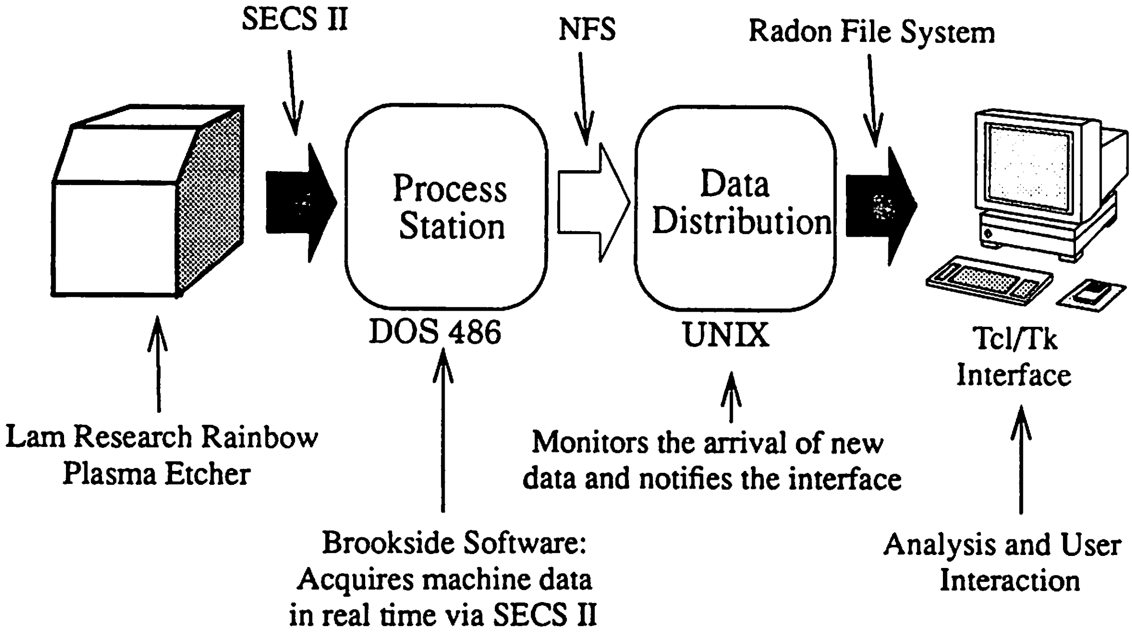

SECS II
NFS
Radon File System
Process Station
Data Distribution
DOS 486
UNIX
Tcl/Tk Interface
Lam Research Rainbow Plasma Etcher
Monitors the arrival of new data and notifies the interface
Brookside Software: Acquires machine data in real time via SECS II
Analysis and User Interaction

Figure 16. A top-level description of the dataflow for the Real-Time SPC module

The system has been implemented using the C and C++ programming languages in the UNIX environment. A Tcl/Tk graphical user interface has also been built. The prototype system is depicted in Figure 17.

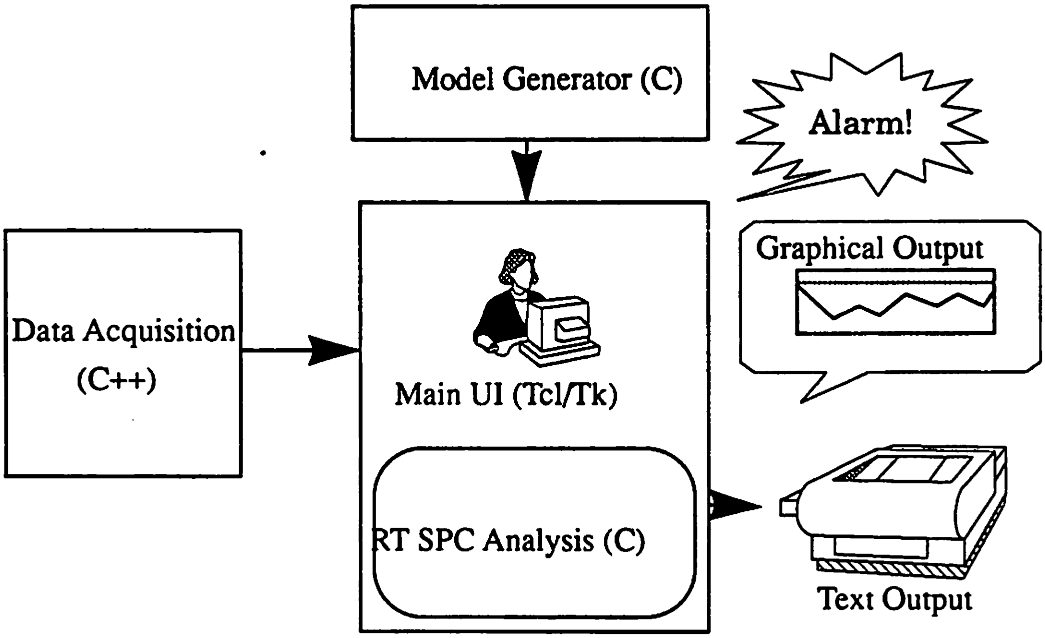
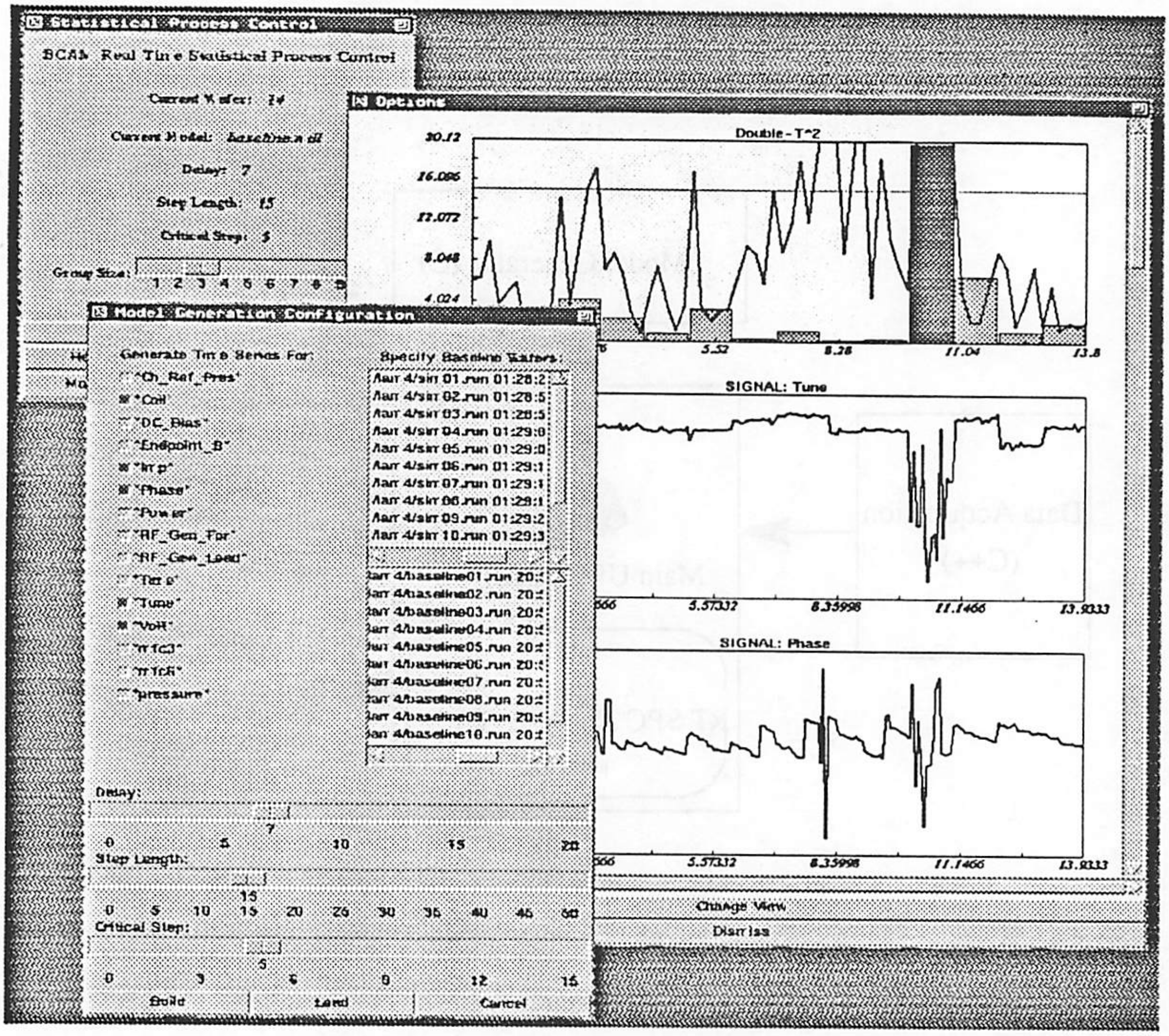


Figure 17. Description of the Tcl/Tk interface

A screen dump of the Real-Time SPC graphical user interface is shown in Figure 18. This interface includes a main panel, a model generation panel, and a Real-Time SPC window.



Statistical Process Control

BCAM Real Time Statistical Process Control

Current Wafer: 14

Current Model: baseline.n dl

Delay: 7

Step Length: 15

Critical Step: 5

Group Size: 1 2 3 4 5 6 7 8 9

Model Generation Configuration

Generate Time Series For:

□ "Ch_Ref_Pres"
☒ "Coil"
□ "DC_Bias"
□ "Endpoint_B"
☒ "Imp"
☒ "Phase"
□ "Power"
□ "RF_Gen_For"
□ "RF_Gen_Load"
□ "Temp"
☒ "Tune"
☒ "Volt"
□ "mfc3"
□ "mfc4"
□ "pressure"

Specify Baseline Wafers:

```
/Aarr 4/sim 01.run 01:28:2
/Aarr 4/sim 02.run 01:28:5
/Aarr 4/sim 03.run 01:28:5
/Aarr 4/sim 04.run 01:29:0
/Aarr 4/sim 05.run 01:29:0
/Aarr 4/sim 06.run 01:29:1
/Aarr 4/sim 07.run 01:29:1
/Aarr 4/sim 08.run 01:29:1
/Aarr 4/sim 09.run 01:29:2
/Aarr 4/sim 10.run 01:29:3
```

```
arr 4/baseline01.run 20:
arr 4/baseline02.run 20:
arr 4/baseline03.run 20:
arr 4/baseline04.run 20:
arr 4/baseline05.run 20:
arr 4/baseline06.run 20:
arr 4/baseline07.run 20:
arr 4/baseline08.run 20:
arr 4/baseline09.run 20:
arr 4/baseline10.run 20:
```

Delay: 7

Step Length: 15

Critical Step: 5

Figure 18. Real-Time SPC screen dump

5.1.2 Automatic ARIMA Time-Series Model Generator

The automatic ARIMA time-series model generation sequence for real-time SPC is illustrated below in Figure 19.

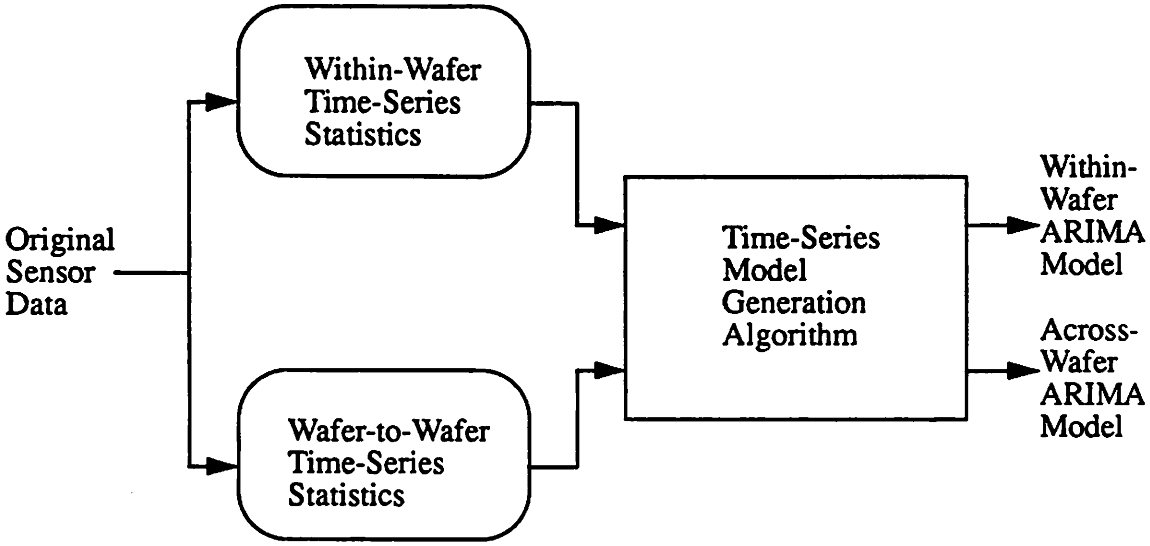

Original Sensor Data
Within-Wafer Time-Series Statistics
Wafer-to-Wafer Time-Series Statistics
Time-Series Model Generation Algorithm
Within-Wafer ARIMA Model
Across-Wafer ARIMA Model

Figure 19. Flow chart for the automatic ARIMA time-series model generation process for real-time SPC.

As shown in Figure 19, the within-wafer and wafer-to-wafer time-series statistics are determined separately through selective sampling described in sections 4.2 and 4.3. These time-series statistics are then used in conjunction with the model generation algorithm described in Chapter 3 in order to generate the appropriate within-wafer and wafer-to-wafer models for each real-time sensor signal.

The functionality of the time-series model generator has been demonstrated by applying it to the BCAM Real-Time SPC system. The model generator has been implemented using the C programming language in the UNIX environment and has been integrated into the BCAM Real-Time SPC module that was described in the previous section. This generator is capable of generating a pair of within-wafer and wafer-to-wafer

ARIMA models in less than five seconds real time and less than one second CPU time on a Sun SPARCstation 2™.

5.2 Experimental Results

5.2.1 Summary of Experiment

The following experiment was demonstrated in the *SRC Real-Time Statistical Process Control Workshop* at the University of California, Berkeley, on May 10-11, 1993. It involves the generation of five pairs of within-wafer and wafer-to-wafer ARIMA time-series models for five signals. These signals were the Coil Position, the Impedance, the Phase Magnitude, the Tune Vane and the Peak-to-Peak Voltage collected from the Lam Research Rainbow plasma etcher in the Berkeley Microfabrication Laboratory. Twelve polysilicon wafers were processed in order to produce the baseline data needed for the generation of the time-series models. The signals, along with their corresponding within-wafer and wafer-to-wafer filtered residuals are shown below in Figure 20.

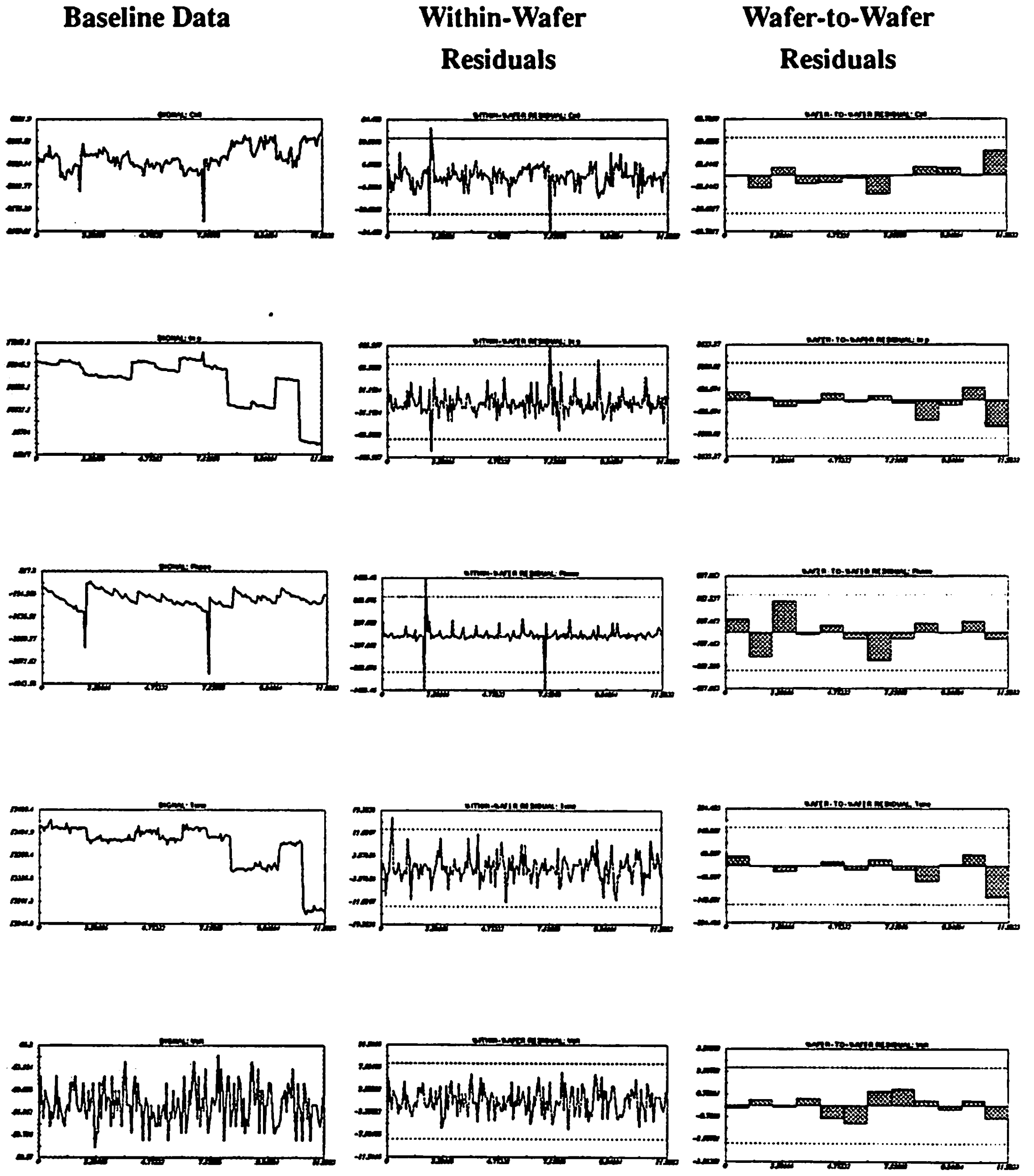


Figure 20. Baseline data for five sensor signals along with their corresponding within-wafer and wafer-to-wafer filtered residuals.

The within-wafer and wafer-to-wafer ARIMA time-series models generated using the automatic time-series model generator is shown in Table 1.

Table 1: Within-Wafer and Wafer-to-Wafer ARIMA Models Used for Experiment

Signal	Within-Wafer Model	Wafer-to-Wafer Model
Coil	ARIMA(2,0,1)	ARIMA(2,0,1)
Impedance	ARIMA(2,1,0)	ARIMA(1,1,0)
Phase	ARIMA(1,0,0)	ARIMA(1,0,2)
Tune	ARIMA(1,0,0)	ARIMA(0,1,0)
Volt	ARIMA(0,0,0)	ARIMA(1,0,1)

After the appropriate baseline models have been built, fourteen wafers were processed and monitored using the BCAM Real-Time SPC scheme through the Tcl/Tk interface described in Chapter 5. Known faults were introduced as follows in Table 2:

Table 2: Description of Wafers in Real-Time SPC Experiment

Wafer #	Description
1-7	Clean wafers with blanket polysilicon layer
8-9	*Wafers with dirty polysilicon film*
10	Clean wafer with blanket polysilicon layer
11	*Wafer from wrong batch with photoresist remaining*
12-14	Clean wafers with blanket polysilicon layer

5.2.2 Within-Wafer Residuals

The original sensor signals and their corresponding within-wafer residuals are shown in Figure 21.

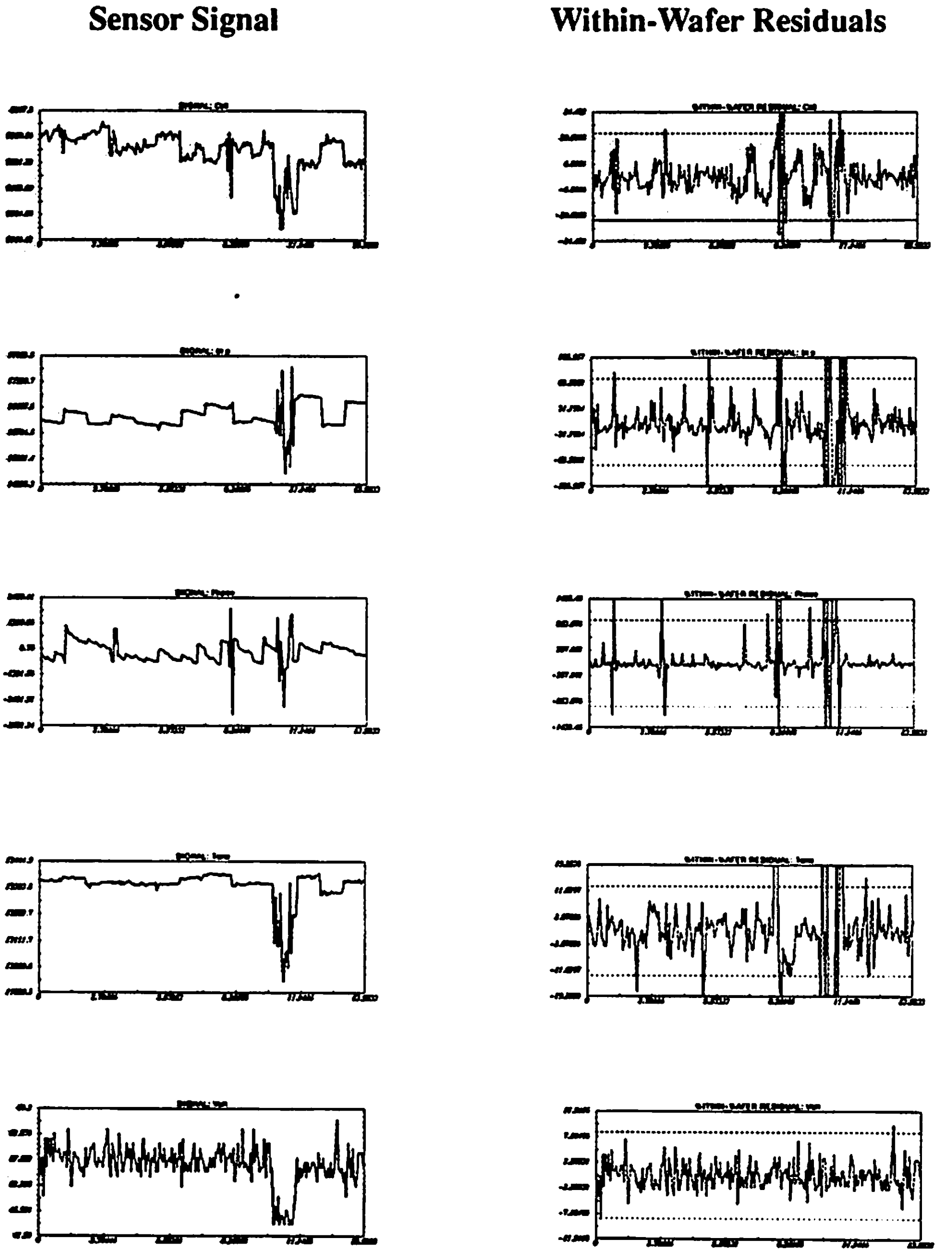


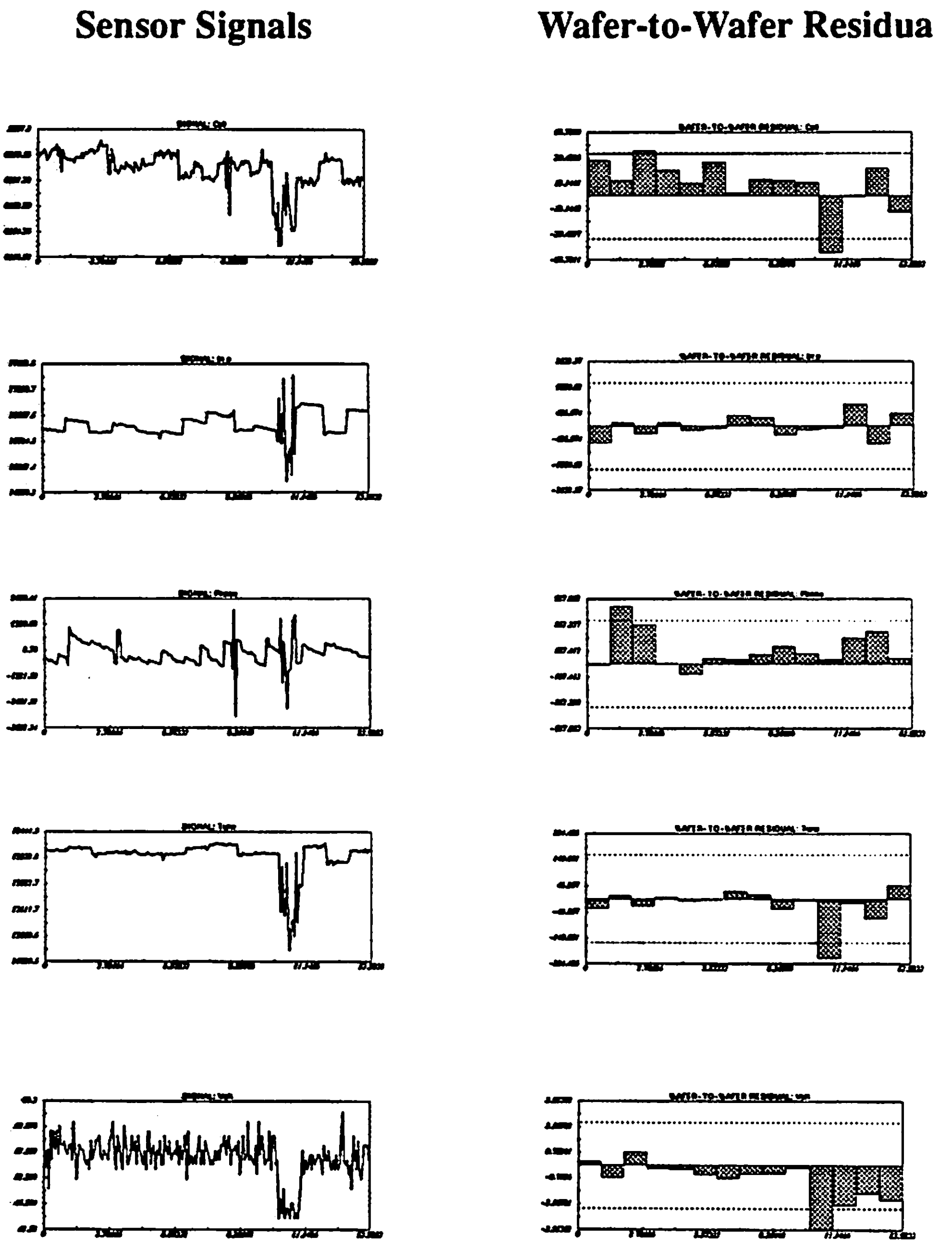
Figure 21. Original Sensor signals and their corresponding within-wafer residuals with a 3-σ control limit.

One can see significant within-wafer problems in wafers #9 and #11. This is obvious because wafer #9 was deposited with dirty polysilicon film and wafer #11 came from the wrong batch and contains unwanted photoresist. One can also see slight within-wafer problems with wafers #3, #6, #8 and #10. (This is most obvious by looking at the original signals and residuals of the Tune Vane, the Phase Magnitude and the Impedance.) Wafer #8, like wafer #9, was also deposited with dirty polysilicon film. Wafers #3, #6 and #10 did not have any known problems.

Wafers #8, #9 and #11 are known to be faulty wafers and are correctly identified by the within-wafer residuals. Wafers #3, #6 and #10 might possess problems that were unknown prior to the runs.

5.2.3 Wafer-to-Wafer Residuals

The original sensor data and their corresponding wafer-to-wafer residuals are shown below in Figure 22.



Sensor Signals

Wafer-to-Wafer Residuals

Figure 22. Original sensor signals and their corresponding wafer-to-wafer residuals with a 3-σ control limit

The wafer-to-wafer residuals tend to detect catastrophic faults, since a large wafer-to-wafer residual usually signifies a major mean shift in the sensor signal. It is therefore obvious by looking at the original signals and the wafer-to-wafer residuals that there are significant problems with wafer #11. This is easily seen by looking at the mean shifts in the Tune Vane, the Peak-to-Peak Voltage, and the Coil Position. The mean shifts in these signals are also reflected in their respective wafer-to-wafer residuals. Thus by looking at the wafer-to-wafer residuals, one can detect a significant problem with wafer #11, which contains some unwanted photoresist.

5.2.4 The Double-T^2 Control Chart

The within-wafer and wafer-to-wafer T^2 statistics are plotted on the Double-T^2 Control Chart shown below in Figure 23.

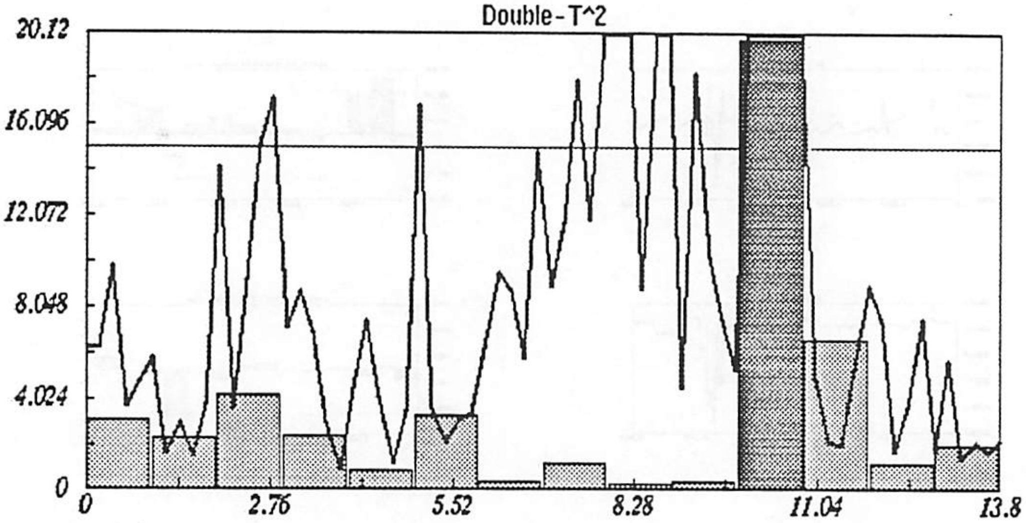


Figure 23. Double-T^2 Control Chart from the experiment. (NOTE: The line plots are the within-wafer T^2s and the bar plots are the wafer-to-wafer T^2s.)

One can see that the within-wafer T^2 statistics were able to signal problems with wafers #3, #6, #8, #9, #10 and #11, with numerous significant T^2 alarms for wafers #9 and

#11. Thus the within-wafer T^2 statistics correctly identified the known problems with wafers #8, #9 and #11. (These problems are itemized in Table 2.) The alarms for wafers #3, #6 and #10 might be either false alarms, or they might signal slight problems with the wafers, which were unknown prior to the run.

The wafer-to-wafer T^2 statistics clearly identified the catastrophic fault in wafer #11, which is a wafer with unwanted photoresist. Since this was the only wafer that cause significant mean shifts in the sensor signals, it was the only one with an wafer-to-wafer T^2 alarm generated.

5.2.5 Results

The experiment shows that the automatic time-series model generator was able to generate satisfactory models for detecting wafer or processing faults in real-time. Furthermore, the modified real-time SPC scheme is shown to be effective in detecting two different types of faults: slight faults caused by within-wafer processing instabilities and catastrophic faults resulting from major mean shifts in the sensor signals.

Chapter 6 Conclusions and Future Work

The BCAM Real-Time Statistical Process Control scheme has been modified, and an automatic time-series model generator has been developed and integrated in the BCAM SPC module. This modified scheme along with the automatic model generation algorithm has been applied on the Lam Research Rainbow single-wafer plasma etcher. The model generation algorithm has demonstrated success in generating useful time-series models for filtering real-time sensor data. Furthermore, the modified real-time SPC scheme has been shown to be superior in detecting processing faults than the originally proposed methodology.

The modifications made to the BCAM Real-Time SPC scheme involve decomposing the original sensor signals into two separate components to be analyzed independently: the within-wafer and wafer-to-wafer time-series components. Separate T^2 statistics for the within-wafer and wafer-to-wafer analyses are then plotted on a Double-T^2 Control Chart. This will allow one to not just detect processing faults, but also determine whether these faults are minor problems resulting from within-wafer processing instabilities, or catastrophic processing errors resulting from major mean shifts in the equipment sensor signals.

The automatic ARIMA time-series model generation algorithm involves determining the Integration, Autoregressive and Moving Average components of the model separately. This algorithm is facilitated by the use of the modified Yule-Walker equations [6]. Furthermore, this algorithm has been modified so that the original equipment sensor signals may be decomposed and separate models generated for the within-wafer and wafer-to-wafer time-series data.

The automatic time-series model generation algorithm has the potential of making several other computer-aided manufacturing applications more practical and robust. These applications include equipment modeling, real-time equipment control, wafer-to-wafer control and real-time equipment diagnosis. Further studies will be conducted in order to determine the feasibility of applying time-series modeling to the CAM applications mentioned above.

In addition, other methods for filtering the sensor data for real-time SPC will be studied. These include possible use of the Kalman Filters, the theory of principle components or just simple exponentially-weighted moving averages.

References

[1] Douglas C. Montgomery, *Introduction to Statistical Quality Control*, 2nd edition, New York: John Wiley & Sons, 1991.

[2] Alan Pankratz, *Forecasting with Univariate Box-Jenkins Models - Concepts and Cases*, New York: John Wiley & Sons, 1983.

[3] Richard J. Harris, *A Primer of Multivariate Statistics*, London: Academic Press, 1975.

[4] Costas J. Spanos, Hai-Fang Guo, Alan Miller and Joanne Levine-Parrill, "Real-Time Statistical Process Control Using Tool Data", *IEEE Transactions on Semiconductor Manufacturing*, Vol.5, No. 4, pp. 308-318, November 1992.

[5] M.S. Bartlett, "On the Theoretical Specification of Sampling Properties of Autocorrelated Time Series", *Journal of the Royal Statistical Society*, Vol. B8, p. 27, 1946.

[6] Steven M. Kay and Stanley Lawrence Marple, Jr., "Spectrum Analysis - A Modern Perspective", *Proceedings of the IEEE*, Vol. 69, No. 11, pp. 1380-1419, November 1981.

[7] G.E.P. Box and G.M. Jenkins, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day, 1976.

[8] Joseph C. Chow, "On Estimating the Orders of an Autoregressive Moving-Average Process with Uncertain Observations", *IEEE Transactions on Automatic Control*, pp. 707-709, October 1972.