A CONTROL AND DIAGNOSTIC SYSTEM FOR THE PHOTOLITHOGRAPHY PROCESS SEQUENCE

by

Sovarong Leang

Memorandum No. UCB/ERL M95/69

25 August 1995

A CONTROL AND DIAGNOSTIC SYSTEM FOR THE PHOTOLITHOGRAPHY PROCESS SEQUENCE

by

Sovarong Leang

Memorandum No. UCB/ERL M95/69

25 August 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Abstract

A Control and Diagnostic System for the Photolithography Process Sequence

by

Sovarong Leang

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Prof. Costas J. Spanos, Chair

This dissertation presents a methodology for developing a generic control and diagnostic system for a sequence of interrelated processes. The goal of the system is to provide an economical way of increasing process capability through innovative use of statistical techniques and probability theory.

The control system consists of a feedback loop and a feed-forward loop. The feedback loop improves the process capability by keeping the outputs of the process centered around target, while the feed-forward controller improves it further, by reducing its variance. Although the control schemes are themselves not novel, the way they are used is. Well known statistical techniques and optimization algorithms have been chosen instead of heuristics, to support the control schemes. Then, the resulting control methodology can be applied to any machine, and its accuracy can be properly quantified. If the equipment models were more complex, the control methodology would still be valid, although better optimization algorithms may needed for the recipe generation and model updating algorithms. Experimental results have shown that our control system can significantly improve the process capability of the photolithography sequence in our laboratory, resulting in photoresist patterns whose variance is reduced by a factor of 2, and whose closeness to target is limited only by measurement error and model prediction error.

A diagnostic system, which complements and is activated by the control system, has also been developed. Its goal is to assist the operator in diagnosing the cause of the decreased machine performance. As in the case of the controller, its structure is also generic and can be applied on any machine. The diagnostic system is based on conventional probability theory, because its mathematical foundations are rigorous, and its assumptions are valid in most process domains. The main novelty of our diagnostic system is that it incorporates both shallow and deep level information as evidence, so that any evidence can be used to diagnose faults. Typically, current diagnostic systems only handle one type of information (i.e, either shallow level or deep level only), which limits their diagnosis capabilities, since some faults can only be diagnosed from deep level information, whereas others can only be diagnosed from shallow level information. From the evidence data, and from the conditional probabilities of faults initially supplied by machine experts (and subsequently updated by the system), the fault probabilities and their bounds are calculated, given a specified confidence level. Theoretical derivation show that the rate of convergence of the fault probabilities follow a multinomial distribution. We have implemented a software version of the diagnostic system, and we have tested it on real photolithography equipment malfunctions and drifts. Although it is often successful in diagnosing the correct fault, the diagnostic system can use further inputs from machine experts. Other possible improvements include better fault signature filtering, and a more efficient way of obtaining and managing the conditional probabilities of faults.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

To stay competitive, semiconductor industries must develop efficient, high yielding manufacturing facilities. One way to increase yield is to reduce and control process variability. This is a difficult task, because not only are semiconductor processes not always well understood, they also drift with time due to equipment aging, depletion of chemicals, or changing ambient conditions. All these compounded instabilities decrease the overall process capability.

One approach to reduce process variation is to use a supervisory system that controls processes on a real-time basis. Applied on modern analytical and processing equipment that have the ability to interact with computer driven controllers, the supervisory system collects information, and manipulates recipes to compensate for process drifts.

However, equipment controllers do not diagnose the cause(s) of the problem, and used alone by themselves, could make a process become unstable. Therefore, we have developed a diagnostic system and coupled it to the controller. The diagnostic system is not designed to replace troubleshooting technicians, but rather to help the operator diagnose problems that degrade machine performances, so that they can be solved properly.

1.2 Thesis Objective and Contribution

This thesis describes the development and the deployment of a generic controller and diagnostic system for a sequence of interrelated processes (Figure 1.1). The goal of the system is to provide an economical way of increasing process capability through innovative use of statistical techniques and probability theory.

We have purposefully chosen to use well known statistical techniques and optimization algorithms, instead of heuristics, to support the control schemes. Then, the resulting control methodology can be applied to any process sequence, and its accuracy can be properly quantified.

The diagnostic system, which compliments and is activated by the control system, has also been developed so that it can be applied on any machine. It is based on conventional probability theory, because its mathematical foundations are rigorous, and its assumptions are valid in most process domains. The main novelty of our diagnostic system is that it incorporates both shallow and deep level information as evidence, so that any evidence can be used to diagnose faults. Typically, current diagnostic systems only handle one type of information (i.e, either shallow level or deep level only), which limits their diagnosis capabilities.

Our demonstration vehicle is the photolithography process sequence. We have implemented both control and diagnostic systems on real photolithography equipment, and the experimental results are very encouraging.

1.3 Thesis Organization

The organization of this dissertation is as follows. Chapter 2 describes the experimental setup. Chapters 3 and 4 address the monitoring system and the equipment models, respectively. Chapter 5 describes the run-by-run control system, and chapter 6, the diagnostic system. Finally, chapter 7 presents experimental results of the control and diagnostic system.

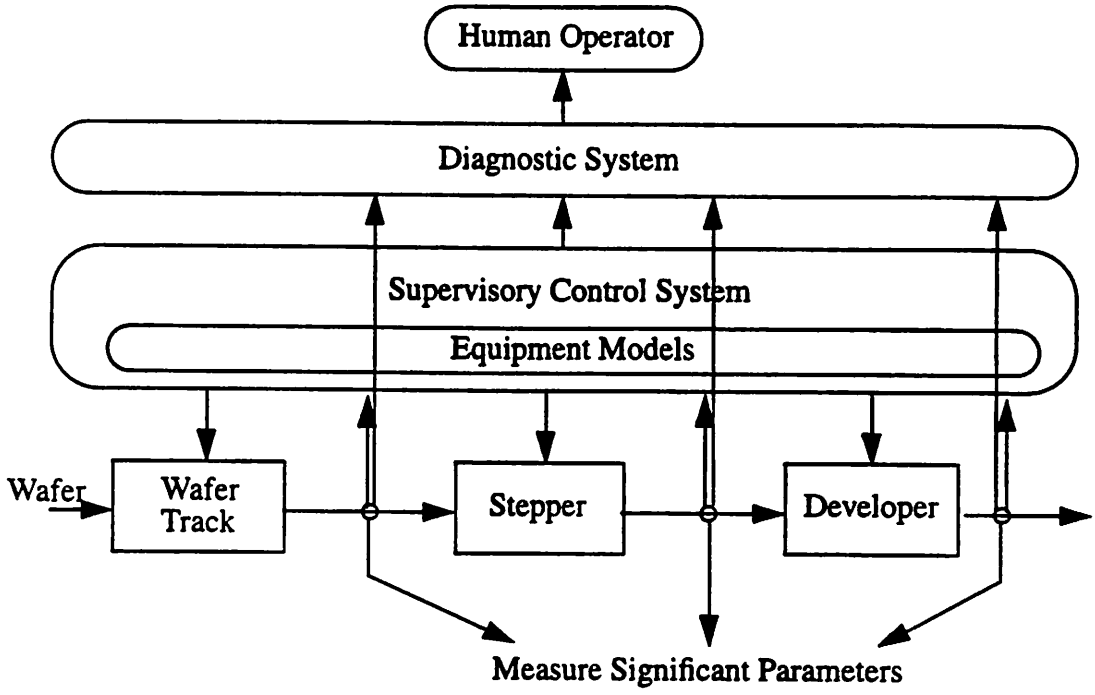


Figure 1.1 Schematic of the Control and Diagnostic System for the Photolithography Sequence.

[This page is intentionally blank]

Chapter 2 Description of the Experimental Setup

2.1 Introduction

This chapter describes the experimental setup of our system. First, a short introduction to photolithography is given, since the goal of our controller is to improve the capability of that process. Then, each equipment of the process sequence is described along with its inherent capabilities. Finally, the test patterns used to test the capability of our system are described.

2.2 Brief Summary of Photolithography

Lithography is the process of transferring geometric shapes from a mask to a silicon wafer. These shapes make up the parts of the circuit, such as gate electrodes, or metal interconnects. In the first step of lithography, a photosensitive polymer film, called photoresist, is applied onto the silicon wafer and then dried. That first step is carried out here, in the Berkeley Microfabrication Laboratory (henceforth called *Microlab*), by a wafer track which spin-coats the wafer with photoresist, and then bakes it at a specific temperature for a predetermined length of time. Next the wafer gets exposed through a photomask with the proper geometrical patterns to ultraviolet light or other radiation. If ultraviolet light is used, the process is then called *photo*lithography. That step is carried out here by a wafer stepper which steps across the wafer, and exposes a small area, called die, using a particular light wavelength, exposure time and focus. Finally, after exposure, the wafer is placed in an ambient that develops the images in the photosensitive material. Depending on the type of polymer used, either the exposed or unexposed areas of the film are removed in the developing process. Since we are using a positive photoresist, the exposed areas are the ones that are removed. The developer solution can either be gaseous (dry development) or liquid (wet development). Our developer uses wet development. Finally, the next step is

etching where the wafer is placed in an ambient that etches the surfaces that are left unprotected by the photoresist patterns [1].

Currently in industry, the first three steps of photolithography, spin-coat and bake, exposure, and development, are lumped together as one single process, called the photolithograhy process sequence. The etching step is typically considered a separate step from the sequence. In our work, we attempt to control the photolithography process sequence by breaking it down into the three steps described above, and monitoring each step separately. The monitoring equipment are described along with the process equipment in the next section.

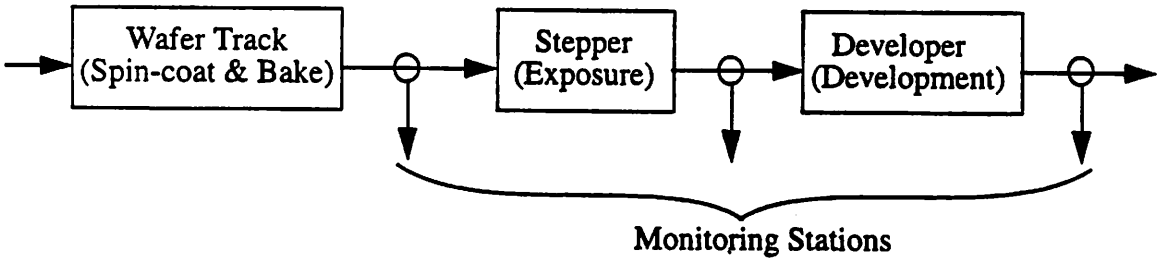


Figure 2.1 Process Flow of our Experimental Setup.

2.3 Equipment Description

This section gives a detailed operational description of the photolithography equipment used in our laboratory, including their capabilities and limitations.

2.3.1 The Spin-Coat and Bake Equipment (or Wafer Track)

The first equipment in the photolithography sequence is the Silicon Valley Group 8626/36 Coater Bake Track System. It is designed to spin-coat and bake 4" wafers [2]. It consists of a chuck that can spin wafers at different speeds, a photoresist dispenser, a hot baking plate, and a cold plate. The duration of each step is controlled by an internal computer in increments of one second. The wafers are loaded into the equipment in a cassette containing up to 24 wafers and are processed individually one after another.

During the coating operation, the wafer is held to the chuck by a vacuum. The spin speed of the chuck can be set from 0 to a maximum of 7000 RPM, in increments of 100 RPM. The chuck actual spin speed is within 20 RPM of the set value. The photoresist dispenser can be moved so that it can dispense the resist at any radial position on the wafer. Throughout our experiment, we dispense the resist at the center of the wafer. The hot bake plate can be programmed in increments of 1°C, but the actual temperature is kept within ±2 °C of the set value. The cold plate is set at room temperature which varies between 20°C and 27°C.

2.3.2 The Stepper

The next equipment in line is the 10X reduction GCA stepper, model 6200 [3]. It is also configured to handle 4" wafers, which are loaded in batch mode. The wavelength of the light source is 365 nm (I-line). The numerical aperture (NA) of the GCA 6200 is 0.32, and its partial coherence parameter (σ) is 0.5. The GCA 6200 is a fully automated stepper that handles one wafer at a time. The stepper has an embedded controller that keeps the dose constant throughout the life of the lamp. The inputs to the stepper are focus and exposure time, which controls the input dose. The focus is measured in μm and is adjusted in increments of 0.5 μm. The exposure time is measured in seconds, and can be controlled to within ±0.01 second.

2.3.3 The Developer

After the wafer is exposed, it is post-baked on the wafer track and then developed by the Silicon Valley Group 8632 Developer Track [2]. Like the other two machines, it is also made to handle 4" wafers and processes one wafer at a time, with the wafers loaded in batch mode. Up to three liquids can be used in the wet process. The SVG 8632 can therefore serve as a developing station as well as a photoresist stripping station. The SVG 8632 is linked to a computer where its recipe programs are stored. Although many parameters can be changed such as the spin speed of the chuck, only the development time has been

varied in our control scheme. The development time can be controlled in increments of one second.

Now that the photolithography machines have been described, the machines used in our monitoring scheme will be discussed.

2.3.4 The Photospectrometer for Reflectance Measurement

The photospectrometer used for collecting reflectance spectrographs is the Inspector by SC Technology [4]. The photospectrometer is run by a software called INS801UV, stored in an accompanying 486 33 MHz personal computer. (The personal computer needs to be at least a 386 25 MHz machine for proper operation.) The photospectrometer measures the reflectance of a wafer from 320 nm to 620 nm, using a xenon light source. It is capable of measuring the thickness of a single layer of resist, oxide, or polysilicon film, as well as two-layer film systems of resist, oxide, and polysilicon. The precision of the readings is heavily dependent on setup operations, which include using very clean reference wafers. The system can be set up to measure wafers in an automated fashion, by using a trigger to start the measurement. We installed a sensor that is activated by the raising of the cold plate of the wafer track. That activates the Inspector which measures a reflectance spectrograph of the wafer and stores it in the hard drive of the personal computer. It is also possible to have the data stored in a logical drive, which is what we have chosen to do. A logical drive is a virtual drive that is linked to the drive of our system server through NFS [5]. That permits data measured on a run time basis to be collected and stored automatically in our system database.

2.3.5 The Critical Dimension (CD) Measurement Computer

The instrument used for measuring CDs is the Nanoline IV Critical Dimension Computer, made by Nanometrics [6]. This instrument measures the CD by scanning horizontally across a portion of the wafer. The reflectance plot is then displayed on the computer

screen. The photoresist covered regions have higher reflectance than the rest of the wafer. The CD is measured at the 35% level of the maximum reflectance, since for our system parameters, measurements are least sensitive to defocus at that particular threshold. This level can be adjusted if needed. The Nanoline system also incorporates several other specific programs, such as one that measures the pitch. The reliability of the Nanoline depends heavily on the type of lens used. The strongest lens (100X) allows the most reliable data collection. The Nanoline measures the CD in μm, with a stated precision of ±0.01 μm. Experimentally, we find that the Nanoline precision is limited to ±0.03 μm.

2.4 Description of the Test Patterns

To obtain reliable readings, the test patterns and measurement locations must be carefully chosen. Although the photoactive compound concentration (PAC) is independent of measurement location, the resist thickness is radially dependent on position. The photoresist is thicker in the middle of the wafer and thinner at the edges [60]. Therefore, to obtain repeatable thickness measurements, measurements are taken at locations that lie on the same radius (Figure 2.2).

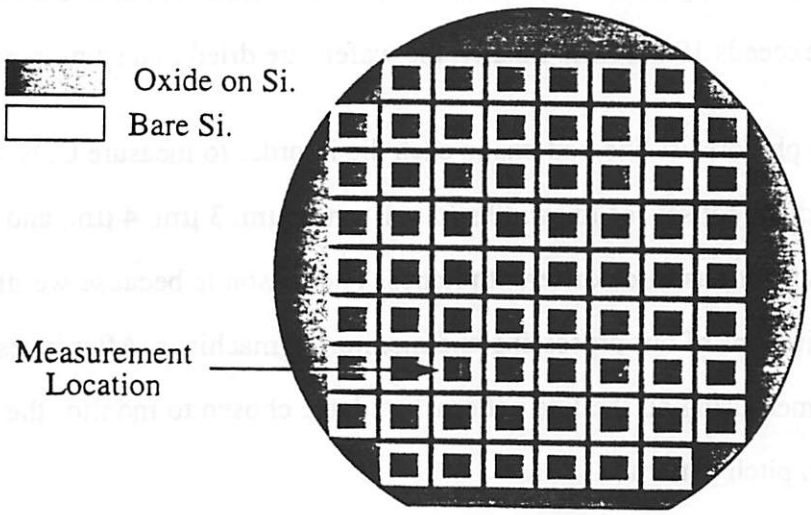


Figure 2.2 Measurement Locations on a Test Wafer.

To obtain the above pattern, we first grew 1000 Å of thermal oxide on Si wafers. Our samples included a random mixture of both <111> and <110> wafers. These wafers were spin-coated with a positive photoresist (OCG 820) at 4600 RPM for 30 seconds, soft-baked at 120°C for 60 seconds, and exposed with a I-line stepper using the mask shown in Figure 2.3. Then we post-baked the wafers at 120°C for 60 seconds, and developed the pattern. This procedure results in a wafer patterned as in Figure 2.2.

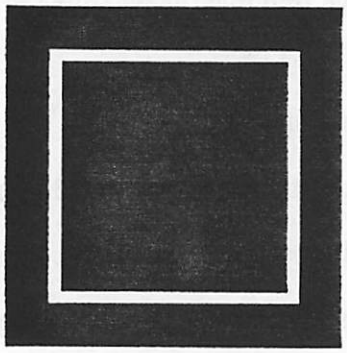


Figure 2.3 Mask for the Wafer Test Pattern.

Before any processing, all wafers are cleaned and dehydrated in a convection oven for 30 minutes. The cleaning procedure involves fully immersing the wafers for 10 minutes in a piranha sink, and then rinsing them in DI water for around 15 minutes, until the cleanliness of the DI water exceeds 10 MΩ·cm. Finally, the wafers are dried in a spin-dryer.

Next, we develop photoresist line patterns in each die in order to measure CDs. These line patterns contain different sets of linewidths (d = 1 μm, 2 μm, 3 μm, 4 μm, and 5 μm (Figure 2.4)), and each set has three different pitches. The reason is because we did not know a priori the limitations of our processing and monitoring machines. After investigating the reliability of measuring all the line patterns, we have chosen to monitor the 2 μm linewidth with a 4 μm pitch pattern.

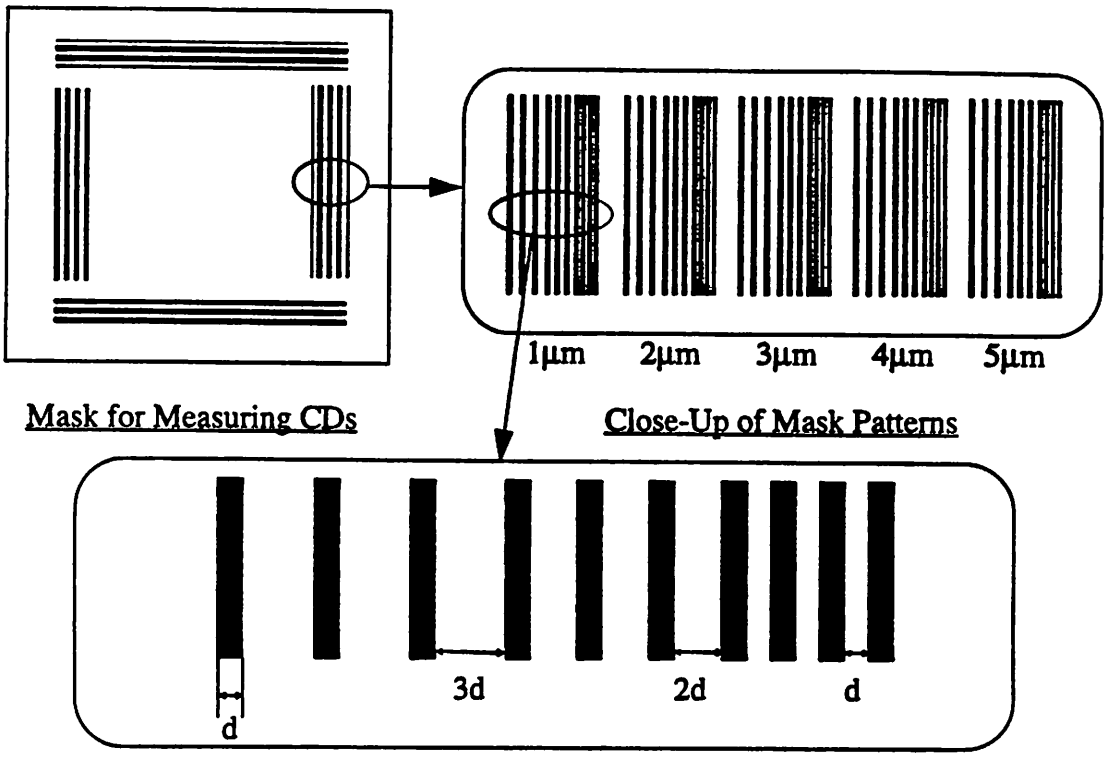


Figure 2.4 Mask Test Pattern

2.5 Summary

This chapter described the experimental setup used to test our control and diagnostic system, which consist of the equipment used in the photolithography sequence, and the procedures for preparing the test wafers. The intention was to give the reader a better understanding of the capabilities of our equipment, so that the capabilities of our control and diagnostic system could be fully appreciated, when they are described in subsequent chapters. Next, we describe how the equipment models are generated.

[This page is intentionally blank]

Chapter 3 Equipment Models

3.1 Introduction

The control and diagnostic system relies on equipment models to characterize the behavior of a piece of equipment, control it and diagnose problems affecting it. There are two approaches to developing equipment models, an empirical one and a physical one. Each approach has its own strengths and weaknesses.

In the first stages of this work, equipment models have been developed using the empirical approach known as Response Surface Modeling [22]. They are easily built, and give highly accurate predictions of the machine's outputs [7]. One caveat however is that the empirical models do not give us much insight about the process. Another one is that they cannot be updated when a change of a machine component has caused the machine outputs to shift beyond the experimental range of the empirical models. In such cases, designed experiments must be run again in order to create new empirical models.

The merits of using a physical approach when creating equipment models include a better understanding of the process, quicker acceptance by the process engineering world, and improved robustness. When a new process is developed, the physical parameters that affect it are well documented by engineers. Therefore, using those parameters in the models makes sense to process engineers. Furthermore, because physical models explain the complete range of the process, new machine components that cause the machine outputs to shift significantly will not render these models obsolete. On the other hand, the caveats of the physical approach are that they are very hard to develop and sometimes, are not as accurate as empirical models.

For our system, our philosophy has been to use physical models whenever their accuracy is not significantly worse than that of the empirical models. Now, since our control

and diagnostic system is focused on photolithography, we describe the parameters used for modeling the photolithography process.

3.2 Relevant Photolithography Parameters

To characterize the state of the wafer after each photolithography step, the following parameters are of interest: photoresist film thickness and chemical properties, numerical aperture (NA) of the stepper lens, exposure dose, and develop time [13]. The photoresist's chemical properties consist of the index of refraction n, and the absorption coefficient k, which depends on the photoactive compound concentration (PAC) inside the resist [15] and Dill's A, and B parameters [14] [15].

$$k = \lambda \frac{A \times PAC + B}{4\pi} \tag{3.1}$$

A is the net absorption of the inhibitor; B, the net absorption of the base resin; and λ is the light wavelength. Both A, and B parameters are also functions of λ [16].

3.3 Photolithography Modeling Parameters

Among these photolithography parameters, we must choose a set of independent parameters that provides a complete picture of the process status. We have chosen to restrict our interest only to parameters that change significantly with time, to facilitate the metrology. The other parameters are considered as fixed constants. A study of each parameter has led us to monitor only resist thickness, T_{res}, and PAC (the PAC metrology is discussed in Chapter 4). These two parameters characterize the status of a wafer well after the spin-coat and bake process, while the change in PAC before and after exposure, ΔPAC, characterizes well the exposure step. After development, we measure the linewidth dimensions (CD).

The other parameters, namely Dill's A, and B parameters, the films' index of refraction and the NA of the stepper lens are considered constant with respect to time. The

dependency of index of refraction on wavelength however is being considered in the calculations. Several handbooks show the following function for n [6]:

$$n = Na + \frac{Nb}{\lambda^2} \tag{3.2}$$

where Na and Nb are called the Cauchy coefficients of the film.

The correlation between T_{res} and PAC has been studied (Figure 3.1). The low correlation coefficient of 0.30 confirms our belief that the two parameters are relatively independent of each other, and therefore both should be monitored in order to track the status of the photoresist.

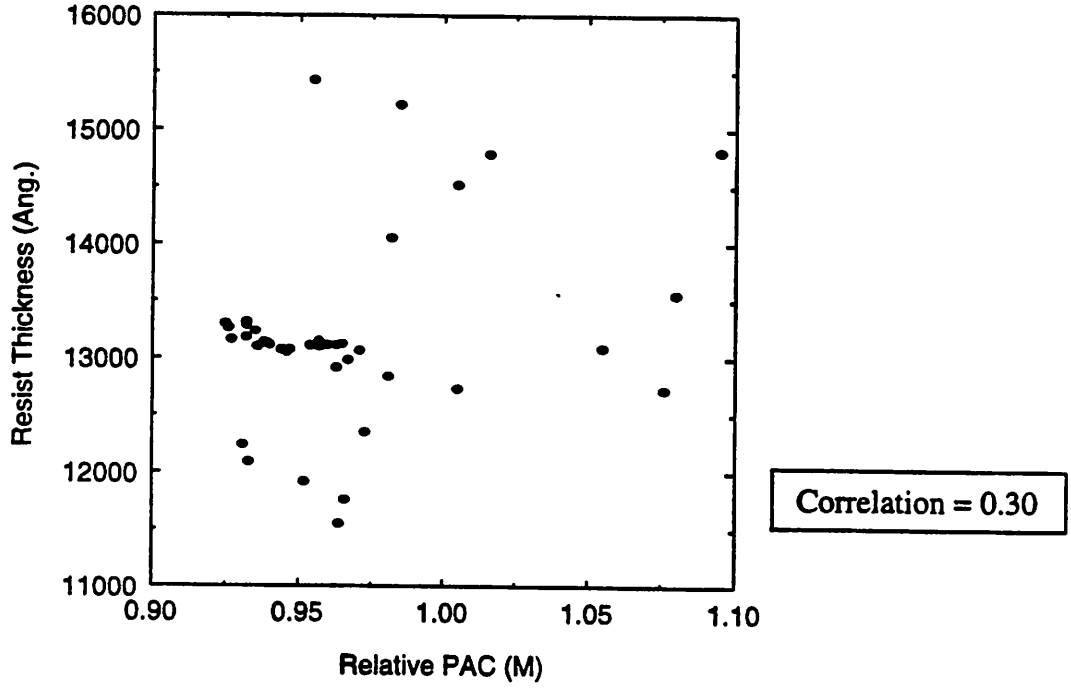


Figure 3.1 Correlation Plot between Resist Thickness and PAC.

3.4 Equipment Model for the Wafer Track

As mentioned in the previous section, modeling the spin-coat & bake process, i.e modeling the wafer track, can be reduced to modeling the resist thickness and PAC. In the literature, the spin-coat and bake process is typically modeled from a fluid dynamic perspective, where the resist thickness has been solved for in terms of spin speed, acceler-

ation, and film viscosity [60]. Although the derivation is very rigorous, the model does not fit experimental results very well, because it has not taken into account the solvent evaporation during the spinning and soft-bake processes. Therefore, an empirical response surface model (RSM) was developed for characterizing photoresist thickness [22], using a statistically designed factorial experiment. As for the PAC, since no previous work on modeling it has been done, we have also developed an empirical RSM model for characterizing it.

The inputs of the equipment models consist of the chuck's spin speed and spin time, the soft bake plate temperature, and the soft bake time, and the outputs are resist thickness and PAC. Since the equipment model of the wafer track has four inputs, the factorial experiment required 16 runs, to which we added eight runs at the standard operating point to help determine the replication error of the process. The experimental settings are shown in Table 3.1.

Table 3.1 Range of Modeling Experiment for the Spin-coat and Bake Equipment.

Input Factors	Lower Setting (-)	Std Setting (0)	Higher Setting (+)
Spin Speed	3600 RPM	4600 RPM	5600 RPM
Spin Time	15 secs	30 secs	90 secs
Soft-Bake Temperature	75 °C	90 °C	105°C
Soft-Bake Time	20 secs	60 secs	100 secs

3.4.1 Resist Thickness Model

The following regression model has been developed for resist thickness [23]:

$$T_{RES} = 1291.98 + \frac{928233}{\sqrt{SPS}} - 1.62BTI - 19.49BTE \qquad (3.3)$$

where SPS represents spin speed in RPM; BTI, baking time in seconds; BTE, baking temperature in degrees Celsius; and T_{RES}, resist thickness in Angstroms. The non-linear trans-

formation was derived from the physical model in [60] and confirmed by residual analysis of the spin speed parameter.

The Analysis of Variance (ANOVA) table [22] of the resist thickness model is shown in Table 3.2. A scatterplot of the resist thickness values predicted by the model versus the corresponding experimental values is given in Figure 3.2. The closer the experimental data are to the y=x line, the more significant the model is.

Table 3.2 ANOVA Table for the Thickness Model of the Wafer Track

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	42	24309606			
Regression	3	24133278	8044426	1779.3	0
Residual	39	176327	4521		
Lack of Fit	27	170587	6318		
Error	12	5740	478		

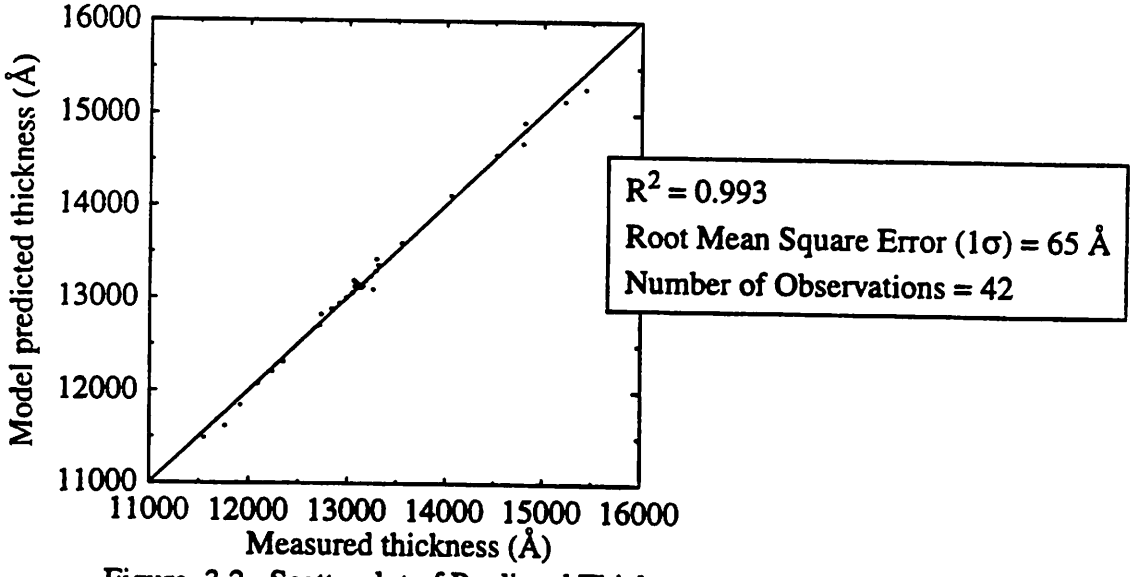


Figure 3.2 Scatterplot of Predicted Thickness vs. Actual Thickness

This resist thickness model has later been improved by incorporating effects of relative humidity and resist bottle level, using a long range monitoring experiment done by the

staff of the Microlab. The staff processed wafers using the same standard settings for one and a half year, enabling us to study the effects of relative humidity, air temperature, and the amount of resist left in the bottle dispenser on thickness. The effect on the PAC however could not be investigated, since its metrology has not been developed yet.

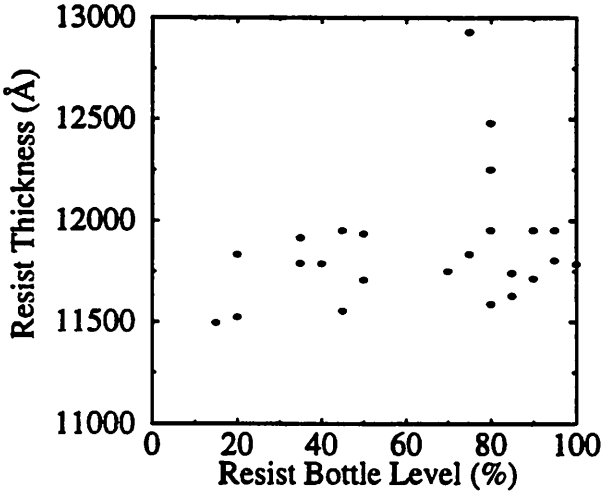


Figure 3.3 Resist Thickness vs. Resist Bottle Dispenser Level.

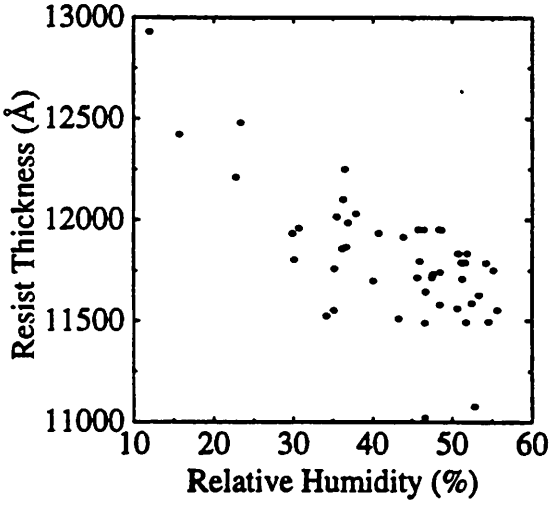


Figure 3.4 Resist Thickness vs. Relative Humidity

Table 3.3 Correlation Between Environmental Variables and Thickness

	Humidity	Air Temperature	Bottle Level
Thickness	-0.79	0.17	0.28

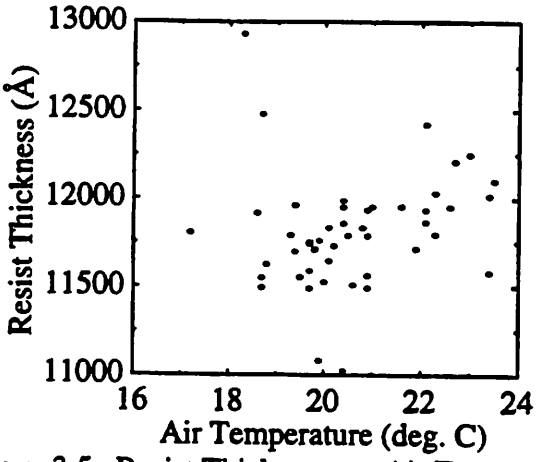


Figure 3.5 Resist Thickness vs. Air Temperature

The resulting thickness model is:

$$T_{RES} = 13657.926 - 21.98H + 1.785BL \tag{3.4}$$

where H corresponds to the relative humidity in %, and BL, to the fraction of resist left in the bottle in %. Its ANOVA table confirms that the model is indeed significant, and the model's fit is summarized in Table 3.5.

Table 3.4 ANOVA Table for the Thickness Model Based on Environmental Factors

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	24	2277974			
Model	2	1475673	737836	19.31	1e-5
Residual	22	802301	38205		

Table 3.5 Summary of Fit for the Thickness Model Based on Environmental Factors

R^2	0.65
Root Mean Square Error (1σ)	195 Å
Number of Observations	24

Assuming that environmental parameters do not interact with machine settings, we have combined this model with the previous one developed from the machine settings:

$$T = 1770.86 + \frac{928233}{\sqrt{SPS}} - 1.62BTI - 19.49BTE - 21.98H + 1.785BL \qquad (3.5)$$

The root mean square error of this model is:

$$s_T = \sqrt{\frac{s_1^2 \cdot df_1 + s_2^2 \cdot df_2}{df_1 + df_2}} = \sqrt{\frac{65^2 \cdot 39 + 195^2 \cdot 22}{39 + 22}} = 128 \text{ Å} \qquad (3.6)$$

where s_1 and s_2 are the standard errors of each model, and df_1 and df_2, their respective degrees of freedom.

To check whether it is actual resist aging or decreasing vapor pressure in the bottle that affects resist thickness, the Microlab staff has experimented with another way of dispensing photoresist: they have purchased photoresist packaged in a pouch and dispense it upside down. This experiment also lasted a year. The scatterplot of resist thickness vs. weight of resist pouch is shown below. Since the two variables do not show much correlation, we conclude that it is decreasing vapor pressure in the resist bottle that affects the thickness, and not resist aging.

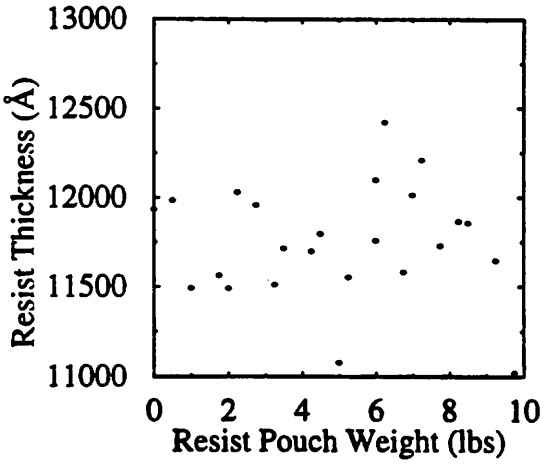


Figure 3.6 Resist Thickness vs. Resist Left in Pouch

3.4.2 Photoactive Compound Concentration (PAC) Model

When unexposed resist is processed under normal conditions, it has a certain absorption. That absorption is quantified by Dill's A and B parameters, assuming a *relative* value of PAC, called M, of 1.0. Under different processing conditions, the absorption can change. We model that as a change in the *relative* value of PAC, which can either exceed

or drop below 1.0. From the same factorial experiment used in developing the resist thickness model, we have developed the following *relative* PAC model for unexposed photoresist (the PAC metrology is discussed in Chapter 4):

$$M = 0.91 + 1.61 \cdot 10^{-3}BTE - (2.10 \cdot 10^{-5})SPS \quad (3.7)$$

Although the F-test shows that the model is significant (F(3, 37) > 7.3e-9), this PAC model is not very precise: its R^2 is only 0.57.

The ANOVA table of the PAC model is summarized in Table 3.6, and the scatterplot of predicted PAC vs. measured PAC is shown in Figure 3.7.

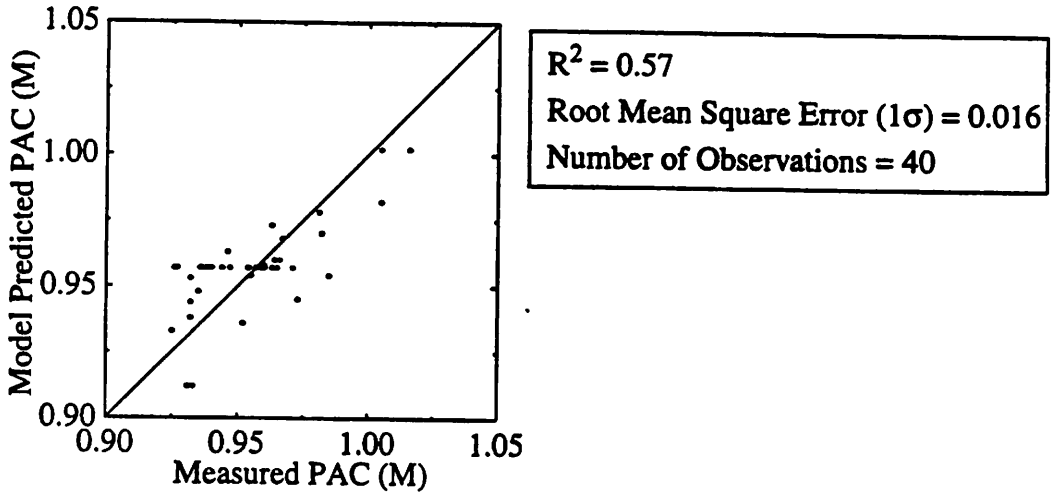


Figure 3.7 Scatterplot of Predicted PAC vs. Measured PAC

Table 3.6 ANOVA Table for the PAC Model of the Wafer Track

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	40	0.02044			
Regression	3	0.01160735	0.005804	24.3116	< 7.3e-9
Residual	37	0.00883265	0.000239		
Lack of Fit	18	0.00595487	0.000331		
Error	19	0.00287778	0.000151		

3.5 Equipment Model for the Stepper

As mentioned in §3.3, the relevant output from the exposure stage is the relative PAC value (M) after exposure. Actually, M varies within the layer of photoresist, due to the exposure intensity interference patterns. However, since our metrology effectively measures the absorption coefficient k, from which we derive M, we cannot measure M as a function of depth, and we are limited to using an average M value for the whole film. The inputs of the exposure model, or stepper model, include T_{RES}, before exposure PAC (actually M), and dose. Although defocus should also be considered among the inputs, we have not included it yet, because we do not have an economical run-by-run sidewall slope metrology, which is the effect of defocus on CD. Such a metrology is currently being investigated [58] [61]. As soon as it is successful, defocus will be added as an input of the stepper equipment model.

3.5.1 Physically-Based Stepper Model

Unlike the spin-coat and bake step, the exposure step has been rather thouroughly investigated. Many exposure models exist, and can be found in photolithography process simulators. For example, SAMPLE [13] employs state-of-the art physical models of exposure and development, that could be very useful to our control and diagnostic system. And if SAMPLE's outputs do not correspond exactly to our machine's outputs, they can easily be fitted to the machine's measured outputs through a simple empirical model.

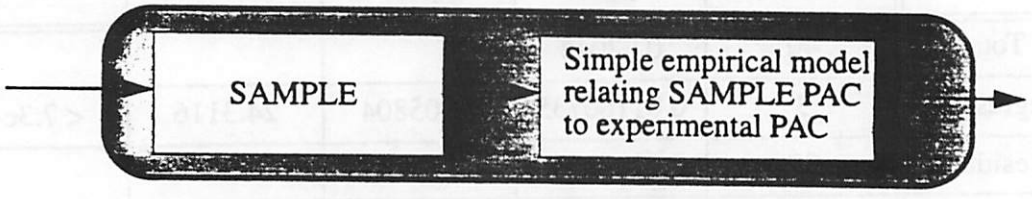


Figure 3.8 Schematic Representation of the Stepper Equipment Model

SAMPLE simulations return an M matrix which shows the amount of exposure in each region of the photoresist. Our PAC metrology, which will be described in Chapter 4, can-

not measure the full M matrix however, because it actually measures the absorption coefficient k and derives from it, an "average" value of M. Therefore, as a first order approximation, we have averaged the M values given by SAMPLE and compared it to our measured value of M. A scatterplot of the two parameters is shown below.

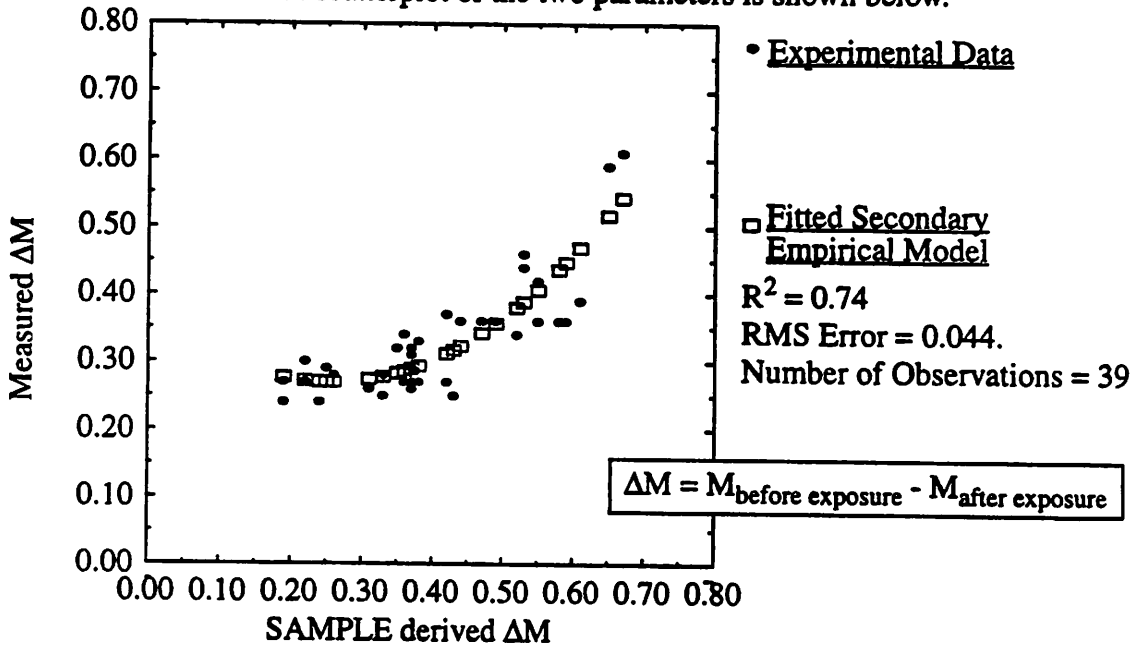


Figure 3.9 Measured ΔM vs SAMPLE derived ΔM

The fitted regression model follows the equation below, and its ANOVA table is presented in Table 3.7:

$$\Delta M_{Meas} = 0.3636 - 0.773\Delta M_{SAMPLE} + 1.554\Delta M_{SAMPLE}^{2} \quad (3.8)$$

Table 3.7 ANOVA Table for the Exposure Model Derived from SAMPLE

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	39	0.27253467			
Regression	3	0.20304031	0.1015	52.59	3e-13
Residual	36	0.06949436	0.00193		

3.5.2 Empirical Stepper Model

Since the prediction error of the model (±0.044) is rather large, we have tried to fit an empirical RSM model to the data, using T_{RES}, $M_{unexposed}$, and dose as inputs. That approach has resulted in the following equipment model:

$$M_{exposed} = M_{unexposed} - 0.64 - 0.000909D + 0.0000112T_{RES} \quad (3.9)$$

where D represents dose in mJ/cm^2. Although its R^2 is lower, this model is more accurate than the previous one, which is based on the physical model. We believe the reason is because *averaging* the M values from SAMPLE's M matrix does not result in the correct "equivalent" M value. An investigation should be done to find the proper filtering method that would result in a more correct M_{SAMPLE} to M_{meas} transformation. The empirical model's ANOVA table is presented in Table 3.8, and a scatterplot of the new model's prediction versus the experimental data is shown below.

Table 3.8 ANOVA Table for the Stepper Model Obtained from Regression Analysis.

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	39	0.12147337			
Regression	3	0.07872493	0.039362	34.07	1.3e-10
Residual	36	0.04274844	0.001155		

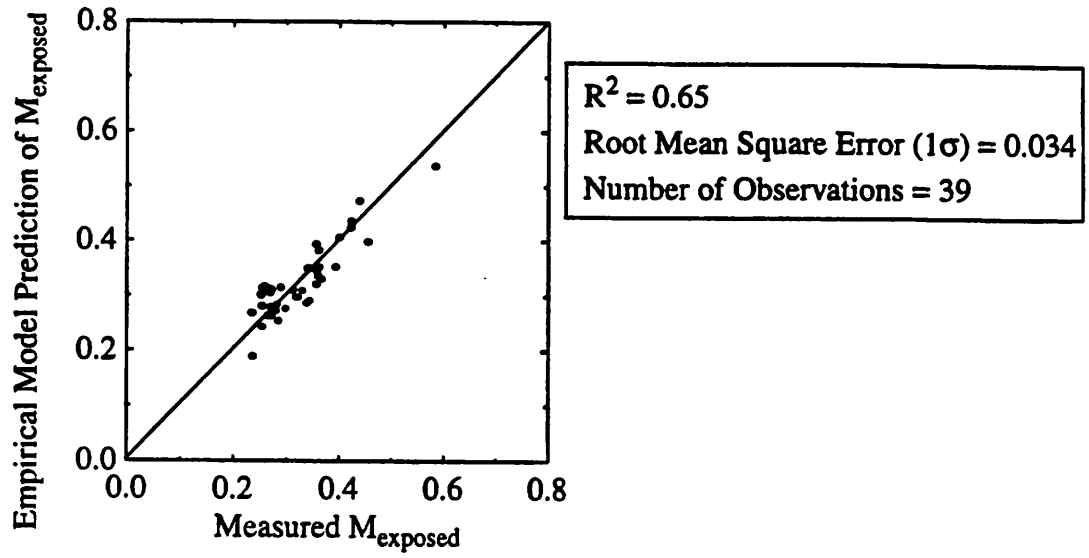


Figure 3.10 Scatterplot of Empirical Stepper Model Prediction vs. Measured PAC

Ultimately, although the model which uses the physical models embedded in SAMPLE is theoretically more robust, we have adopted the empirical one instead for our control and diagnostic system, because of its superior accuracy. Had both approaches lead to models with similar performances, the one based on physical models would have been preferred, since it is theoretically more robust.

3.6 Equipment Model for the Developer

The inputs of the developer model consist of T_{RES}, $M_{exposed}$, and develop time, D_t. The output of the model is CD. Ideally, the slope of the sidewalls should also be included among the outputs, but as previously explained, we lack an economical metrology for it.

3.6.1 Physically Based Developer Model

There exists a physical model for the develop step within the photolithography process simulator SAMPLE, and we use it in a similar fashion as we used the physical exposure model (Figure 3.11).

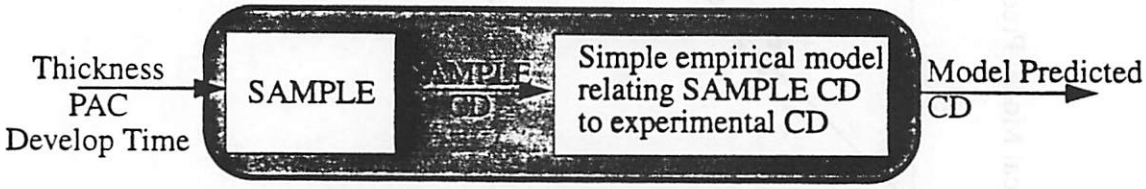


Figure 3.11 Schematic Representation of the Developer Equipment Model

The simple regression model that links the measured CD to SAMPLE's CD is shown in equation (3.10), its ANOVA table is presented in Table 3.9, and a scatterplot of the model predicted CDs vs. the measured CDs is shown in Figure 3.14.

$$CD = 1.286CD_{SAMPLE} - 1.107 \tag{3.10}$$

Table 3.9 ANOVA Table for the Developer Model Derived from SAMPLE

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	35	0.71265714			
Regression	2	0.59127869	0.591279	160.76	9.7e-18
Residual	33	0.12137845	0.003678		

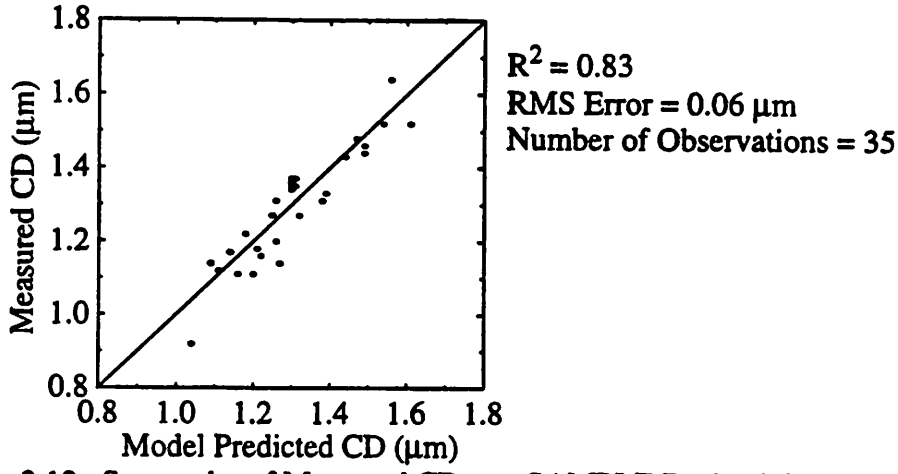


Figure 3.12 Scatterplot of Measured CDs vs. SAMPLE Derived CDs

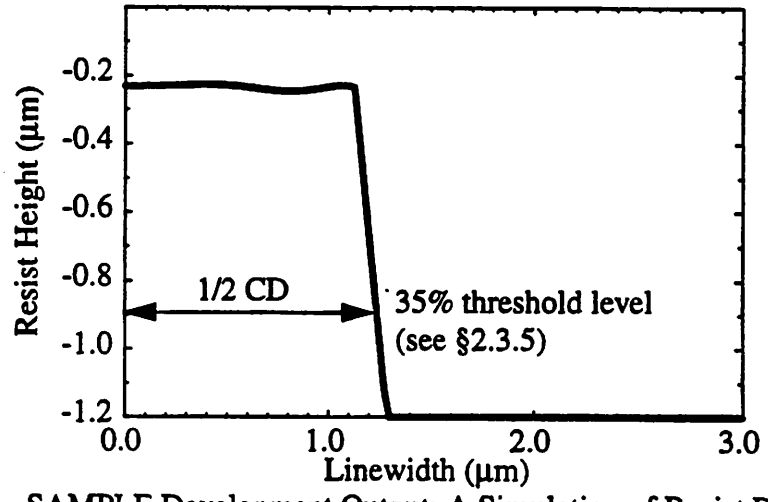


Figure 3.13 SAMPLE Development Output: A Simulation of Resist Profile

3.6.2 Empirical Developer Model

Now, we look into the possibility of using a pure regression model to fit the experimentally measured CDs. The resulting model is:

$$CD = -3.503 + 0.00034T_{RES} + 18.25M_{exposed} - 0.0013(T_{RES} \cdot M_{exposed}) \quad (3.11)$$
$$-0.11(M_{exposed} \cdot D_t) + 7.9 \cdot 10^{-6} \cdot T_{RES} \cdot M_{exposed} \cdot D_t$$

where D_t represents develop time in seconds. This time, the empirical model is much worse than the physically based one (Table 3.10). Its prediction error is 0.11 μm (Figure

3.14), twice that of the last model's and significantly larger than the accuracy of the CD metrology (see §2.3.5).

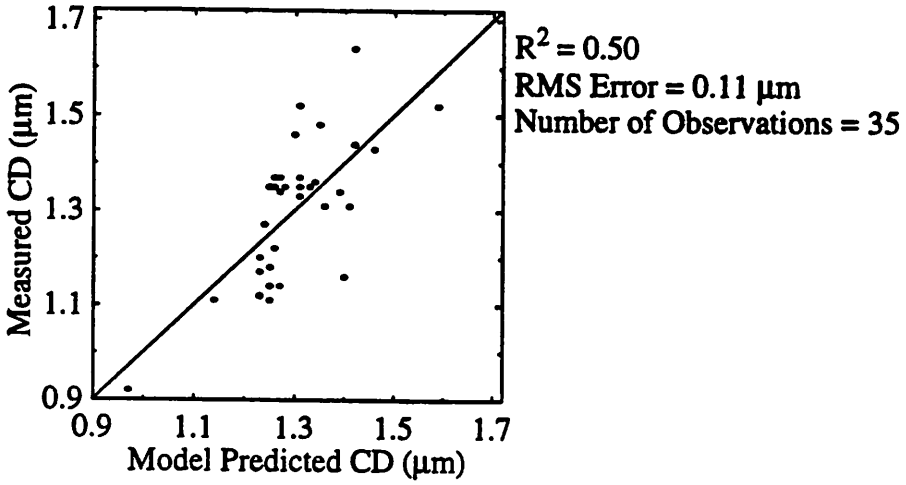


Figure 3.14 Scatterplot of Measured CDs vs. Model Predicted CDs

Table 3.10 ANOVA Table for the Empirically Derived Developer Model

Source	Degrees of Freedom	Sum of Squares	Mean Square	F-Ratio	Significance
Total	35	0.71265714			
Regression	6	0.35834366	0.071669	5.866	0.0004
Residual	29	0.35431348	0.012218		

This time, since the physically based developer model is significantly more accurate than the empirical one, it is the one used in our control and diagnostic system.

3.7 Summary

Equipment models have been developed for each step of the photolithography sequence. The inputs of the equipment models consist of the machine input settings, previous machine's outputs, and modelable environmental parameters. Their outputs are economically measurable parameters that have been carefully chosen to reflect the status of the process. We have investigated models that are based on physical process simulators and models that are based on stepwise regression analysis. Prediction accuracy is the main criterion for selecting which model to use in our control and diagnostic system. In each

case, it has been sufficient to clearly determine the superior model. The wafer track's and the stepper's equipment models are based on empirical regression models, while the developer's is based on a physical process simulator, SAMPLE [13]. If the physically based models could predict the machines' output(s) as accurately as the empirical models, they would take preference, because of their robustness and insight. Next, we illustrate how we measure the parameters used in our equipment models.

[This page is intentionally blank]

Chapter 4 Metrology for Photolithography Process Control

4.1 Introduction

In order to build the previously described equipment models, metrologies for the photoresist thickness, PAC and CD must be developed. While the first and last ones are easily measured through a photospectrometer, a novel metrology is needed for measuring PAC. In this chapter, we will present such a metrology and characterize its capabilities.

4.2 A New Metrology for Characterizing PAC

Presently, both photospectrometry and ellipsometry are capable of measuring resist thickness, but not PAC [83]. To measure the latter parameter, we expand on photospectrometry, by improving the analysis of a reflectance spectrograph. The concept of the metrology, depicted in Figure 4.1, is as follows: after measuring a reflectance spectrograph of a wafer, a theoretically derived reflectance spectrograph is fitted onto it through an optimizer [17][18]. The T_{res} and PAC values used in the curve that best fits the experimental data constitute the measurement results.

The advantage from using this metrology for measuring thickness comes from the fact that we are also solving for the optimal index of refraction of the photoresist, when solving for the optimal thickness. This results in a more accurate thickness measurement than the one returned by the commercial photospectrometer, which assumes a constant value for the index of refraction of the photoresist.

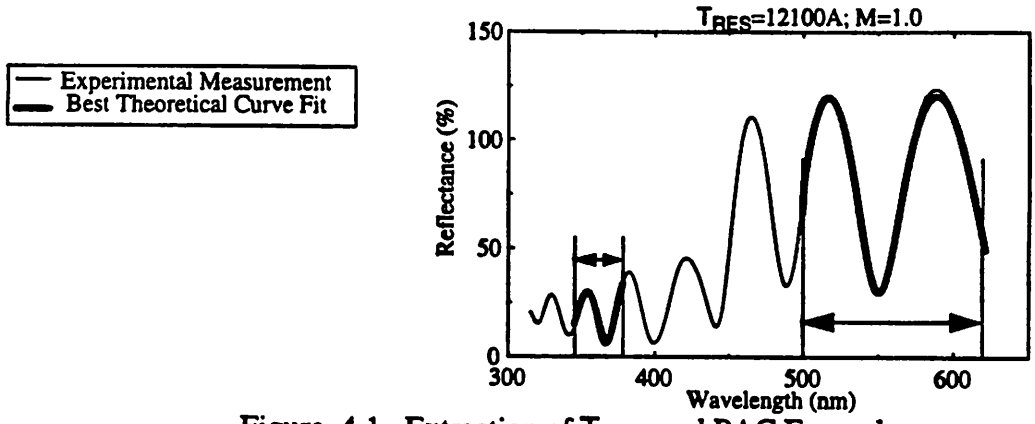


Figure 4.1 Extraction of T_{RES} and PAC Example

4.2.1 Determination of Photoresist Thickness

At high wavelength, the photoresist absorption of light is very close to zero, because Dill's A and B parameters are essentially zero (equation (3.1)). Therefore, since the level of PAC will not alter the reflectance of the light at high wavelengths, solving for the best curve fit at these wavelengths essentially reduces to solving for the photoresist thickness. Besides decoupling the extraction of PAC and thickness, solving for the photoresist thickness at high wavelengths also has the following advantage: photoresist thickness affects the periodicity and values of maxima and minima of the reflectance spectrograph of a wafer [6], and these three characteristic parameters are better defined at high wavelengths. For the OCG 820 photoresist used in our experiment, the high wavelengths at which the photoresist absorption is zero range from 500 to 620 nm. The full theoretical derivation of a reflectance spectrograph is presented by Born & Wolf [17] and is summarized below for wafers with a film stack, consisting of a coating of field oxide underneath a coating of photoresist. We have assumed that the silicon substrate is semi-infinite, i.e, no power is transmitted past the silicon wafer.

The optical properties of a layer of film are described by its *characteristic matrix* $\mathbf{M_c}$. Both transmittance and reflectance of the layer of film can be derived from the compo-

nents of M_c. Describing films through their characteristic matrix becomes very useful when analyzing a stack of films: if two adjacent layers have characteristic matrices M_{c1} and M_{c2} respectively, the characteristic matrix of the stack of films will be $M_c = M_{c1} \bullet M_{c2}$. The matrix M_c is given by:

$$M_c = \begin{bmatrix} \cos(k_0 \cdot n \cdot l) & \frac{1}{i \cdot n}\sin(k_0 \cdot n \cdot l) \\ \frac{n}{i}\sin(k_0 \cdot n \cdot l) & \cos(k_0 \cdot n \cdot l) \end{bmatrix} \tag{4.1}$$

where

$$k_0 = \frac{2 \cdot \pi}{\lambda} \tag{4.2}$$

and l is the film thickness. For our two layer film system, the characteristic matrix is given by:

$$M'_c = \begin{bmatrix} m'_{11} & m'_{12} \\ m'_{21} & m'_{22} \end{bmatrix} = \begin{bmatrix} \cos(k_0 \cdot n_{res} \cdot l_{res}) & \frac{1}{i \cdot n_{res}}\sin(k_0 \cdot n_{res} \cdot l_{res}) \\ \frac{n_{res}}{i}\sin(k_0 \cdot n_{res} \cdot l_{res}) & \cos(k_0 \cdot n_{res} \cdot l_{res}) \end{bmatrix} \bullet \tag{4.3}$$

$$\begin{bmatrix} \cos(k_0 \cdot n_{ox} \cdot l_{ox}) & \frac{1}{i \cdot n_{ox}}\sin(k_0 \cdot n_{ox} \cdot l_{ox}) \\ \frac{n_{ox}}{i}\sin(k_0 \cdot n_{ox} \cdot l_{ox}) & \cos(k_0 \cdot n_{ox} \cdot l_{ox}) \end{bmatrix}$$

The reflectance of the stack of film is given by

$$R = |r|^2 = \frac{(m'_{11} + m'_{12} \cdot n_{si}) \cdot n_{air} - (m'_{21} + m'_{22} \cdot n_{si})}{(m'_{11} + m'_{12} \cdot n_{si}) \cdot n_{air} + (m'_{21} + m'_{22} \cdot n_{si})} \tag{4.4}$$

The parameters with subscript *si* correspond to parameters of silicon; those with subscript *res* correspond to parameters of photoresist; those with subscript *ox* correspond to parameters of oxide.

The expression of reflectance described above is not the one we actually use. When measuring reflectance, we always measure it relative to that of a reference silicon wafer.

This silicon wafer has some native oxide on it. Therefore, when we simulate the reflectance graph of our wafer, we divide the reflectance expression above by the reflectance of a silicon wafer covered with native oxide, which is given below:

$$R'' = |r|^2 = \frac{(m''_{11} + m''_{12} \cdot n_{si}) \cdot n_{air} - (m''_{21} + m''_{22} \cdot n_{si})}{(m''_{11} + m''_{12} \cdot n_{si}) \cdot n_{air} + (m''_{21} + m''_{22} \cdot n_{si})} \tag{4.5}$$

where

$$M_c'' = \begin{bmatrix} m''_{11} & m''_{12} \\ m''_{21} & m''_{22} \end{bmatrix} = \begin{bmatrix} \cos(k0 \cdot n_{ox} \cdot l_{nox}) & \frac{1}{i \cdot n_{ox}} \sin(k0 \cdot n_{ox} \cdot l_{nox}) \\ \frac{n_{ox}}{i} \sin(k0 \cdot n_{ox} \cdot l_{nox}) & \cos(k0 \cdot n_{ox} \cdot l_{nox}) \end{bmatrix} \tag{4.6}$$

and l_{nox} is the native oxide thickness and is assumed to be 35Å.

Given an initial guess of resist thickness taken from the value returned by the photospectrometer [4], a theoretical reflectance curve is derived from these equations. Then, an optimizer is used to find the theoretical curve that best fits the experimental data between the range of wavelengths of 500 - 620 nm. The resist thickness used in that best fit is considered as the actual resist thickness.

4.2.2 Determination of PAC

PAC is derived from the absorption coefficient k of the photoresist (equation (3.1)). Therefore it is found at shorter wavelengths, where photoresist is absorptive. The range of these short wavelengths, given our experimental setup and our brand of photoresist, is 320 - 430 nm. 320 nm marks the low end of the spectrograph of the xenon lamp of the spectrometer, while 430 nm marks the high end of the wavelengths that are absorbed by the photoresist. For the purpose of extracting PAC, we have limited our range of interest to 350 - 380 nm, in the center of which lies the exposure wavelength of 365 nm. The reason we have ignored the 320 - 349 nm wavelengths is because the light intensity of the xenon light source is not stable at those wavelengths; and we have ignored the 380 - 430 nm

wavelengths, because it is a range of transition in the absorption characteristics of the photoresist, where the assumptions underlying our equations are not valid.

Using the thickness value found from the previous step, we fit theoretically derived curves to the experimental reflectance graph in the narrow 350 - 380 nm range. The PAC value used in the best fitting theoretical reflectance graph is considered as the actual PAC value. Theoretically, the PAC value before exposure is 1.0 and 0.0 after full exposure. During actual processing though, the PAC is around 1.0 or slightly less before exposure, and significantly lower, but positive, after exposure. More specifically, we have found that the average PAC before exposure is 0.97 and 0.32 after exposure.

Sometimes though, the extracted PAC before exposure value exceeds 1.0. The reason is as follows: when measuring PAC, we are actually measuring the absorption coefficient k (§3.2). Equation (3.1) shows that the measured PAC value depends on the value used for Dill's A parameter. We have taken the A parameter value from a chemical handbook [16], which assumes that the photoresist is processed around its normal operating point. If the photoresist is not processed under normal conditions, its absorption coefficient can change. When this occurs, the PAC measured using the fixed A value can exceed 1.0. Therefore, since the measured PAC values are relative to the standard value of Dill's A parameter, we call them *relative* PAC values and denote them by the symbol M (not to be confused with the characteristic matrix M_c).

In summary, we have extended the theories underlying photospectrometry, and developed a metrology for measuring resist thickness and PAC. We call this metrology TaME, which stands for T and M Extraction.

4.3 Experimental Results

We evaluate now the performance of TaME by comparing experimental reflectance spectrographs to TaME derived simulated spectrographs. Note that the TaME method has only tried to fit the following portions of the curve: 350 - 380 nm, and 500 - 620 nm.

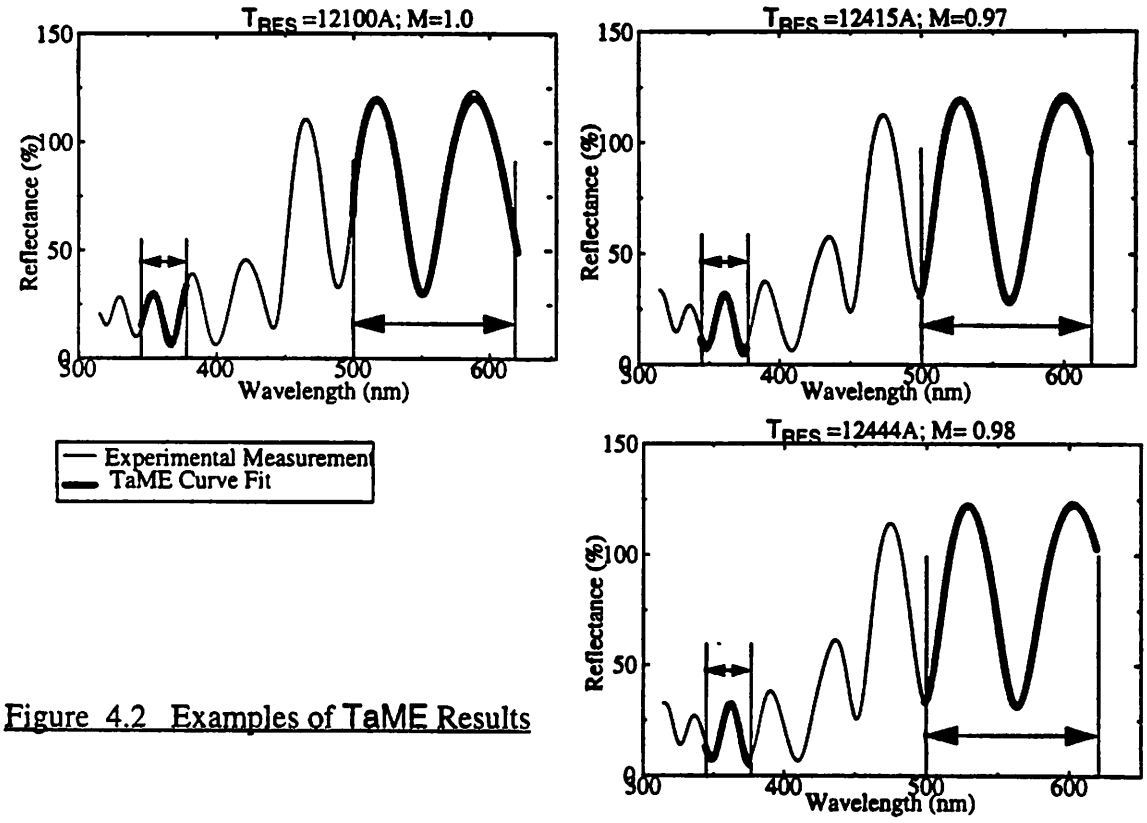


Figure 4.2 Examples of TaME Results

Finally, before closing the discussion on the details of the TaME methodology, we notice that the introduction of an additional "offset" parameter increases the accuracy of the fit. This offset parameter is added to the theoretical spectrograph before fitting it to the experimental data. The justification of such a parameter includes problems originating from the angle of incidence of the probe which can vary slightly during processing due to machine vibrations, or from a dirty reference wafer. Because the important signal for the photospectrometer is the normally reflected light from the wafer, if the setup vibrates and results in a significant change in the angle of incidence, the intensity of the reflected light will vary, distorting the information returned to the spectrometer.

In summary, the TaME algorithm consists of solving two minimization problems. T_{RES} is found by solving equation (4.7).

$$\min \sum_{\lambda = 500nm}^{620nm} (R_{meas}(\lambda, T_{RES}) - (R_{theo}(\lambda, T_{RES}) + R_{offset}))^2 \quad (4.7)$$

and PAC (M) is found by solving equation (4.8):

$$\min \sum_{\lambda = 350nm}^{380nm} (R_{meas}(\lambda, T_{RES}, M) - (R_{theo}(\lambda, T_{RES}, M) + R_{offset}))^2 \quad (4.8)$$

4.4 Measurements Characterization

4.4.1 Outlier Filtering Methodology

Our installation does not work perfectly all the time. Sometimes, erroneous measurements are obtained as a result of an imprecise angle of incidence of the probe. A good criterion for choosing when to trust the TaME results and when not to is the error between the best fitting theoretical graph and the experimental graph. Assuming that the errors of the n samples of a wafer are normally distributed, there is a well-known relationship between the range of the n errors, and the standard deviation of that distribution. The control chart based on that relationship is the range chart, and we use it to filter out all data that lies outside the 3 standard deviations of the distribution of the n samples. The upper control limit (UCL) of the acceptable error is determined by [21].

$$UCL = \bar{x} + 3\frac{d_3}{d_2}\bar{x} \quad (4.9)$$

where $\bar{x}$ is the average error; d_3 the standard deviation of the distribution of the relative range; and d_2 the mean of the distribution of the relative range. Both d_3 and d_2 are tabulated functions of the sample size [21].

Once outliers are found, we recompute the UCL from the rest of the data, and repeat the filtering process until no data lie beyond the UCL (Figure 4.3). The main sources of bad TaME measurements are machine and metrology setup vibration and misalignment of the probe relative to the die area to be measured. As an indication of the robustness of the

method, we have shown below a sample of 50 residual errors between the TaME theoretical graphs and the experimental graphs. Each measurement was performed on a different wafer (Figure 4.3). Clearly, one solution to our current problem is to increase the sample size when measuring each wafer.

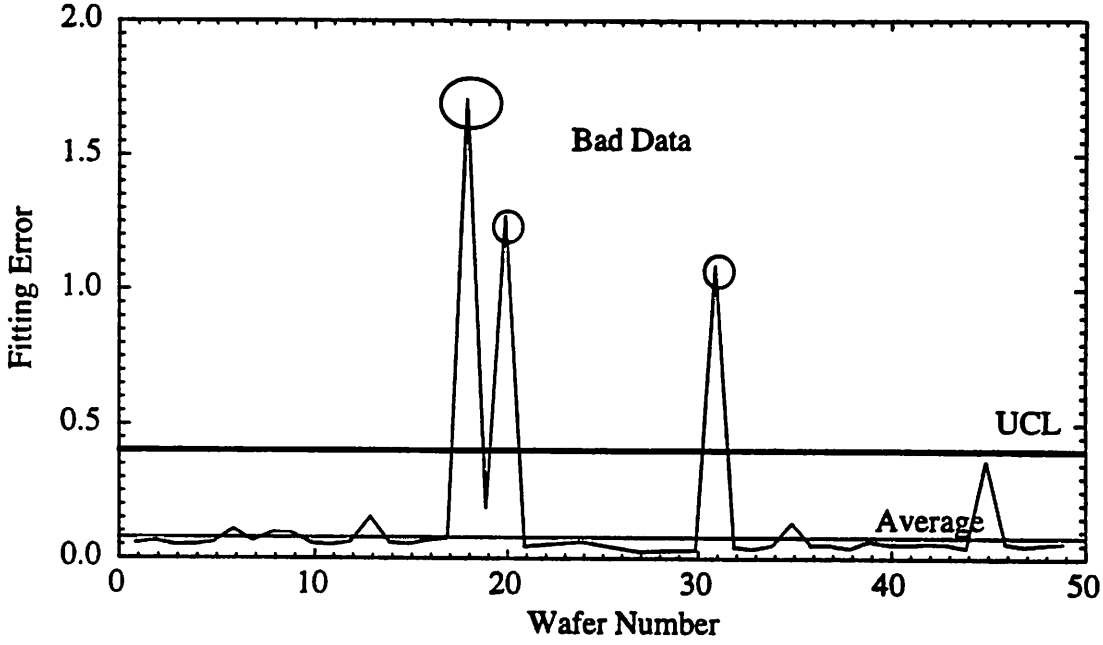


Figure 4.3 Methodology for Filtering out Bad TaME Measurements

4.4.2 Characterization of the Repeatability of TaME Measurements

Now, we characterize the repeatability of the TaME methodology. To quantify the measurement error, we spin-coat 7 wafers and measure them 7 times around the same spot, although not exactly the same spot, since this metrology will affect the photoresist. Therefore, some of the variations observed in the following graphs are also due to wafer non-uniformity. The results are shown below in Figure 4.4.

Next, we calculate the average range of each output characteristic, and determine at the 95% level of confidence the gauge error of the TaME method [21] [32].

Wafer #	1	2	3	4	5	6	7
Average Thickness (Ang.)	12727	12745	12708	12693	12722	12693	12686
Range (Ang.)	46	120	49	59	43	71	34

Average Range $\overline{R}$ = 60 Ang.

Gauge Error at the 95% level of confidence = 4 * $\overline{R}$/d2 = 89 Ang.

Wafer #	1	2	3	4	5	6	7
Average M	0.66	0.67	0.68	0.66	0.64	0.63	0.65
Range	.02	.05	.11	.06	.02	.04	.05

Average Range $\overline{R}$ = .05

Gauge Error at the 95% level of confidence = 4 * $\overline{R}$/d2 =.07

Table 4.1 Determination of the Gauge Error of the TaME Method

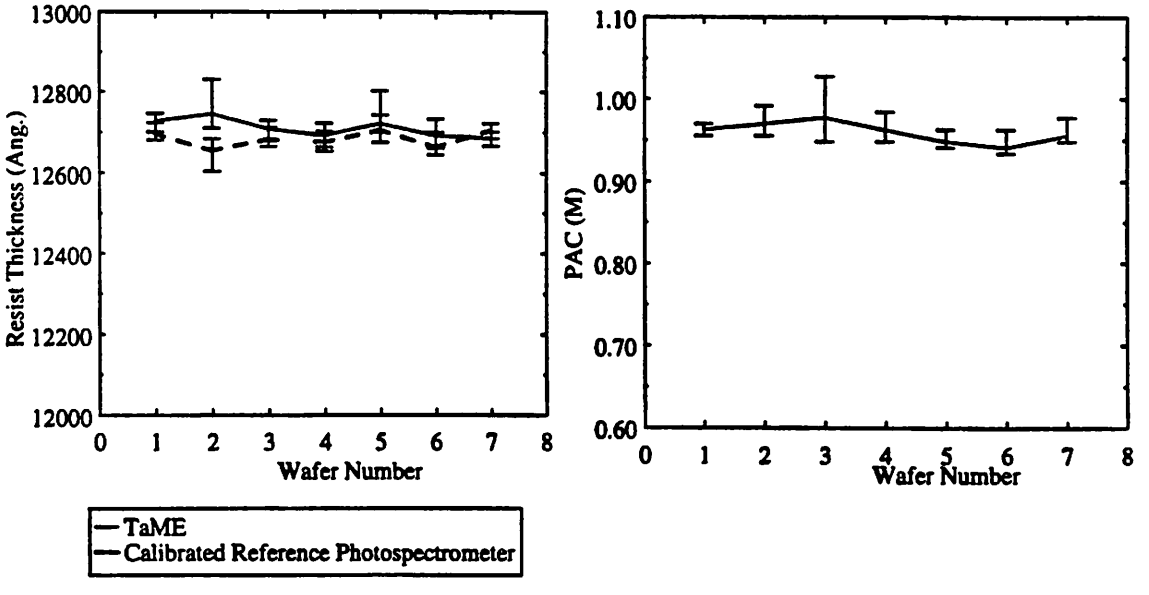


Figure 4.4 Thickness & PAC Measurement Repeatability

4.4.3 Characterization of Misalignment Effects

A major fault that causes measurement errors is misalignment of the probe with respect to patterned features on the wafer. To determine the sensitivity of the TaME method toward misalignment, we spin-coat 3 wafers using different recipes, in order to obtain resist films with different properties and expose them with a blank mask, so that each die is fully exposed. Finally, we measure the reflectance of each wafer 5 times in the following manner. Measurement #1: the wafer is aligned so that the probe footprint is completely on top of a die - 0% of the area is unexposed. Measurement #2: approximately 25% of the area probed is in the unexposed strip between the dies. Measurement #3: the probe footprint falls on only 50% of a die. Measurement #4: approximately 75% of the probe footprint falls on unexposed area. Finally, the last measurement is made completely on the unexposed strip between the dies. The results are shown below. Clearly, aligning the probe to the wafer is very important. This alignment problem is especially significant when measuring PAC after exposure. If the probe falls on the unexposed strip between the dies, the measurement needs to be redone. One solution is to pass the wafer through a flat finder before measuring. Then, we only need to align the probe once with the test die, and all subsequent wafers will also be properly aligned.

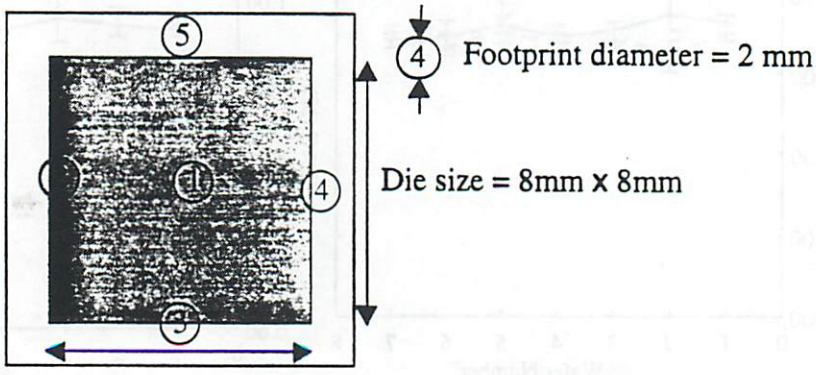


Figure 4.5 Descriptive Drawing of the Misalignment Experiment

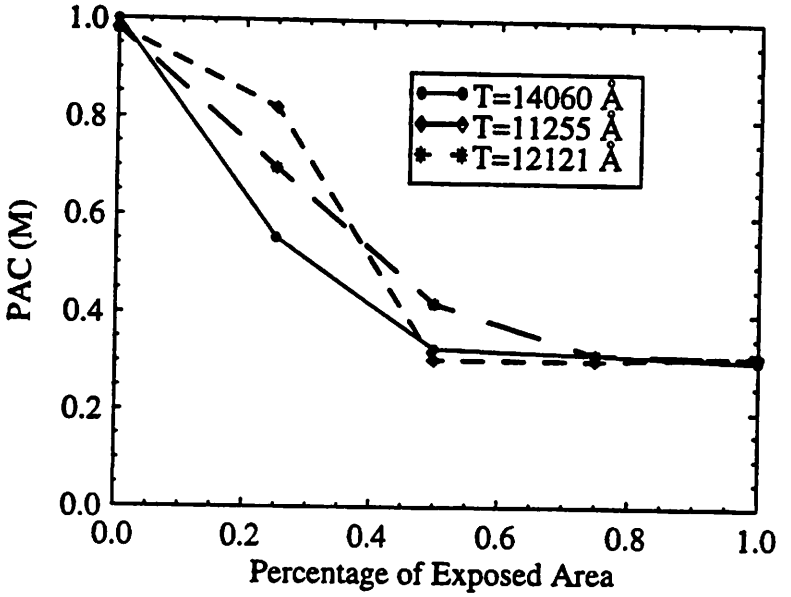


Figure 4.6 PAC vs. Exposed Area

4.4.4 Effect of Probing Time on Measurements

Next, we look at the effect of extensive probing time on the resist. Since we are probing the photoresist in its absorptive range, we are partially exposing it during measurement. This undesired effect is unfortunately necessary, since we want to measure the PAC inside the resist. To quantify the severity of this parasitic exposure problem, we conduct the following experiment. First, we measure repeatedly a wafer on the same spot for T_1 seconds. Next, we process the wafer in a regular fashion and time how long the wafer stays underneath the probe during an actual measurement (T_0). Then, we plot the degradation of PAC vs. T_1/T_0, which represents the number of measurements. Figures 4.7and 4.8 show the extent of the damage incurred during measurement. Figure 4.8 is a set of reflectance graphs, each taken with different lengths of probing time. Figure 4.7 shows the PAC values that correspond to these reflectance graphs vs. probing time. Clearly, an extensive probing time can be destructive, but as long as the total dose is minimal, we do not affect the chemical properties of the photoresist significantly.

Finally, to minimize this and the misalignment problem, we have also taken the following actions: we have reduced the aperture, i.e footprint of the probe and use a more focused beam. We have attached a mechanism to the source of the probe light that would allow us to modulate the amount of light emanating from the probe. Finally, we have developed a mechanism that allows us to rapidly align the probe beam to a die. This system can be further improved by adding an automated mechanical shutter that limits the exposure time during a measurement, and by using a faster computer to support the photospectrometer so that all the wavelengths can be scanned faster.

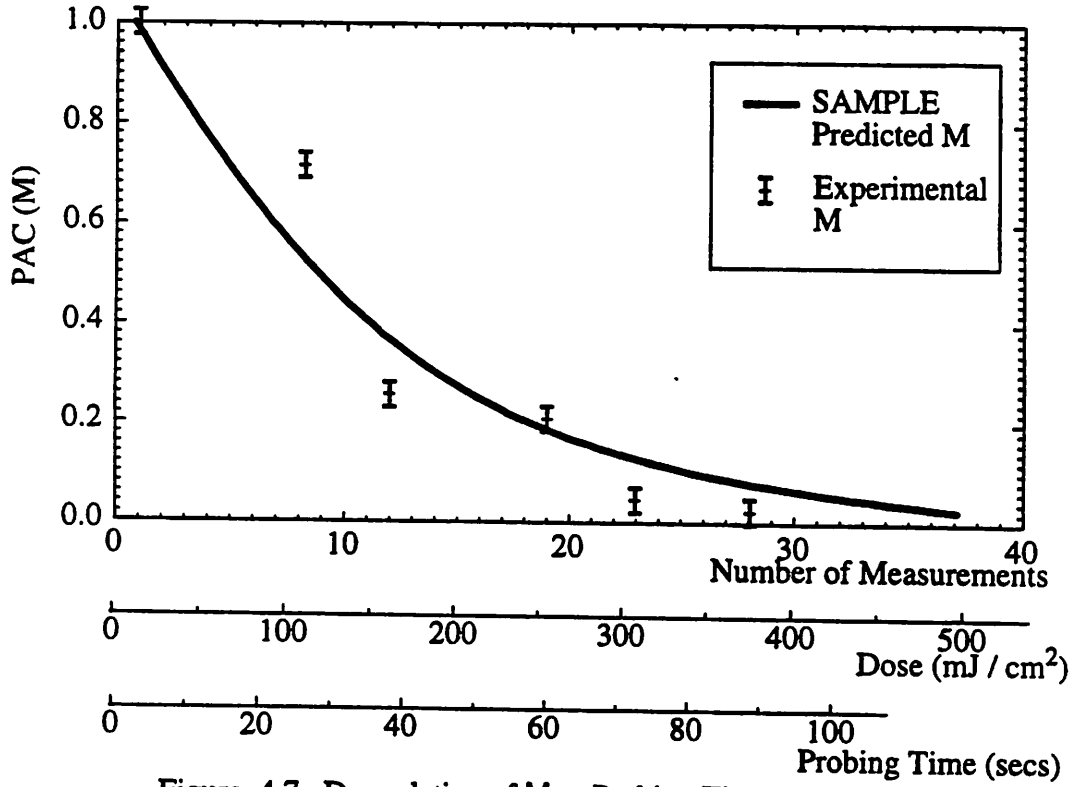


Figure 4.7 Degradation of M as Probing Time Increases

4.5 Conclusion

In conclusion, we have developed a novel metrology for measuring photoresist film thickness and photoactive compound concentration. This allows us to accurately characterize not only the physical, but also chemical properties of the resist film during photoli-

thography. The metrology and its caveats have been fully characterized and documented in this chapter. Next, we develop the photolithography control system.

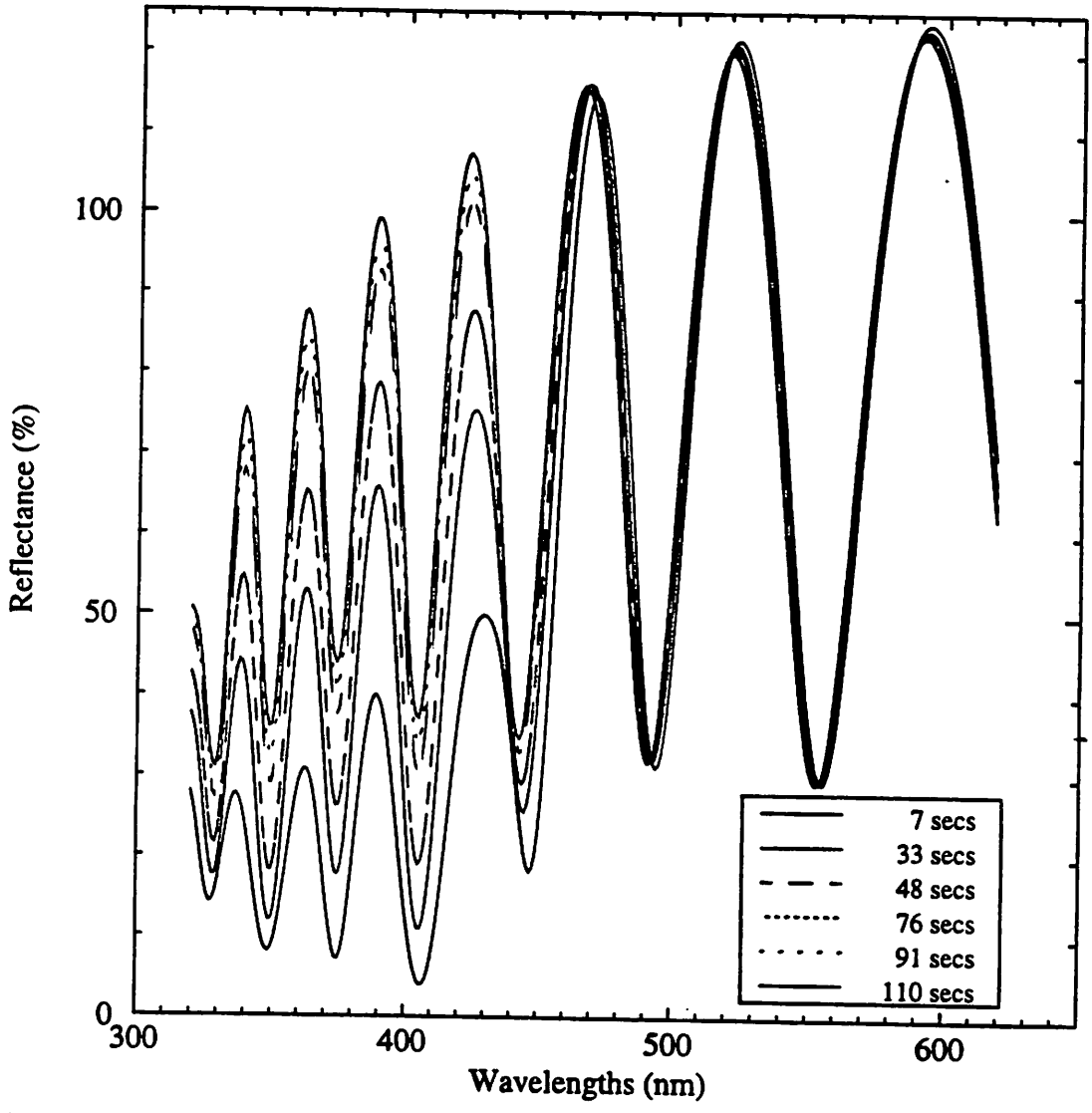


Figure 4.8 Reflectance Graphs of the Same Wafer with Varying Probing Time

[This page is intentionally blank]

Chapter 5 Supervisory Control System

5.1 Introduction

The goal of a supervisory control system is to improve the reliability and accuracy of a process sequence without significantly increasing the cost. We achieve that task by monitoring the process and ensuring that the outputs of all machines stay on or as close as possible to their respective targets. The control system consists of a feedback mechanism which ensures that the outputs of the current machine stay centered around their respective target, and a feed-forward mechanism which acts on the machine downstream to correct for variability introduced by the current machine. The feedback mechanism is described first.

5.2 Feedback Control System

5.2.1 Goal

The goal of the feedback controller is to ensure that the distribution of the process outputs stay centered on target. Triggered by *control alarms* which detect output drifts, the feedback controller first updates the equipment models of the machine, and then finds a new recipe to bring the machine's outputs back on target. If the machine has multiple outputs which cannot be brought back on target by a new recipe, due to correlation among outputs, a *compromise* recipe which brings all the outputs as close as possible back on target will be generated.

5.2.2 Background

Although heuristic algorithms for control have been reported in the semiconductor industry [24], we have chosen to base our approach on formal statistical methods. Statistically based algorithms offer several advantages over heuristic approaches, since they can

be adapted to a large number of processes and, once in place, are robust enough to be useful in an actual manufacturing environment.

Several attempts have already been made to formalize this procedure, including the MIT Run-by-Run Controller [25][82], Ultramax™ [26], and Texas Instrument's PCC Controller [79][80][81].

The MIT Run-by-Run Controller offers multivariate control and model adaptation of processes that exhibit linear relationships between inputs and outputs [25], and later versions integrate more general model adaptation and multivariate applications [82]. Ultramax is a commercial software for sequential process optimization and process control, that can also handle multiple inputs and outputs. Although the details of its operation are proprietary, Ultramax uses a variant of the evolutionary operation algorithm (EVOP) [21] to find the optimum operating point. Ultramax offers the significant advantage that no prior model of the process is required; however, it does require continuous changes on the process in order to derive such a model. Texas Instrument's PCC Controller is a supervisory controller that, like our controller, is designed to correct drifting processes. Applied to the etching process, it has proven itself experimentally to be very efficient at correcting process drifts due to equipment aging.

Although there are several similarities between the PCC Controller and ours, ours is designed to control multiple interrelated processes, such as the photolithography sequence, which are often currently lumped as one process and controlled as one process in industry. As to the MIT's Run-by-Run Controller, which, like ours, also uses well known and established statistical techniques to provide a robust process control, it currently does not handle non-linear process models.

5.2.3 Detection of Process Disturbances

Disturbances can be classified into two main types. The first one manifests itself through sudden significant changes in the process output. This indicates the presence of a problem that needs to be corrected by an operator. This type of disturbance triggers what we call *malfunction alarms*. The second type of disturbance manifests itself as a systematic process drift, which can be corrected by an appropriate recipe change. This type of disturbance triggers what we call *control alarms*. The schemes for detecting these two types of alarm are described next.

5.2.3.1 Malfunction Alarms

Malfunction alarms identify conditions which require operator attention. These are cases where the variation of a monitored parameter increases, or when we encounter sudden changes that are not consistent enough to be compensated by recipe adjustments. A malfunction alarm is also generated if the change cannot be compensated unless one (or more) of the controlling parameters moves beyond its acceptable range.

These conditions can be identified with the application of a special SPC scheme that can accommodate multiple parameters (as several process parameters are being monitored). This scheme must be able to ignore intentional changes in equipment settings such as those that might occur due to control algorithms. Such a SPC scheme has been developed using a combination of the Regression Chart [27] and Hotelling's T^2 statistic [28].

Under this scheme, malfunction alarms are generated in two stages: first, the equipment models are used to predict the new measurements. Then, the difference between the reading and the model prediction is analyzed. When the process is under statistical control, this difference is a random number with a known mean and variance. This variance is calculated using the prediction error of the model, as well as the observed variation of the

equipment. The method is described for univariate regression models in [27] and it has been generalized for multivariable response surface models in [29].

Let $\bar{y}$ be the p x 1 vector corresponding to p equipment outputs, each element being the average reading of n samples. Let $\hat{y}$ be the p x 1 vector predicted by the equipment models. If the process is under control, the *residual vector* $(\bar{y} - \hat{y})$ follows a multivariate normal distribution with mean **0**, and variance **Σ**. Once estimates of these parameters have been computed (equ. (5.2) - (5.6)), the multiple responses are merged together using the T^2 statistic [21].

$$T^2 = n(\bar{y} - \hat{y})^T S^{-1} (\bar{y} - \hat{y}) \tag{5.1}$$

where n is the sample size, and **S** the estimated covariance matrix of a process assumed to be in statistical control.

Usually, even processes in statistical control can change with time, which results in a continuously changing covariance matrix **S**. We have chosen not to monitor the change in **S** and use instead the estimated **S** from the analysis of the designed experiment, when the process is assumed to be in control.

It is calculated as follows [21]: let m be the number of wafers used in the designed experiments, we first calculate the average reading of each wafer:

$$\bar{y}_{jk} = \frac{1}{n} \sum_{i=1}^{n} y_{ijk}, \; i = 1,..., n; \; j = 1,...,p; \; k = 1,...,m. \tag{5.2}$$

Next, we calculate the covariance, variance, and mean response of the m wafers.

$$\bar{y}_j = \frac{1}{m} \sum_{i=1}^{m} \bar{y}_{jk}, \; j = 1,...,p; \; k = 1,...,m. \tag{5.3}$$

$$S^2_j = \frac{1}{m-1} \sum_{i=1}^{m} (\bar{y}_{jk} - \bar{y}_j)^2 \tag{5.4}$$

$$S_{jh} = \frac{1}{m-1}\sum_{i=1}^{m}(\bar{y}_{jk}-\bar{y}_j)(\bar{y}_{hk}-\bar{y}_h) \quad , j = 1,...,p;\ h = 1,...,p;\ j \neq h \tag{5.5}$$

Finally, we form the estimated covariance matrix **S**:

$$\mathbf{S} = \begin{bmatrix} S^2_1 & \cdots & S_{1p} \\ & S^2_j & S_{jp} \\ & & S^2_p \end{bmatrix} \tag{5.6}$$

Once the T^2 statistic is calculated, it is plotted on a single-sided control chart whose upper control limit (UCL) can be formally set at the desired probability of erroneously stopping a good process, by using the F distribution [21].

$$UCL = \frac{p \cdot (\mathcal{N}-1) \cdot F(p, \mathcal{N}-p)}{\mathcal{N}-p} \tag{5.7}$$

where $\mathcal{N}$ is the sample size during the production runs. Note that the sample size n used to calculate S is *different* from the sample size $\mathcal{N}$ used to determine the UCL.

When the UCL is exceeded, the automated control system stops and a human operator investigates the malfunction, the same way he would have investigated a traditional SPC out-of-control condition (Figure 5.1).

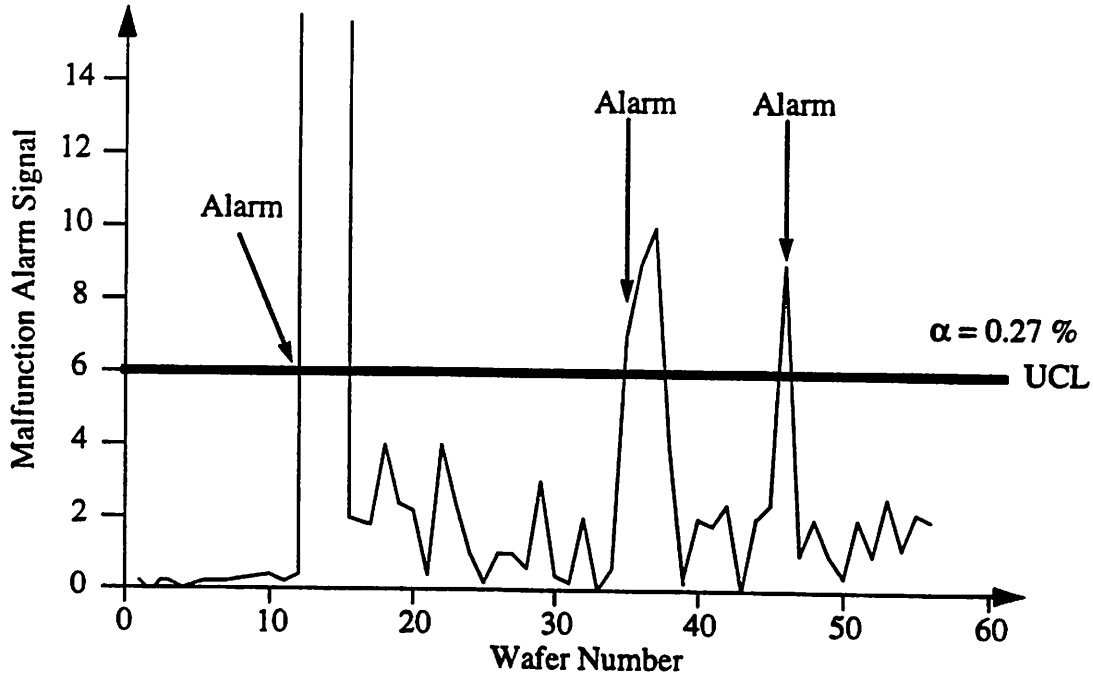


Figure 5.1 Malfunction Alarm Generation.

5.2.3.2 Alarms for Feedback Control

Control alarms identify process drifts and trigger the feedback control system. The drifts are detected by a multivariate cumulative sum (CUSUM) scheme that is very efficient at identifying small, consistent changes, while ignoring outliers that are not useful for feedback corrections. This type of disturbance can be compensated by appropriate recipe changes.

The alarm generation is based on Crosier's multivariate CUSUM scheme [30]. Several other schemes have been investigated, but none of them seems superior to Crosier's [31] [33]. Crosier's scheme forms a CUSUM vector directly from the residuals between the experimental data $\mathbf{y}_n$ and their respective model predictions $\hat{\mathbf{y}}$, after shrinking them by a factor of $\left(1-\frac{\kappa}{C_n}\right)$.

$$\mathbf{s}_n = \mathbf{0} \text{ if } C_n < \kappa \tag{5.8}$$

$$\mathbf{s}_n = (\mathbf{s}_{n-1} + \mathbf{y}_n - \hat{\mathbf{y}})\left(1-\frac{\kappa}{C_n}\right) \text{ if } C_n \geq \kappa \tag{5.9}$$

where C_n is the variance-normalized length of the residual CUSUM vector $(\mathbf{s}_{n-1} + \mathbf{y}_n - \hat{\mathbf{y}})$, i.e,

$$C_n = \sqrt{\left[(\mathbf{s}_{n-1} + \mathbf{y}_n - \hat{\mathbf{y}})^T \mathbf{S}^{-1} (\mathbf{s}_{n-1} + \mathbf{y}_n - \hat{\mathbf{y}})\right]} \tag{5.10}$$

The reason for shrinking the residual CUSUM vector by $\left(1-\frac{\kappa}{C_n}\right)$, and the significance of C_n and κ are fully explained in [30]. $\mathbf{S}$ is the same estimate of the covariance matrix used for generating malfunction alarms and is obtained from the designed experiments, when the process is in control.

Typically, we want a process to return to its original target. Sometimes, this is not always possible, because the multiple outputs are not completely independent of each other. A corollary of this is that measurements should not be compared against fixed targets, which are sometimes unattainable, since it would generate control alarms too often.

The comparison of the experimental data to the model predictions, on the other hand, would generate an alarm only if the updated models do not represent the experimental data well. Since this is exactly what is desired, the control alarm is then set off only when the model represents the data inadequately.

This scheme yields an alarm when the variance-normalized length of the residual CUSUM vector s_n is greater than a constant η:

$$Y_n = \sqrt{\left[s_n^T S^{-1} s_n\right]} > \eta \tag{5.11}$$

The sensitivity of the alarm depends on the number of output parameters p, and the constants κ and η, which can be adjusted for the desired probability α of stopping erroneously a good process. Equivalently, we can adjust the average run length (ARL) between false alarms when the process is in control, also called *on-target ARL*. The methodology for tuning the sensitivity of control alarms is described next.

5.2.4 Methodology for Tuning the Sensitivity of the Control Alarm

The sensitivity of a control alarm can be tuned by selecting either a desired on-target ARL, or a desired type I error α, since they are directly related by:

$$\text{on-target ARL} = \frac{1}{\alpha} \tag{5.12}$$

We choose to describe the tuning process, starting from a desired on-target ARL. Note that if the on-target ARL is chosen too high, the alarm will not be very sensitive to an out of control process, i.e the "off-target ARL" will be too high, which could result in many wafers processed out-of-control. On the other hand, if the on-target ARL is chosen too low, too many false alarms would be triggered and the operator loses trust in the alarm detection system. Therefore the value of the on-target ARL must be carefully chosen either from analysis of past historical data or by a process engineer.

Given a desired value of κ (which will be discussed next), the parameter η is determined directly by our choice of an on-target ARL. Although there exists no known analytical equation that links η to ARL, we can derive the relationship from Monte Carlo simulations [30]. As an example, we describe the process in the case that an on-target ARL of 200 is desired. First given a value for κ, we look for a lower bound on η. Starting with an initial guess for η, we simulate up to 1200 runs or until an alarm signal is given. We repeat that simulation 50 times and obtain the average ARL. If the ARL is in the range of 200, we decrement η, until the average ARL is below 200. The lower bound on η is now found. Next we repeat the 50 simulations and compute the average ARL, incrementing η by 0.1 at a time, until the average ARL is above 200. Then we fit a linear regression through the resulting average ARL and compute the parameter η corresponding to an ARL of 200. Finally, we repeat the whole process again for various κ. Plots of the resulting simulations are shown below.

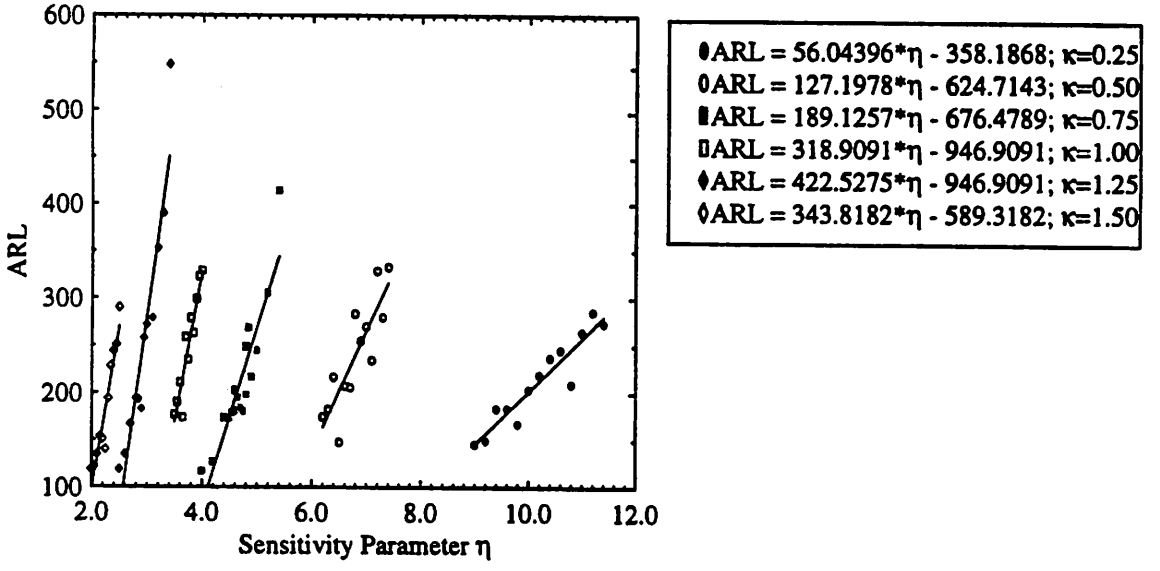


Figure 5.2 On-Target ARL vs. Parameters η and κ.

Note that the number of outputs p has been kept fixed during all simulations, because it is characteristic of a process. A different parameter p would result in different ARL vs. η relationships. In all simulations, we have used a value of 2 for p, because the wafer track has 2 outputs, and it is the only machine in our process sequence with multiple outputs.

We have also used independent data with a unit variance, so that we can use the identity matrix for S [30]. This is proper since the data is supposed to be random and normally distributed [34].

Parameter κ is related to the desired amount of shift in the mean vector to be detected [30]. To detect a K standard deviation shift in the mean vector ($K > 0$), we calculate the noncentrality parameter d.

$$d = \sqrt{\left[(K\sigma)^T S^{-1} (K\sigma)\right]} \tag{5.13}$$

where σ is the standard deviation vector of $\bar{y}$. Studies by Crosier have found that choosing $\kappa = d/2$ minimizes the off-target ARL of an out-of-control process with a noncentrality parameter d [30]. In other words, that value of κ makes the alarm optimally sensitive to the amount of shift represented by the noncentrality parameter d, and minimizes the number of wafers processed out-of-control. During actual processing however, the amount of process shift is not known in advance. To investigate which κ is best overall, we have plotted the off-target ARL vs. the noncentrality parameter d for various values of κ, given the same on-target ARL value of 200 (Figure 5.3). Notice that all the graphs are very similar,

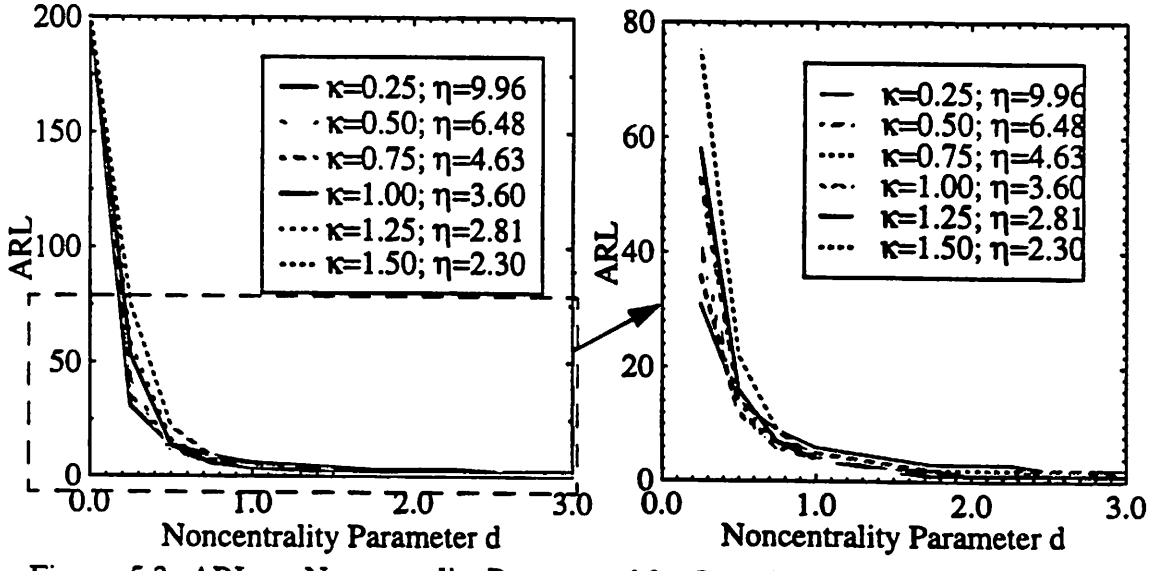


Figure 5.3 ARL vs. Noncentrality Parameter d for Combinations of (η, κ) such that on-target ARL = 200, with number of outputs p = 2.

when d is greater than 0.75, whereas there are great differences when d is small. Therefore when the process shift is large, the choice of a κ optimized for large shifts does not result in a significantly more sensitive alarm. Any value of κ results in approximately the same off-target ARL for large process shifts. On the other hand, when the process shift is small, the choice of a κ optimized for small process shifts results in a significantly more sensitive alarm. Therefore, the default value of κ in our controller is the smallest one, which is 0.25.

Finally, after specifying the desired on-target ARL and parameter κ, we select the parameter η from Figure 5.2, and the parameters of the control alarm are set.

5.3 Algorithm for Adaptively Updating Equipment Models

Although the original models offer a comprehensive representation of the process, their accuracy decreases over time, since equipment age, components are replaced, and environmental conditions change. Thus, it is important to develop a methodology for updating these models to the current status of the process. In our control methodology, adaptation occurs every time a control alarm is issued, since that signals when the model accuracy is not tolerable anymore. The model update algorithm we have developed is designed to work with random data, because our measurements will come from a production line, instead of from an off-line controlled experiment. The scheme is described next [35] [12].

5.3.1 Terminology and Data Conditioning

The model update algorithm is based on stepwise regression, which uses matrix computations. The kxn *input setting matrix* **X** contains the n input settings of the k process runs, which are then fed into a kxt *model term* matrix **T**, which stores the input settings as model terms. t corresponds to the number of terms inside the model, which can also be understood as the number of coefficients in the model, excluding the constant term.

As an example, let's assume that 2 wafers are processed by the wafer track. The first one is processed under a spin speed (SPS) of 4600 RPM, a baking time (BTI) of 60 seconds, and a baking temperature (BTE) of 90°C. The second wafer is processed under the following recipe: (SPS, BTI, BTE) = (4800, 65, 90). The resist thickness model coming out of the wafer track has the following terms: $1/(\sqrt{SPS})$, BTI, and BTE. All that information is stored as follows:

$$\mathbf{X} = \begin{bmatrix} 4600 & 60 & 90 \\ 4800 & 65 & 90 \end{bmatrix} \text{ and } \mathbf{T} = \begin{bmatrix} 1/(\sqrt{4600}) & 60 & 90 \\ 1/(\sqrt{4800}) & 65 & 90 \end{bmatrix} \tag{5.14}$$

Note that **X** and **T** do not necessarily have the same number of columns. If the resist thickness model also contained the term SPS, **T** would have had 4 columns: SPS, $1/(\sqrt{SPS})$, BTI, and BTE.

$$\mathbf{T} = \begin{bmatrix} 4600 & \frac{1}{\sqrt{4600}} & 60 & 90 \\ 4800 & \frac{1}{\sqrt{4800}} & 65 & 90 \end{bmatrix} \tag{5.15}$$

Next, the algorithm applies two transformations to **T** to prevent it from being ill-conditioned. First, it centers the resulting matrix, by substracting the average of each column, and then divides it by a *range matrix* **D**, so that the variances of each term are of comparable magnitudes. **D** is defined as a t x t diagonal matrix which contains the experimental range of each model term. This results in a matrix **Y**, which is composed of unitless numbers, with comparable magnitudes.

$$\mathbf{Y} = (\mathbf{T} - \mathbf{T}_{ave}) \cdot \mathbf{D}^{-1} \tag{5.16}$$

The second transformation that the algorithm applies on matrix **Y** is a Principal Component (PC) transformation, to ensure that each column of **Y** is orthogonal to each other. This is necessary in order to apply stepwise regression. Next, we briefly describe the PC step.

5.3.2 Principal Component Transformation

The principle components of a set of variables are linear combinations of the original variables, with the special property that they are all orthogonal to each other [28]. This is represented mathematically by the following two equations:

$$\mathbf{PC} = \mathbf{X} \cdot \mathbf{B} \tag{5.17}$$

$$b_i S_x b_j = 0, \text{ for } i \neq j \tag{5.18}$$

where **PC** is the set of principle components of variables X_i's; each column of **B**, b_i, contains the coefficients for one principle component; and S_x is the covariance matrix of the **X**'s.

To obtain the coefficients **B** for our matrix **Y**, we take the covariance matrix of the **Y**'s, S_Y, and find its eigenvectors:

$$\mathbf{B} \cdot \Lambda \cdot \mathbf{B}^T = S_Y \tag{5.19}$$

where Λ is a t x t diagonal matrix containing the t eigenvalues of S_Y; and **B** contains the columns of corresponding eigenvectors of S_Y. Next, we simply transform the matrix **Y** into its principle components $\mathbf{Y}_{PC}$ as follows:

$$\mathbf{Y}_{PC} = \mathbf{Y} \cdot \mathbf{B} \tag{5.20}$$

5.3.3 Description of the Model Update Algorithm

Now that all the terminologies and data conditioning have been explained, we present the model update algorithm [35][12]. The first step of the model update algorithm consists of entering all the machine settings into the input setting matrix **X**. Since the performance of the machine changes with time, we do not weight the outputs obtained from older settings as much as that obtained from newer settings. Therefore, we have applied a *forget-*

ting factor w_{kk} to our input settings, emphasizing the more recent ones over the older ones. (The variable k corresponds to the number of sets of input settings).

$$\mathbf{X'} = \mathbf{W} \cdot \mathbf{X} \tag{5.21}$$

where **W** is a diagonal matrix containing the forgetting factor w_{kk} of each set of input settings. In our implementation, the number of sets of input settings is also limited to a specific number, called *window size*, and is based on how often the machine performance drifts with time. Older wafers are effectively ignored by the model update algorithm.

Next, we transform the weighted input setting matrix **X'** into a model term matrix **T**, which we also transform into a unitless matrix **Y** through equ. (5.16) to avoid ill-conditioned matrix calculations. Then, we find the principal components of **Y**, $\mathbf{Y_{pc}}$.

Next, the difference between the measurements and the current model predictions, defined by a k x p *output discrepancy matrix* $\Delta\mathbf{z}$, is calculated. As before, p is the number of output variables, and k, the number of sets of input settings, i.e, the number of wafers in the window. The output discrepancy matrix is computed as follows for each output variable i, i = 1, ..., p:

$$\Delta\mathbf{z}_i = \mathbf{z}_{i,\,meas} - \mathbf{z}_{i,\,model} = \mathbf{z}_{i,\,meas} - (\mathbf{Y}_{pc}^{T} \cdot \gamma + c_0) \tag{5.22}$$

where $\gamma = \mathbf{B}^{T} \cdot \mathbf{D} \cdot \mathbf{c}$ represents the vector of term coefficients of the model, transformed into the principal component space; **c** is a tx1 vector containing all the model coefficients; c_0 is the constant term; and **D** is the range matrix.

Finally, stepwise regression is performed, considering each PC separately, in order to obtain a *vector of correction term coefficients* $\Delta\gamma$. The statistical significance based on the student-t distribution of each correction coefficient $\Delta\gamma_j$ (j = 1, ..., t) is calculated. If it is greater than a certain threshold, the correction coefficient is updated to $\Delta\gamma_j$; otherwise, it is set to zero. Next, $\mathbf{Y_{pc}}$ is multiplied by the updated set of new coefficients $\Delta\gamma$ and sub-

stracted from the output discrepancy vector $\Delta\mathbf{z}$. If the resulting constant term Δc_0 is significant, it is also updated. Finally, the modified correction coefficients $\Delta\gamma$ are transformed back to their original space, resulting in a set of correction coefficients $\Delta\mathbf{c}$, and added to the current model coefficients $\mathbf{c}$, to result in a newly updated set of coefficients $\mathbf{c}_{updated}$.

$$\mathbf{c}_{updated} = \mathbf{c} + \Delta\mathbf{c} = \mathbf{c} + \mathbf{D}^{-1} \cdot \mathbf{B} \cdot \Delta\gamma \tag{5.23}$$

$$c_{0_{updated}} = c_0 + \Delta c_0 \tag{5.24}$$

This concludes the equipment model update. The next step of the feedback controller is to find a new recipe that will bring the machine's outputs back on target.

5.4 Automated Recipe Generation

5.4.1 Algorithm

Once the equipment model has been updated to reflect the new state of the process, a new recipe is typically needed to bring the process responses back on target. That task is mathematically formulated as follows [35][36]:

Solve for $\mathbf{X}$, such that

$$f(\mathbf{X}) \cong \hat{\mathbf{z}} \tag{5.25}$$

where $f(\mathbf{X}) = \mathbf{T}^T \cdot \mathbf{c} + c_0$, from the previous section, and $\hat{\mathbf{z}}$ is the desired output from the machine.

Subject to the constraints

$$\mathbf{E}_m \leq \mathbf{X} \leq \mathbf{E}_M \tag{5.26}$$

where $\mathbf{E}_m$ correspond to the set of minimum input settings; and $\mathbf{E}_M$, the set of maximum input settings.

This is a typical optimization problem, which can be solved in many different ways. Several optimizers have been studied and implemented [21] [35] [36] [18], all with similar satisfactory results. Therefore, we have chosen the most simple one, the iterative Gauss-Seidel algorithm [35] [36].

At iteration i, the algorithm linearizes the function f(**X**) as follows:

$$\mathbf{f(X)} \cong \mathbf{f(X_i)} + \mathbf{A_i} \cdot (\mathbf{X_{i+1}} - \mathbf{X_i}) \tag{5.27}$$

where $\mathbf{A_i} = \frac{\partial \mathbf{f}}{\partial \mathbf{X}}(\mathbf{X_i})$.

Let p be the number of outputs variables, and k the number of input variables, we have three possible cases:

1. $\mathbf{k = p}$. The system is well determined and either has one solution or none at all. If a solution exists, it is given by:

$$\mathbf{X_{i+1}} = \mathbf{X_i} + \Delta\mathbf{X_i} = \mathbf{X_i} - \mathbf{A_i}^{-1}[\mathbf{f(X_i)} - \hat{\mathbf{z}}] \tag{5.28}$$

2. $\mathbf{k > p}$. The system is either over-determined or well determined. If $\mathbf{A_i A_i}^T$ is invertible, a solution to the system is given by [35]:

$$\mathbf{X_{i+1}} = \mathbf{X_i} + \Delta\mathbf{X_i} = \mathbf{X_i} - \mathbf{A_i}^T[\mathbf{A_i A_i}^T]^{-1}[\mathbf{f(X_i)} - \hat{\mathbf{z}}] \tag{5.29}$$

3. $\mathbf{k < p}$. The system is under-determined. If $\mathbf{A_i}^T\mathbf{A_i}$ is invertible, the solution to the system is given by [35]:

$$\mathbf{X_{i+1}} = \mathbf{X_i} + \Delta\mathbf{X_i} = \mathbf{X_i} - [\mathbf{A_i}^T\mathbf{A_i}]^{-1}\mathbf{A_i}^T[\mathbf{f(X_i)} - \hat{\mathbf{z}}] \tag{5.30}$$

Derivations of these results can be found in Bombay's thesis [35].

Typically however, the output variables must be weighted, because their effect on the final output are different. The input settings **X** are also weighted, because some settings are more easily changed than others. Let **O** be a p x p diagonal matrix, which contains the weights of $\hat{\mathbf{z}}$, and **I**, a k x k diagonal matrix, which contains the weights of **X**. Equation (5.27) is now transformed as follows [35]:

$$f(\mathbf{X}) \cong f(\mathbf{X}_i) + \mathbf{A'}_i \cdot (\mathbf{X}_{i+1} - \mathbf{X}_i) \tag{5.31}$$

where $\mathbf{A'}_i = \mathbf{O}^{-1} \cdot \mathbf{A}_i \cdot \mathbf{I}$.

As before, there are three cases.

1. $\mathbf{k} = \mathbf{p}$. The solution is still the same as before and is given by:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \Delta\mathbf{X}_i = \mathbf{X}_i - \mathbf{A}_i^{-1}[f(\mathbf{X}_i) - \hat{\mathbf{z}}] \tag{5.32}$$

2. $\mathbf{k} > \mathbf{p}$. If $\mathbf{A'}_i\mathbf{A'}_i^{\mathrm{T}}$ is invertible, the solution is given by:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \Delta\mathbf{X}_i = \mathbf{X}_i - \mathbf{I}\mathbf{A'}_i^{\mathrm{T}}[\mathbf{A'}_i\mathbf{A'}_i^{\mathrm{T}}]^{-1}\mathbf{O}^{-1}[f(\mathbf{X}_i) - \hat{\mathbf{z}}] \tag{5.33}$$

3. $\mathbf{k} < \mathbf{p}$. If $\mathbf{A'}_i^{\mathrm{T}}\mathbf{A'}_i$ is invertible, the solution is given by:

$$\mathbf{X}_{i+1} = \mathbf{X}_i + \Delta\mathbf{X}_i = \mathbf{X}_i - \mathbf{I}[\mathbf{A'}_i^{\mathrm{T}}\mathbf{A'}_i]^{-1}\mathbf{A'}_i^{\mathrm{T}}\mathbf{O}^{-1}[f(\mathbf{X}_i) - \hat{\mathbf{z}}] \tag{5.34}$$

To satisfy the constraints set by equ. (5.26), the algorithm freezes any input value violating a constraint to the value of the constraint itself, and reduces the dimension space of the search by one. The optimization algorithm then continues the search with the other inputs.

5.4.2 Methodology for Choosing the Weights for Output Variables

In the past [35], the weights of the output variables have been derived from the specification limits:

$$\Delta\hat{\mathbf{z}}' = \mathbf{O}^{-1}\Delta\hat{\mathbf{z}} = \mathbf{O}^{-1}[f(\mathbf{X}_i) - \hat{\mathbf{z}}] \tag{5.35}$$

where $\mathbf{O} = 2 \cdot \min(\mathrm{USL} - \hat{\mathbf{z}}, \hat{\mathbf{z}} - \mathrm{LSL})$.

We believe a better weighting scheme follows the sensitivity of the final process output (in this case, CD) on the intermediate output variables. For example, if the CD is as sensitive to a 3% change in PAC as to a 100 Å change in resist thickness, the following weights should be used: $\mathbf{O} = \begin{bmatrix} 0.03 \\ 100 \end{bmatrix}$.

More formally, the output weights **O** are chosen as follows:

$$\mathbf{O} = \begin{bmatrix} 1/\frac{\partial z_{final}}{\partial z_1} \\ \cdots \\ 1/\frac{\partial z_{final}}{\partial z_p} \end{bmatrix} \tag{5.36}$$

where p is the number of output variables, and z_{final} is CD for our process sequence.

5.4.3 Weights for Input Variables

Weights for input variables are needed, because some input settings have a wider range of operation than others, or can be changed more easily than others. Weights are then used to favor changing the input settings that would cause less side effects to the process. For example, changing the spin speed of a wafer track is preferable to changing the baking temperature. Currently, the weights for the input variables are given by the inverse of the input setting range.

5.5 Summary of the Feedback Control System

In this section, we have presented an implementation of feedback control on a semiconductor manufacturing step (Figure 5.4). The feedback control algorithm is based on the formal generation of malfunction and control alarms, an adaptive model updating strategy, and an automated recipe generation system. This algorithm has been implemented on various machines in the Microlab and experimental results, presented in chapter 7, show that

the capability of the process sequence is significantly improved when this control algorithm is applied on every machine of the sequence.

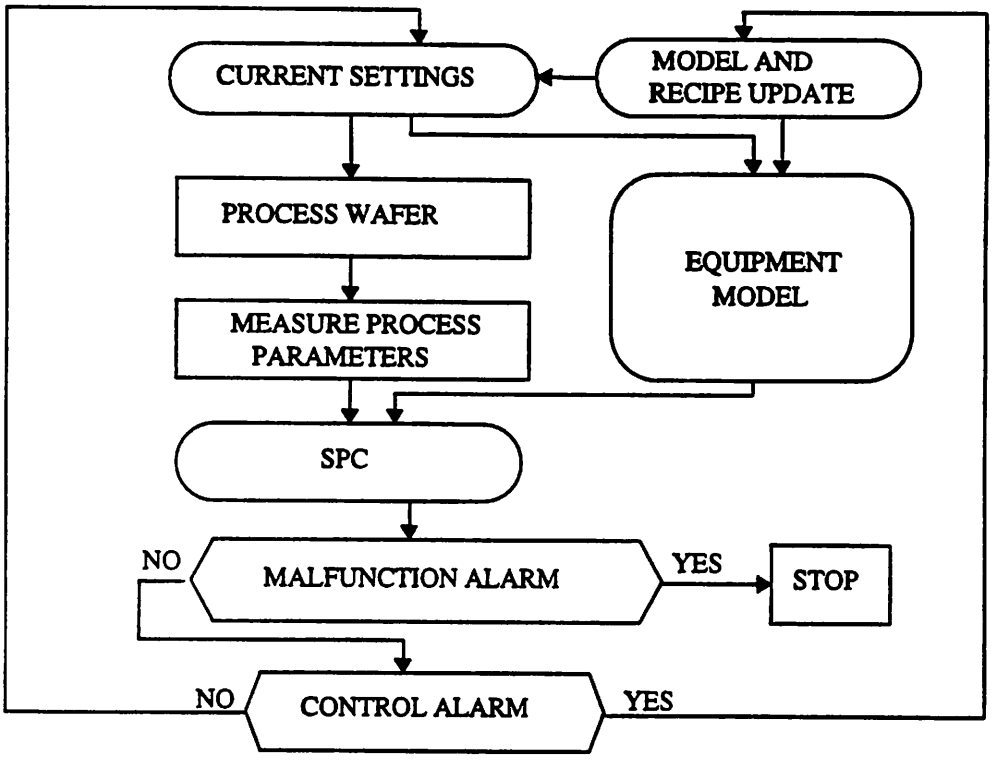


Figure 5.4 Schematic Representation of the Feedback Procedure.

5.6 Feed-Forward Control

5.6.1 Feed-Forward Control Paradigm

The primary task of the feed-forward control mechanism is to adjust downstream process step(s) in order to compensate for the variability of the current machine [11]. The feed-forward controller complements the feedback controller which centers the process on the target, by reducing the process variability. Before processing the wafer on the next equipment, the outputs of the current step are analyzed to see if they are likely to produce a wafer within specifications after the next step, assuming normal settings. If the analysis comes back positive, no feed-forward control is done on the wafer. However, if the analysis shows that the wafer is unlikely to meet specifications, a feed-forward alarm is triggered and activates the feed-forward controller which then finds a corrective recipe for the

next machine, using the same recipe generation described in §5.4 (Figure 5.5). In highly controllable process steps, the feed-forward controller can even compensate for inherent variability of previous steps, thereby increasing the overall process capability.

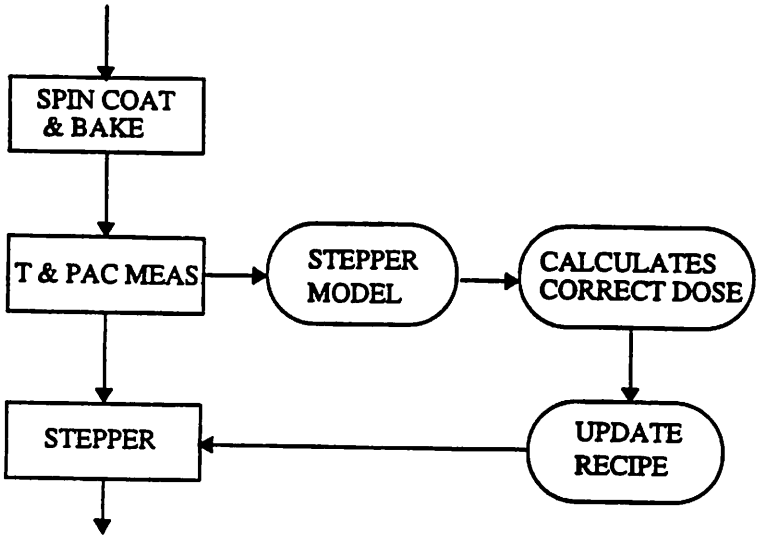


Figure 5.5 Example of the Feed-Forward Control Procedure Applied to a Stepper.

Currently however, feed-forward control mechanisms are not well accepted in the semiconductor industry because of the high stakes involved. A corrective action that worsens a process is not tolerated. That is why we activate the feed-forward control only when the problem is clearly confirmed. Like the feedback control mechanism, this mechanism is also activated by a formal statistical test.

5.6.2 Feed-Forward Alarm

The feed-forward alarm is a variant of the acceptance chart, whose properties are fully discussed in [21]. For a fraction of nonconforming wafers of at most δ, the true process mean μ is bounded by μ_L and μ_U, defined below:

$$\mu_L = \text{LSL} + Z_\delta \sigma, \text{ and } \mu_U = \text{USL} - Z_\delta \sigma \qquad (5.37)$$

where Z_δ is the upper $100(1 - \delta)$ percentage point of the normal distribution, and σ is the process variability when the process is in control. We find an estimate of σ by running the

standard process for a significant amount of time during which the process is believed to be in control, and then by calculating the standard deviation of the process output. Now, given a specified type I error of α, the upper and lower control limits of the feed-forward alarm are set at:

$$LCL = \mu_L - Z_\alpha \sigma_{pred} = LSL + Z_\delta \sigma - Z_\alpha \sigma_{pred} \tag{5.38}$$

$$UCL = \mu_U + Z_\alpha \sigma_{pred} = USL - Z_\delta \sigma + Z_\alpha \sigma_{pred} \tag{5.39}$$

where Z_α is the upper $100(1 - \alpha)$ percentage point of the standard normal distribution, and σ_{pred} is the prediction error of the equipment model of the machine. σ_{pred} is defined as the average error of the fitted values $\hat{y}_i$, and is calculated from σ_{model}, which is the standard error between the modeled data y_i and their fitted values $\hat{y}_i$ ($i = 1, .., N$, where N is the number of wafers used in building the equipment model of the machine) [22].

$$\sigma_{model} = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2} \tag{5.40}$$

$$\sigma_{pred} = \sqrt{\frac{t}{N}} \sigma_{model} \tag{5.41}$$

where t is the number of degrees of freedom used by the equipment model.

When the predicted output falls between the lower and upper control limits, no feed-forward action is taken. On the other hand, when a prediction falls outside the control limits, an alarm signals the feed-forward control mechanism to generate new recipe(s) for the next machine(s) in the sequence, in order to prevent the final process output from drifting outside the specification limits. Although the recipe generator always tries at first to correct the error at the next step, its success is not guaranteed and may require looking at sev-

eral subsequent steps. If the situation cannot be corrected at all, the feed-forward controller sends the wafer to be stripped and recoated.

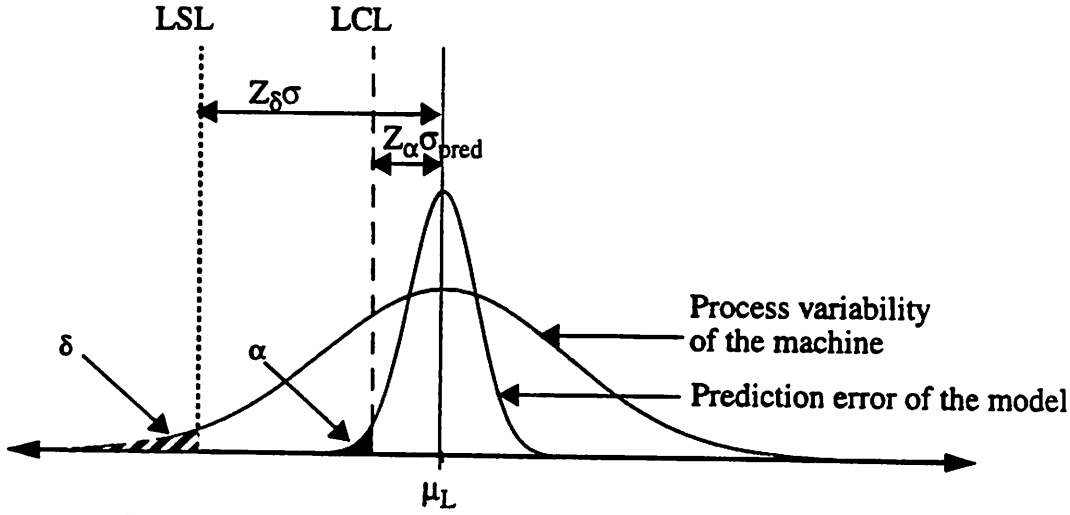


Figure 5.6 Derivation of LCL for Feed-Forward Alarm.

5.7 Summary

In conclusion, we have developed and implemented a robust supervisory control system that is capable of reducing the variability of a process sequence, and centering the process mean back on target. It achieves these tasks by applying statistical process control techniques on accurate equipment models. The control system consists of a feedback loop and a feed-forward loop. The feedback loop tracks the performance of each machine, using adaptive equipment models, and ensures that the distribution of the process step is centered around its target. Then, the feed-forward loop checks if standard settings on subsequent process steps would result in a correctly processed wafer. If the process outputs are predicted to be off-target, it will correct for the shortcomings of the present machine by generating customized recipes at subsequent process steps.

[This page is intentionally blank]

Chapter 6 The Photolithography Diagnostic System

6.1 Introduction

The supervisory controller described in the previous chapter is capable of correcting a drifting process, but does not diagnose the problem that causes the drift to occur. However, without diagnosing and then correcting the problem permanently, maintaining a stable, in-control process is very difficult. There lies our motivation for adding a diagnostic system to the controller. The goal of our photolithography diagnostic system is to assist a quality inspector in finding the faults that degrade the capability of the process, and is not intended to replace troubleshooting technicians. In other words, this diagnostic system does not find the problems that cause machines to break down, but rather those that cause machines' performances to change.

6.2 Anatomy of a Diagnostic System

A diagnostic system typically consists of three components: an inference engine, a knowledge base and a user interface [37] [38]. The function of the user interface is to provide an interface between the diagnostic system and the user, since all diagnostic systems are expected to work in conjunction with a human expert.

The knowledge base is the component that contains the expert information needed to diagnose the problem. This information can be represented in several ways, such as a set of rules, frames, semantic nets [38], belief networks [39], or equipment models [53][84][85]. Typically, a set of rules is used since it is the most easily implemented and understood, yet still efficient. A rule usually has an "if... then..." format, where the *if* part is called a symptom or an evidence, and the *then* part is called a fault. The expert information stored in the knowledge base can be of different levels of reasoning. It can vary from a shallow level, where the set of rules has been derived purely from the experience of human

experts, to a very deep level, where the set of rules has been derived from actual theories related to the domain [40] [41].

Finally, the third component, the inference engine, uses the information stored in the knowledge base to diagnose the problem, following one of several probability theories [37] - [47].

6.3 Description Of The Two Knowledge Base Approaches

6.3.1 Deep Level Knowledge Bases

A "deep level" diagnostic system uses models of the domain to infer the cause of the problem. It is very powerful when these models are physically based and well known, because deep level diagnostic systems can find the root cause of the problem by itself, by deriving it from the theory of the domain [37][41][44]. If instead the domain model is empirical, it is not as powerful because the model could be wrong, or the evidence could lie outside the experimental range of the model [53][84][85]. In either case however, a deep level diagnostic system is still very desirable because it can find the proper solutions in unanticipated situations, and it is not pigeon-holed into any fault.

The main disadvantage of deep systems lies in their difficult implementation. Most fields seldom have a complete theoretical foundation, and typically depend on numerous empirical results, which can also be incomplete.

6.3.2 Shallow Level Diagnostic Systems

On the other extreme, a shallow level knowledge base comes purely from the experience of human experts [37]. This allows a highly efficient diagnostic system to be built in a very short time. The accuracy of the diagnosis depends purely on the level of expertise of the human experts. MYCIN [42] and Internist [43] are good examples of well tuned shallow level diagnostic systems.

The disadvantage of a shallow level system, however, is that unless human experts include a cause in the knowledge base, it will never find that particular cause. Also, if human experts forget to link some symptoms to some causes, the system gets pigeonholed into a wrong fault. Finally, if human experts are wrong or simply just not knowledgeable enough, the knowledge base is equally faulty. In summary, a shallow level diagnostic system depends completely on human expertise and its shortcomings reflect ours.

Our diagnostic system uses a combination of both deep and shallow level knowledge bases. While sensors malfunctions and incorrect input settings are diagnosed from equipment models and measurements, environmental and maintenance related problems are diagnosed from operator observations, machine sensor alarms, and maintenance logs.

6.4 Probability Theories Used in the Inference Engine

Several formal theories have been developed for handling uncertainty. They invariably have a methodology for combining evidences and generating a diagnosis from them. These theories have been investigated in great detail [45]. Their conclusion can be summarized as follows.

Being the oldest theory, Bayesian theory is the most well-developed one and has become the benchmark against which all other theories are compared [45]. There is a well formalized procedure for implementing a diagnostic system based on Bayesian theory and it is based on the following equation:

$$p(F_i|E_1E_2...E_n) = \frac{p(E_1E_2...E_n|F_i) \times p(F_i)}{\sum_{k=1}^{m} p(E_1E_2...E_n|F_k) \times p(F_k)}, \quad i = 1, ..., m. \qquad (6.1)$$

where n is the number of evidences and m, the number of faults. The variable F_i represents the i-th fault, and E_j, the j-th evidence. This equation allows experts to turn the rules around and calculate conditional probabilities of faults. Bayesian theory has a few caveats

however: first, the values of a large number of conditional probabilities must be obtained ($N_{Cond.\ Prob} = n \cdot m$). Another one is that there is no explicit representation of ignorance.

One advantage of Demspter-Shafer theory [46] is that it is possible to explicitly represent ignorance, whereas in probability theory, ignorance is an implicit part of the probability assignments. In Dempster-Shafer theory, ignorance is defined as the difference between the *plausibility* and the *belief* of an event, instead of one minus the negation of an event. Unfortunately, Demspter-Shafer theory requires the fault space to be even larger than Bayesian theory. Dempster-Shafer theory lists all the faults into a concept called frame of discernment Θ, defined as an exhaustive set of mutually exclusive events, which resembles the fault space in Bayesian theory. However, given n faults, it can consist of up to 2^n elements, representing all possible subsets of Θ. This leads to a similar problem encountered in Bayesian theory, except that it is worse because human experts are required to estimate a larger number of belief values. The other main caveat of Dempster-Shafer theory is that it offers no procedure for implementating a diagnostic system [37]. Therefore, we have rejected this theory for our diagnostic system.

Finally, Possibility theory [47], an extension of fuzzy set theory, handles categorical and qualitative data well, because it represents their fuzziness explicitly. However, there is some ambiguity in the interpretation and definition of fuzzy quantifiers that need to be studied further and resolved, and like Dempster-Shafer theory, Possibility theory also does not have rigorous procedures for developing diagnostic systems [37].

In our diagnostic system, since our data are purely quantitative and the structure of the domain relatively simple, we have been able to simply use basic probability theory, which is the foundation of Bayesian theory. The mathematical formulations are well developed and a rigorous procedure exists for developing diagnostic systems from it.

6.5 Definitions of Terms Used in Our Diagnostic System

We define now the terms used in our diagnostic system. When a process goes out-of-control, the diagnostic system runs through a list of possible faults, called **fault space**, and calculates the probability of each fault. In our software implementation, the fault space of a machine is represented as a vector of faults. For example, the fault space of the stepper is: $\mathcal{F}^{stepper} = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8, F_9, F_{10}, F_{11}, F_{12}\}$:

Table 6.1 Fault Space of the Stepper

Fault Index	Fault Names
F_1	Wrong Input Thickness
F_2	Wrong Input PAC
F_3	Wrong Dose
F_4	Bad Lamp
F_5	PAC Meas. Error
F_6	Bad Lamp Strike
F_7	Damaged Filter Optics
F_8	Bad Shutter Timing Circuit
F_9	Bad Light Integrating Circuit
F_{10}	Environmental Temperature
F_{11}	Miscellaneous Fault
F_{12}	No Fault

The **evidence space** $\mathcal{E}$ used to deduce the fault consists of a list of **pieces of evidence**. Currently, the knowledge base of our diagnostic system contains five pieces of evidence: $\mathcal{E} = \{E_1, E_2, E_3, E_4, E_5\}$. The first four represent shallow level information, while the last one contains deep level information, derived from equipment models.

Table 6.2 Evidence Space Description

Evidence Index	Pieces of Evidence
E_1	Operator Observation

Table 6.2 Evidence Space Description

Evidence Index	Pieces of Evidence
E_2	Age of Critical Machine Component
E_3	Machine Output
E_4	Type of Alarm
E_5	Fit of the Output(s) predicted by a Hypothetical Input change, to the measurement

The information used to diagnose any machine is composed of these five pieces of evidence.

Each piece of evidence is divided into a set of discrete, independent **variables**, which are specific to each equipment. For example, in the case of the wafer track, the piece of evidence E_3, "Machine Output", has two variables: $(E_{3,1},E_{3,2})$ = ("Thickness Measurement", "PAC Measurement").

The **value space** of each variable is discrete, i.e, the value of each variable is stored in a specific **category** with a specific probability. The sum of probabilities of all categories of a variable equals one, i.e all the categories of a variable are mutually exclusive and exhaustive. For example, the variable $E_{3,1}$, "Thickness Measurement", is divided into two categories, $\{E^{+}_{3,1}, E^{-}_{3,1}\}$. $E^{+}_{3,1}$ corresponds to a thickness value greater than the value predicted by the model, while $E^{-}_{3,1}$ corresponds to a thickness value less than the value predicted by the model.

Therefore, given a process described by m pieces of evidence E_i (i = 1, .., m), each E_i being divided into n_i discrete variables $E_{i,k}$ (k = 1, .., n_i), and each variable $E_{i,k}$ being divided into $q_{i,k}$ categories $E^{j}_{i,k}$ (j=1, .., $q_{i,k}$), the evidence space is then divided into $\mathcal{N}$ mutually exclusive and exhaustive **combinations**.

$$\text{Number of combinations} = \mathcal{N} = \prod_{i=1}^{m} \prod_{k=1}^{n_i} q_{i,k} \qquad (6.2)$$

We define here by **combination**, a vector of categories, each belonging to a different variable. For example, the data structure of the evidence space of the exposure step is shown below:

Piece of Evidence	Variables	Values		
Operator Observation (E_1)	Temp. Sensor Out of Range ($E_{1,1}$)	True ($E_{1,1}^+$)	False ($E_{1,1}^-$)	
Age of Machine Component (E_2)	Lamp Age ($E_{2,1}$)	New ($E_{2,1}^-$)	Old ($E_{2,1}^+$)	
	Filter Age ($E_{2,2}$)	New ($E_{2,2}^-$)	Old ($E_{2,2}^+$)	
Machine Output (E_3)	ΔPAC ($E_{3,1}$)	Above Target ($E_{3,1}^+$)	Below Target ($E_{3,1}^-$)	
Type of Alarm (E_4)	N/A†	Malfunction Alarm (E_4^M)	Control Alarm (E_4^C)	False Alarm (E_4^0)
Fit of Output Predicted by a Hypothetical Input Change to the Measurement† (E_5)	Wrong Input Thick ($E_{5,1}$)	Perfect Fit ($E_{5,1}^+$)	No Fit ($E_{5,1}^-$)	
	Wrong Input PAC ($E_{5,2}$)	Perfect Fit ($E_{5,2}^+$)	No Fit ($E_{5,2}^-$)	
	Wrong Input Dose ($E_{5,3}$)	Perfect Fit ($E_{5,3}^+$)	No Fit ($E_{5,3}^-$)	

Table 6.3 Data Structure of the Evidence Space of the Stepper

† Note that E_4 has no variable, and that E_5 has as many variables as there are inputs to the machine.The difference between pieces of evidence having no variable and one variable is purely philosophical: a piece of evidence with only one variable can theoretically have more variables, if more variables are later deemed necessary for diagnostic purposes, while a piece of evidence having no variable will have none forever, because of the way it was defined. But mathematically, formulae treat pieces of evidence that have no variable, the same way as if they have one variable.

6.6 Calculation of Fault Probabilities

Before describing the calculations of the fault probabilities, we first state the assumptions underlying the theory of our diagnostic system. All combinations of evidence are assumed to be mutually exclusive and collectively exhaustive, i.e,

$$\sum_{j=1}^{\mathcal{N}} p(C_j) = 1.0 \qquad (6.3)$$

The *observed* evidence then gets straddled over several combinations, with a different probability for each combination. We have assumed that any useful evidence is contained in a variable of one of the five pieces of evidence. Although such a claim is probably false for our current evidence space of each machine (Figure 6.9 - Figure 6.11), the data structure of the evidence space facilitates its update, as more useful evidence emerge.

The probability of each fault is based on the relative frequency of faults for a given combination of evidence. The relative frequency of a fault F_i for a given combination of evidence C_j ($j = 1, .., \mathcal{N}$) is called a conditional probability and is denoted by $p(F_i|C_j)$. Typically, in Bayesian diagnostic systems, the conditional probabilities of faults, $p(F_i|C_j)$, are determined from conditional probabilities of evidence, $p(C_j|F_i)$, and prior probabilities of faults, $p(F_i)$:

$$p(F_i|C_j) = \frac{p(C_j|F_i) \times p(F_i)}{p(C_j)} \tag{6.4}$$

We have assumed, however, that the estimates of the conditional probabilities of faults, $p(F_i|C_j)$, are given directly by machine experts (and then subsequently automatically updated by the diagnostic system (equations (6.8) and (6.9))), avoiding the need to determine prior probabilities of faults. The probability of a fault F_i ($i = 1, .., N_F$) is then calculated as follows:

$$p(F_i) = \sum_{j=1}^{\mathcal{N}} p(F_i|C_j) \times p(C_j) \tag{6.5}$$

where N_F is the number of faults, and $\mathcal{N}$ is defined in equation (6.2). While the conditional probabilities of faults are obtained from the database of the diagnostic system, the probabilities of combinations of evidence are calculated from the observed evidence (see §6.7).

The accuracy of the conditional probabilities of faults for a combination of evidence C_j improves with the number of occurrences, N_j. If a combination of evidence C_j has been diagnosed and linked to a fault F_i, $p(F_i|C_j)$ is updated as follows:

$$p(F_i|C_j) = \frac{N_j \times p(F_i|C_j)_{old} + 1}{N_j + 1} \tag{6.6}$$

while the other faults have their conditional probabilities updated as follows:

$$p(F_k|C_j) = \frac{N_j \times p(F_k|C_j)_{old}}{N_j + 1} \tag{6.7}$$

However, most of the time, the observed evidence cannot be linked to just one single particular combination of evidence. Rather, it straddles over several combinations of evidence, each with a probability of $p(C_j)$. Therefore, if fault F_i was the real fault, $p(F_i|C_j)$ is updated as follows:

$$p(F_i|C_j) = \frac{N_j \times p(F_i|C_j)_{old} + p(C_j)}{N_j + p(C_j)} \text{ , for all } C_j. \tag{6.8}$$

and the conditional probability of the other faults are updated as follows:

$$p(F_k|C_j) = \frac{N_j \times p(F_k|C_j)_{old}}{N_j + p(C_j)} \text{, for all } C_j. \tag{6.9}$$

The number of occurrences of C_j, N_j, is then updated to:

$$N_{j,\,new} = N_{j,\,old} + p(C_j) \tag{6.10}$$

Clearly, N_j will not be an integer anymore, but rather a real number.

The combinations of evidence which occur more frequently will have their conditional probabilities determined with more precision and accuracy. The relationship between these three parameters is formally derived in section §6.9. In the initial implementation of the diagnostic system though, we have no record of diagnosis cases beyond the experience of machine operators. Their opinions, albeit subjective, provide the initial conditional probabilities for each fault. We will describe how we extract initial estimates of conditional probabilities from their experience, and combine their different opinions into one single set of conditional probabilities for the diagnostic system, in section §6.8. Before we discuss how to obtain values for conditional probabilities, we first show how to calculate the probability of a combination of evidence, $p(C_j)$.

6.7 Calculation of the Probability of a Combination of Evidence

The knowledge base used for diagnostic operations is best represented by influence diagrams. An influence diagram is a scheme developed by SRI researchers to model complex decision problems involving uncertainty [57]. An inference diagram is an acyclic directed graph with nodes representing variables and arcs representing the relationships between variables. More specifically, an arc going from node A to node B represents the conditional influence of A on B. These influences are then calculated using Bayesian probability theory. Note that what is important in an influence diagram is the absence of an arc, rather than the presence of one, since the latter describes only a *possible* dependency of B on A, where as the lack of arc between A and B makes a stronger statement by marking the *independence* between A and B.

The diagnostic operations described in this thesis can be represented by the following influence diagram:

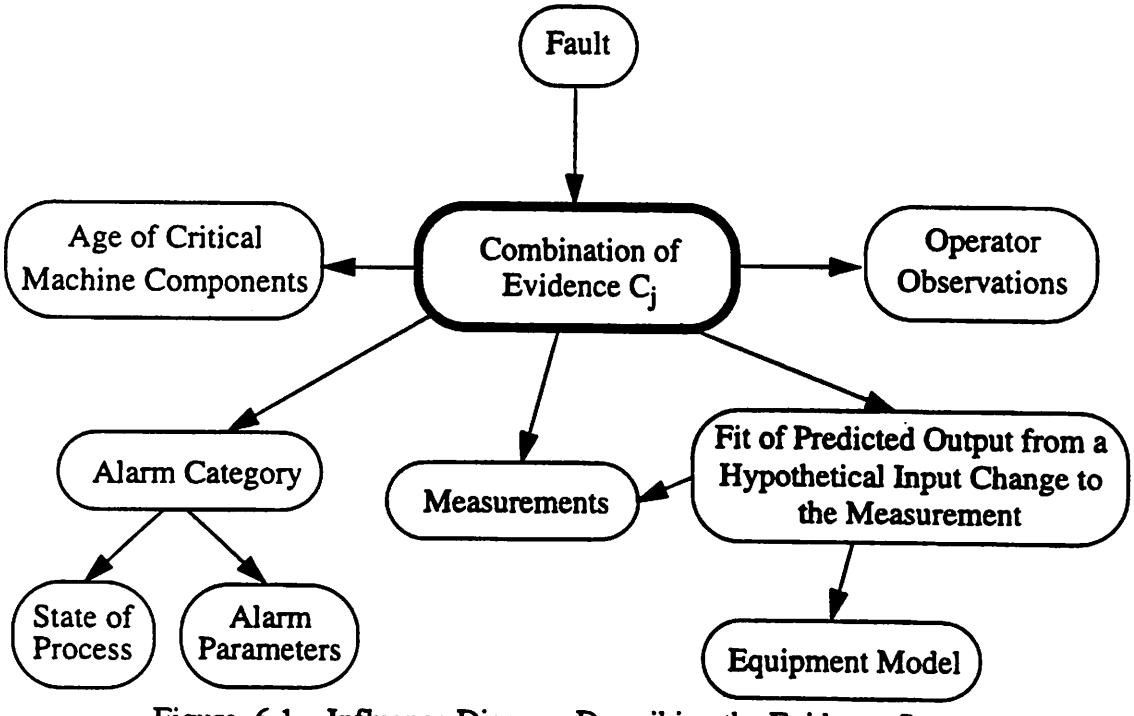


Figure 6.1 Influence Diagram Describing the Evidence Space

The centerpiece of Figure 6.1 is the probability of a combination of evidence C_j, $p(C_j)$. It corresponds to the probability of matching the observed evidence to the combination of evidence C_j. Since all the pieces of evidence are independent of each other, and their variables are also independent of each other, $p(C_j)$ is calculated by taking the product of the probabilities of the variables of the pieces of evidence. We will show next how to calculate the probability of a variable of any piece of evidence.

6.7.1 Probability of Categories of "Operator Observation" (E_1)

The piece of evidence *"Operator Observations"* consists of observations about physical attributes of the wafer, such as "streaks on wafer" or "circular patterns on wafer", and alarms from equipment sensors that are not connected to the computer running the control and diagnostic systems. These alarms alert the operator directly, who in turn logs them in the diagnostic system. Variables of this piece of evidence are divided into two categories: $(E^+_{1,i}, E^-_{1,i})$ = ("True", "False"). So, for example, circular patterns of photoresist on a wafer after the spin-coat and bake step is an observation that either exists or does not. If the observation actually occurred, then

$$p(E^+_{1,i}) = 1, \text{ and } p(E^-_{1,i}) = 0 \tag{6.11}$$

If the observation has not occurred, the probabilities are reversed.

6.7.2 Probability of Categories of "Machine Component Age" (E_2)

Each variable of the piece of evidence *"Age of Critical Process/Machine Components"* corresponds to the age of a distinct process/machine component. Each variable of E_2 is divided into two categories: $(E^+_{2,i}, E^-_{2,i})$ = ("Old", "New").

Given the age and the life distribution of a particular component *i*, we calculate the probability $p(E^-_{2,i})$ that this component would be classified as "new", by relating $p(E^-_{2,i})$ to the probability of failure of the component. Various life distribution functions have been

analyzed in the literature [69][70]. We have chosen the Weibull distribution instead of any other distribution to represent the life function of machine components, because it can be adjusted to fit most life distributions. Its mathematical form is:

$$R(t) = \exp[-(t/n)^B] \tag{6.12}$$

R(t) is the probability that the component is not likely to fail, and corresponds to the probability that the component is "new". B is the shape parameter and n is the scale parameter or characteristic life of the component, defined as the life at which 63.2% of the population has failed [70]. When B is less than 1, the failure rate decreases with time, and vice-versa, when B is greater than 1. In our case, machine components invariably degrade with time, and therefore B will always be greater than 1.

The Weibull *Mean Time to Failure* (MTTF) is given by [70]:

$$MTTF = n\Gamma(1 + 1/B) \tag{6.13}$$

where $\Gamma(\bullet)$ is the complete gamma function.

There are several ways of estimating the scale and shape parameters n and B [77]. The easiest one estimates them by plotting the cumulative number of failures vs. the time of failure, and then extracting the 16.7% ($\hat{y}_1$), 97.4% ($\hat{y}_2$), and 63.2% ($\hat{n}$) percentiles. The shape factor is estimated from $\hat{y}_1$ and $\hat{y}_2$:

$$\hat{B} = \frac{2.989}{\log(\hat{y}_2/\hat{y}_1)} \tag{6.14}$$

and the scale factor is estimated by $\hat{n}$.

If in the initial stage of the system implementation, there is not enough data to plot the distribution of failure of the machine component, an educated guess must be made for $\hat{n}$, while $\hat{B}$ can take on a value of 3.5, which makes the Weibull distribution approximate a normal distribution. Other ways of estimating n and B are suggested in [77].

In summary,

$$p(E^{-}_{2,i}) = R_i(t_i) = \exp\left[-(t_i/n_i)^{B_i}\right] \quad \text{and} \quad p(E^{+}_{2,i}) = 1 - R_i(t_i) \tag{6.15}$$

6.7.3 Probability of Categories of "Machine Outputs" (E_3)

The piece of evidence *"Machine Outputs"* is divided into p variables, which correspond to the p machine outputs. Each variable is in turn divided into 2 categories, $\{ E^{+}_{3,i}, E^{-}_{3,i} \}$ (i = 1 , ..., p). Let

$$\Delta E_{3,i} = (E_{3,i})_{\text{Measured}} - (E_{3,i})_{\text{Model Prediction}} \tag{6.16}$$

the probabilities of the two categories are defined as follows:

$$p(E^{+}_{3,i}) = p(\Delta E_{3,i} > 0) \quad \text{and} \quad p(E^{-}_{3,i}) = p(\Delta E_{3,i} < 0) \tag{6.17}$$

The measurement value, $E_{3,i}$, depends however on whether the operator measured the wafer correctly or not. Let O represent "Operator Aptitude", which is divided into 2 categories $\{O_g, O_b\}$ = {"Good Operator", "Bad Operator"}. The probability of a category of E_3 is calculated as follows (Figure 6.2):

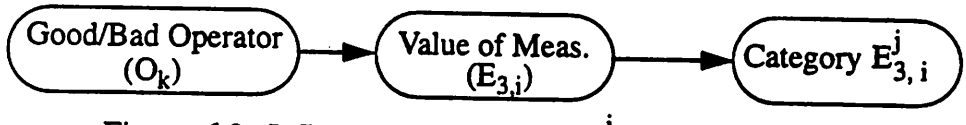


Figure 6.2 Influence Diagram of $p(E^{j}_{3,i})$, (j = "+" or "-")

$$p(E^{j}_{3,i}) = p(E^{j}_{3,i} \mid O_g) \times p(O_g) + p(E^{j}_{3,i} \mid O_b) \times p(O_b) \quad , j = \{\text{"+"}, \text{"-"}\} \tag{6.18}$$

$p(E^{j}_{3,i} \mid O_g)$ is calculated as follows:

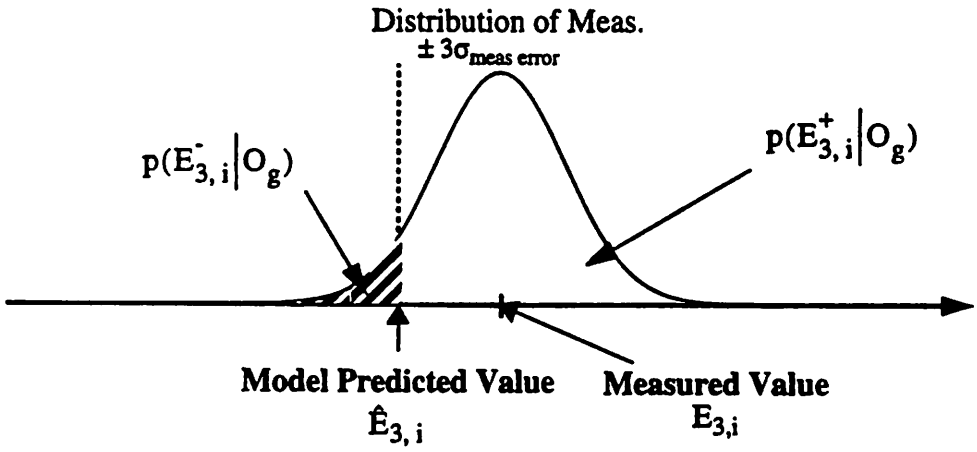


Figure 6.3 Determination of $p(E^{j}_{3,i}|O_g)$

Let $E_{3,i}$ be the measurement value; $\hat{E}_{3,i}$ the model predicted value; σ_1, the standard measurement error (obtained from multiple experiments at the standard setting when the process is in-control); $\Phi(\bullet)$, the cumulative gaussian distribution function; and erf(•), the error function:

$$p(E^{-}_{3,i}|O_g) = \Phi\left(\frac{\hat{E}_{3,i} - E_{3,i}}{\sigma_1}\right) = \left(\text{erf}\left(\frac{\hat{E}_{3,i} - E_{3,i}}{\sigma_1\sqrt{2}}\right) + 1\right)/2 \tag{6.19}$$

$$p(E^{+}_{3,i}|O_g) = 1 - p(E^{-}_{3,i}|O_g) \tag{6.20}$$

If the measurement $E_{3,i}$ is incorrect, it has equal probability of being in either categories:

$$p(E^{-}_{3,i}|O_b) = p(E^{+}_{3,i}|O_b) = 1/2 \tag{6.21}$$

Next, we calculate $p(O_g)$ and $p(O_b)$, where $p(O_g)$ and $p(O_b)$ represent the probability that the operator performed the measurement correctly, and incorrectly, respectively:

$$p(O_g) = 1 - \frac{\text{Number of Meas. Errors}}{\text{Total Number of Faults}} \tag{6.22}$$

$$p(O_b) = 1 - p(O_g) \tag{6.23}$$

where the total number of faults is the combined number of malfunction alarms and control alarms, since the diagnostic system is activated by these alarms.

6.7.4 Probability of Categories of "Type of Alarm" (E_4)

The piece of evidence *"Type of Alarm"* has no variable, and is directly divided into three categories: (E_4^M, E_4^C, E_4^0) = ("True Malfunction Alarm", "True Control Alarm", "False Alarm"). All process states, which include the three categories of E_4, are shown below in the two VEM diagrams.

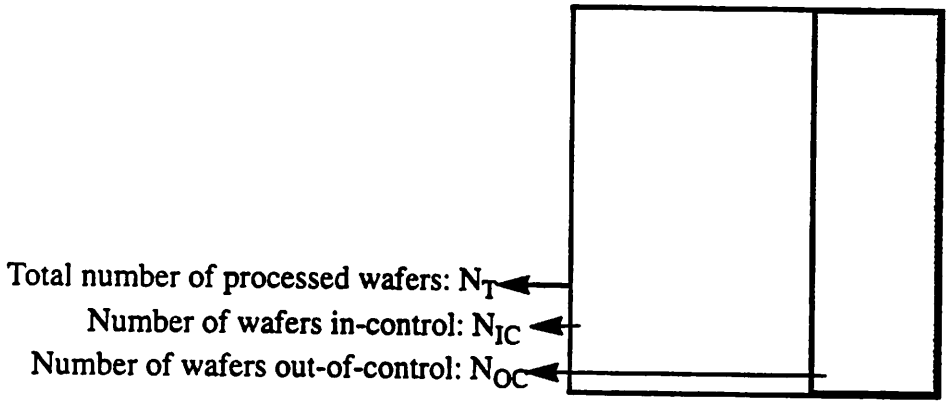
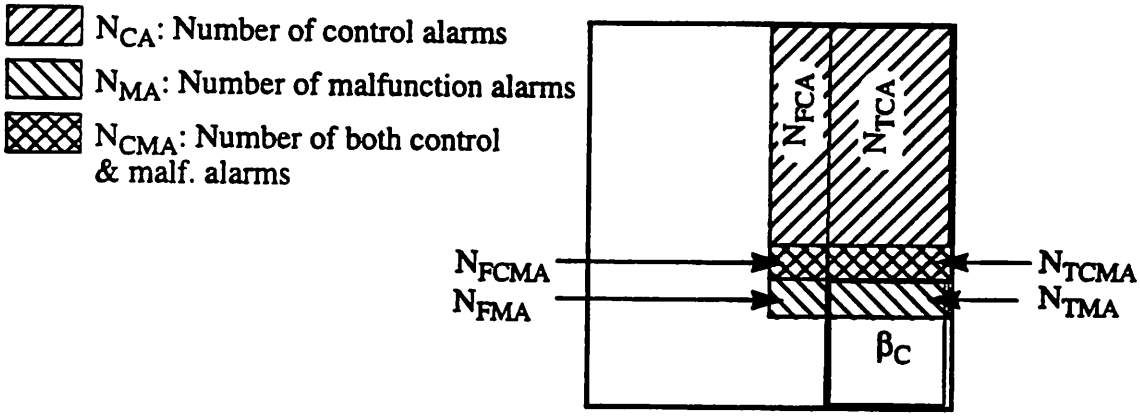


Figure 6.4 VEM Diagram Showing In-control / Out-of-control Wafers



N_{FCA}: # of false control (only) alarms
N_{FCMA}: # of false control & malf. alarms
N_{FMA}: # of false malf. (only) alarms
N_{TCA}: # of true control (only) alarms
N_{TCMA}: # of true control & malf. alarms
N_{TMA}: # of true malf. (only) alarms

Type I error of a Control alarm: $\alpha_C = (N_{FCA} + N_{FCMA}) / (N_{CA} + N_{CMA})$
Type II error of a Control alarm: $\beta_C = (N_{OC} - (N_{TCA} + N_{TCMA})) / N_{OC}$
Type I error of a Malf. alarm: $\alpha_M = (N_{FMA} + N_{FCMA}) / (N_{MA} + N_{CMA})$

Figure 6.5 VEM Diagram Showing All Possible States of an Alarm.

Let $\bar{C}$ and C represent the out-of-control and in-control states of a process, respectively. We will now calculate the probability of a true control alarm, $p(A_C, \bar{C})$, the probability of a false control alarm, $p(A_C, C)$, the probability of missing a control alarm, $p(\bar{A}_C, \bar{C})$, and the probability of having no control alarms when the process is in-control, $p(\bar{A}_C, C)$. Therefore, we defined these probabilities as follows:

$$p(A_C, \bar{C}) = \left(\frac{N_{TCA} + N_{TCMA}}{N_{OC}}\right) \cdot \frac{N_{OC}}{N_T} = \frac{(1 - \alpha_C)(N_{CA} + N_{CMA})}{N_T} \tag{6.24}$$

$$p(A_C, C) = \left(\frac{N_{FCA} + N_{FCMA}}{N_{IC}}\right) \cdot \frac{N_{IC}}{N_T} = \frac{\alpha_C(N_{CA} + N_{CMA})}{N_T} \tag{6.25}$$

$$p(\bar{A}_C, \bar{C}) = \beta_C \cdot p(\bar{C}) \tag{6.26}$$

$$p(\bar{A}_C, C) = \left(1 - \frac{(N_{FCA} + N_{FCMA})}{N_{IC}}\right) \cdot \frac{N_{IC}}{N_T} = \left(1 - \frac{(N_{FCA} + N_{FCMA})}{N_{IC}}\right) \cdot p(C) \tag{6.27}$$

If a control alarm has been triggered, the probability of a true control alarm, $p(E_4^C)$, and that of a false control alarm, $p(E_4^0)$, are respectively:

$$p(E_4^C) = p(A_C, \bar{C}|A_C) = \frac{p(A_C, \bar{C})}{p(A_C)} = \frac{p(A_C, \bar{C})}{p(A_C, \bar{C}) + p(A_C, C)} = 1 - \alpha_C \quad (6.28)$$

$$p(E_4^0) = p(A_C, C|A_C) = \frac{p(A_C, C)}{p(A_C)} = \frac{p(A_C, C)}{p(A_C, \bar{C}) + p(A_C, C)} = \alpha_C \quad (6.29)$$

Similarly, we calculate now the probability of a true malfunction alarm, $p(A_M, \bar{C})$, the probability of a false malfunction alarm, $p(A_M, C)$, the probability of missing a malfunction alarm, $p(\bar{A}_M, \bar{C})$, and the probability of having no malfunction alarms when the process is in-control, $p(\bar{A}_M, C)$.

$$p(A_M, \bar{C}) = \left(\frac{(1-\alpha_M)(N_{TMA} + N_{TCMA})}{N_{OC}}\right) \cdot \frac{N_{OC}}{N_T} \quad (6.30)$$

$$p(A_M, C) = \left(\frac{\alpha_M(N_{TMA} + N_{TCMA})}{N_{IC}}\right) \cdot \frac{N_{IC}}{N_T} \quad (6.31)$$

$$p(\bar{A}_M, \bar{C}) = \left(1 - \frac{(1-\alpha_M)(N_{TMA} + N_{TCMA})}{N_{OC}}\right) \cdot \frac{N_{OC}}{N_T} \quad (6.32)$$

$$p(\bar{A}_M, C) = \left(1 - \frac{\alpha_M(N_{TMA} + N_{TCMA})}{N_{IC}}\right) \cdot \frac{N_{IC}}{N_T} \quad (6.33)$$

If a malfunction alarm has been triggered, the probability of a true malfunction alarm, $p(E_4^M)$, and that of a false malfunction alarm, $p(E_4^0)$, are given by:

$$p(E_4^M) = p(A_M, \bar{C}|A_M) = \frac{p(A_M, \bar{C})}{p(A_M)} = \frac{p(A_M, \bar{C})}{p(A_M, \bar{C}) + p(A_M, C)} = 1 - \alpha_M \quad (6.34)$$

$$p(E_4^0) = p(A_M, C|A_M) = \frac{p(A_M, C)}{p(A_M)} = \frac{p(A_M, C)}{p(A_M, \bar{C}) + p(A_M, C)} = \alpha_M \quad (6.35)$$

Note that in our control system, a malfunction alarm takes precedence over a control alarm, i.e, a control alarm is defined as an alarm triggered by the control alarm mechanism only, whereas a malfunction alarm includes alarms triggered by the malfunction alarm

mechanism alone AND alarms triggered by both malfunction and control mechanisms. Therefore, if both alarms are triggered, they will be treated as a malfunction alarm only. Values for the type I errors of the malfunction and control alarms, α_M and α_C, and the value of the type II error of the control alarm, β_C, are specified by the user, when activating the controller.

6.7.5 Probability of Categories of a "Hypothetical Input Change" (E_5)

The piece of evidence "*Fit of Predicted Outputs from a Hypothetical Input Change to the Measurements*", is the only one based on deep level knowledge. Extracted from the equipment models and measurement values, it corresponds to how well the predicted output(s), assuming a hypothetical input change, matches the measurements. It is similar to the algorithms used by May [53], and Saxena and Unruh [85], but instead of attributing the probability directly to a faulty input, our algorithm attributes the probability instead to the evidence that a faulty input could have caused the problem. Then, if multiple inputs could each have caused the problem, the diagnostic system will calculate the probability of each one, depending on their past frequency (equ. (6.5)). This piece of evidence is divided into as many variables as there are inputs to the machine, because each variable corresponds to the fit of the predicted outputs to the measurements, assuming only that a single input has changed. Each variable is divided into 2 categories, $\{ E^{+}_{5,i}, E^{-}_{5,i} \}$ (i = 1, ..., n and n is the number of inputs). The first category corresponds to when the outputs predicted by a change of the input match the measurements exactly, while the second one corresponds to when they do not match at all. The computation of the probability of the category is based on solving backwards the equipment models using the measurements, and then analyzing the difference between the predicted outputs assuming the hypothetical input change and the actual measurements.

The problem of solving the equipment models backwards can be stated as follows. Let p be the number of process outputs, and t the number of model input terms (such as x_1, x_2, x_1^2, x_1/x_3, ...), an equipment model takes the following form [53]:

$$\underset{\sim}{Y} = K\underset{\sim}{X} + \underset{\sim}{E} \tag{6.36}$$

where $\underset{\sim}{Y}$ is a column vector of p normalized responses; $\underset{\sim}{X}$, a column vector of t model input terms; K, a pxt array of regression coefficients; and $\underset{\sim}{E}$, a pxl residual column vector. Note that the p responses must be standardized to avoid ill-conditioned matrix calculations, due to their widely divergent magnitudes. This becomes especially important when their residuals will be combined together in subsequent analysis. To standardize responses, transform them as follows:

$$\underset{\sim}{Y} = \sigma^{-1}(\underset{\sim}{y} - \underset{\sim}{\bar{y}}_0) \tag{6.37}$$

where $\underset{\sim}{y}$ is a column vector of the p responses; $\underset{\sim}{\bar{y}}_0$, a column vector of the p responses taken at the operating point from the designed experiments; and σ, the diagnonal pxp array of standard deviations of the process outputs.

$$\sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & \sigma_p \end{bmatrix} \tag{6.38}$$

Generally, the input terms of an equipment model are *nonlinear*, consisting of nonlinear combinations of input settings. To solve the process outputs' models in reverse, they must linearized with respect to the input settings. This is achieved by approximating them by a linear truncated Taylor series expansion.

$$\hat{\underset{\sim}{Y}} + \Delta\hat{\underset{\sim}{Y}} = K\underset{\sim}{X} + K\frac{\partial}{\partial x_i}\underset{\sim}{X}\Delta x_i, \; i = 1, \dots, n \tag{6.39}$$

where x_i is one of n input settings and Δx_i is a scalar representing a change in input x_i.

Next, equ. (6.39) is solved backwards by minimizing the sum of squares of the residuals:

$$f(\Delta x_i) = \left(\underset{\sim}{Y} - K\underset{\sim}{X} - K\frac{\partial}{\partial x_i}\underset{\sim}{X}\Delta x_i\right)^2 , i = 1, ..., n \tag{6.40}$$

Note that the residuals do not need to be weighted since the responses have been already previously normalized. The column vector of residuals $\underset{\sim}{R}$ is given by:

$$\underset{\sim}{R} = \underset{\sim}{Y} - K\underset{\sim}{X} - K\frac{\partial}{\partial x_i}\underset{\sim}{X}\Delta x_i \tag{6.41}$$

The solution to equ. (6.40) is found by setting the derivative of $f(\Delta x_i)$ to 0, and solving for Δx_i.

$$\Delta x_i = \frac{\left(K\frac{\partial}{\partial x_i}\underset{\sim}{X}\right)^T\underset{\sim}{Y} - \left(K\frac{\partial}{\partial x_i}\underset{\sim}{X}\right)^T K\underset{\sim}{X}}{\left(K\frac{\partial}{\partial x_i}\underset{\sim}{X}\right)^T\left(K\frac{\partial}{\partial x_i}\underset{\sim}{X}\right)} \tag{6.42}$$

To check the significance of the input shift Δx_i, we check if all residuals are normally distributed around zero. If confirmed, there is a strong probability that the outputs' shifts can be attributed to the faulty input parameter in question. We apply the Hotelling's T^2 statistic on $\underset{\sim}{R}$ to determine if it is not significantly different from zero.

$$T^2_{\alpha, q, N-q} = \mathcal{N}\underset{\sim}{R}^T\Sigma^{-1}\underset{\sim}{R} \tag{6.43}$$

where Σ is the covariance matrix of the original designed experiment, i.e the one used in developing the equipment model, and acts as an estimate of the actual covariance matrix of the present runs. $\mathcal{N}$ is the sample size of the outputs during production runs. Finally, to find the statistical significance that Δx_i has actually caused the observed output shifts, we find the α that satisfies

$$T^2_{\alpha, q, N-q} = \frac{q \cdot (N-1) \cdot F_\alpha(q, N-q)}{N-q} , \tag{6.44}$$

We then interpret that significance α as the probability that the predicted outputs match the measurements, assuming the input change Δx_i:

$$p(E^{+}_{5,i}) = \alpha_i \text{ and } p(E^{-}_{5,i}) = 1 - \alpha_i \tag{6.45}$$

6.7.6 Example of Evidence Probability Calculation

To clarify how to calculate the probability of a combination of evidence, we calculate the probability of the combination that best fits the following example.

A control alarm was triggered on the wafer track. Its α_C and β_C error parameters were 5% and 20% respectively. The type I error α_M of the malfunction alarm was also set at 5%. The thickness value was 12330Å. The predicted value was 12240Å. The standard deviation of the thickness during processing was 65Å. The PAC value was 0.97. The predicted value was 0.98. Its standard deviation was 0.023. There was no circular pattern, nor any streak on the wafer. (These are the only relevant observable parameters for this machine.) The reference wafer has just been cleaned.

Other pertinent data are: the diagnostic system has recorded up to now 31 malfunction alarms and 102 control alarms. The number of measurement errors is 5. The number of total processed wafers is 562. The characteristic life of the reference wafer is 11 days.

All the pieces of evidence for this machine, and their categories are summarized below, with the most probable category being highlighted in bold.

Piece of Evidence	Variables	Values		
Operator Observation (E_1)	Circular Patterns ($E_{1,1}$)	True (+) prob = 0.0	**False (-) prob = 1.0**	
	Streaks ($E_{1,2}$)	True (+) prob = 0.0	**False (-) prob = 1.0**	
Age of Machine Component (E_2)	Reference Wafer Age ($E_{2,1}$)	Old (+) prob = 0.0	**New (-) prob = 1.0**	
Machine Output (E_3)	Thickness ($E_{3,1}$)	**Above Target (+) prob = 0.9614**	Below Target (-) prob = 0.0386	
	PAC ($E_{3,2}$)	Above Target (+) prob = 0.335	**Below Target (-) prob = 0.665**	
Type of Alarm (E_4)	N/A	Malfunction Alarm (M) prob = 0.0	**Control Alarm (C) prob = 0.965**	False Alarm (0) prob = 0.035
Fit of Output Predicted by a Hypothetical Input Change to the Measurement† (E_5)	Wrong Input Spin Speed ($E_{5,1}$)	**Perfect Fit prob = 0.99**	No Fit prob = 0.01	
	Wrong Input Spin Time($E_{5,2}$)	Perfect Fit prob = 0.001	**No Fit prob = 0.999**	
	Wrong Input Bake Temp ($E_{5,3}$)	**Perfect Fit prob = 0.95**	No Fit prob = 0.05	
	Wrong Input Bake Time ($E_{5,4}$)	Perfect Fit prob = 0.01	**No Fit prob = 0.99**	
	Wrong Input Humidity ($E_{5,5}$)	**Perfect Fit prob = 0.999**	No Fit prob = 0.001	
	Wrong Input Bottle Level ($E_{5,6}$)	Perfect Fit prob = 0.01	**No Fit prob = 0.99**	

Most probable combination of evidence:

$$\max_{\text{over all } j} p(C_j) = p(E^-_{1,1}E^-_{1,2}E^-_{2,1}E^+_{3,1}E^-_{3,2}E^C_4E^+_{5,1}E^-_{5,2}E^+_{5,3}E^-_{5,4}E^+_{5,5}E^-_{5,6})$$

Figure 6.6 How Evidence from a Wafer Track Gets Categorized into Combinations

1. $p(E^-_{1,1}) = p(E^-_{1,2}) = 1.0$
2. Cumulative distribution of the reference wafer's lifespan: (16.7 percentile, 63.2 percentile, 97.4 percentile) = (5, 11, 21) days.

$$B = 2.989/\log(21/5) = 4.80 \Rightarrow p(E^-_{2,1}) = \exp[-(1/11)^{4.80}] \approx 1.0$$

3. $$p(E^{+}_{3,1}\,|\,O_g) = 1 - \Phi\left(\frac{12240 - 12330}{22}\right) = 0.96562 \quad .$$

$$\Rightarrow p(E^{+}_{3,1}) = 0.96562 \cdot \frac{557}{562} + 0.5 \cdot \frac{5}{562} = 0.9614$$

4. $$p(E^{-}_{3,2}\,|\,O_g) = \Phi\left(\frac{0.98 - 0.97}{0.023}\right) = 0.6664$$

$$\Rightarrow p(E^{-}_{3,2}) = 0.6664 \cdot \frac{557}{562} + 0.5 \cdot \frac{5}{562} = 0.665$$

5. $$N_{OC} = \frac{(1 - 0.05)(102 + 10)}{1 - 0.2} = 133$$

$$\Rightarrow p(E^{C}_{4}) = \frac{(1 - 0.2)133 - 8}{(1 - 0.2)133 + 0.05 \cdot 102 - (1 - 0.05)10} = 0.965$$

6. $$T^2_{SPS} = 0.04 \Rightarrow p(E^{+}_{5,1}) = 0.99$$

7. $$T^2_{SPT} = 1.76 \Rightarrow p(E^{-}_{5,2}) = 0.999$$

8. $$T^2_{BTE} = 0.21 \Rightarrow p(E^{+}_{5,3}) = 0.95$$

9. $$T^2_{BTI} = 2.01 \Rightarrow p(E^{-}_{5,4}) = 0.99$$

10. $$T^2_{H} = 0.02 \Rightarrow p(E^{+}_{5,5}) = 0.999$$

11. $$T^2_{BL} = 1.60 \Rightarrow p(E^{-}_{5,6}) = 0.99$$

The final probability that the observed symptoms fit this combination of evidence is:

$$p(E^{-}_{1,1}E^{-}_{1,2}E^{-}_{2,1}E^{+}_{3,1}E^{-}_{3,2}E^{C}_{4}E^{+}_{5,1}E^{-}_{5,2}E^{+}_{5,3}E^{-}_{5,4}E^{+}_{5,5}E^{-}_{5,6}) = \qquad (6.46)$$

$$1 \cdot 1 \cdot 1 \cdot 0.9614 \cdot 0.665 \cdot 0.965 \cdot 0.99 \cdot 0.999 \cdot 0.95 \cdot 0.99 \cdot 0.999 \cdot 0.99 = 0.568$$

6.8 Determination of Conditional Probabilities

The conditional probabilities of faults correspond to the relative frequency of faults given a combination of evidence. They are typically obtained from the diagnosis database

(equ. (6.8)). Initially, however, these conditional probabilities must be estimated from the experience of machine operators. To decrease the subjectivity of the expert opinions and increase the accuracy of the estimates, we poll several machine operators, and then combine their opinions into one single set of conditional probabilities.

6.8.1 Previous Work on Information Filtering

The problem of filtering information from human experts and assessing their estimations has been thoroughly attacked in the past. Two versions of the problem have been studied in the literature. In the first version, experts evaluate characteristics of objects, such as the height of a tower, or the speed of a car [65][66][71]. The methodology then consists of first calibrating the human experts, to see whether they tend to be overconfident or conservative with their estimates. By asking the experts to predict several parameters and then comparing their predictions to the actual values known to the testers, the testers quantify the biases of the human experts. Finally, the concensus of all the estimates is obtained by taking the most likely value from the "calibrated" experts.

In our case, such a methodology cannot be used because our problem addresses probabilities of events, whose sum must always equal one. The problem rises from the calibration of the predictions. If an expert tends to be too conservative with his probability estimates, we cannot compensate his predictions by increasing them by a certain amount. Many papers have studied various facets of the concensus problem when the experts are trying to predict probabilities [67][68][72][73][74][75][76]. These papers are more useful to our work.

6.8.2 Choosing a Methodology for Combining Probability Estimates

The methodology consists of collecting subjective probability estimates from human experts, and then combining them using either a weighted average scheme [67][72][74], or first transforming them into their natural conjugates and then combining them using

Bayes' Theorem [68][72][73][75]. The weights are based on ranking the human experts, using one of several scoring schemes, described in [72].

In order to obtain accurate estimates of probability distributions, Winkler has documented a methodology for interviewing experts [75]. The questionnaire uses four different techniques, CDF - Cumulative Distribution Function, HFS - Hypothetical Future Sample, EPS - Equivalent Prior Sample Information, and PDF - Probability Density Function. A distribution of the probability estimate is obtained from each technique. If they are widely different, it implies that the expert did not understand one or more questions, and after being trained on the concepts, is asked to give his/her probability distribution estimates again. On the other hand, if the probability distributions are close to each other, they are averaged into one single distribution.

Next, we must choose between either a weighted average (W-A) or a natural conjugate (N-C) method to combine the probability distributions. Winkler investigated that problem using a Bernoulli process to generate data. This was done assuming that there are only two experts, in order to simplify the situation. (The notation used in the figures below to denote a probability distribution is $p_{i,j(r,n)}$. The subscript i corresponds to the fault F_i, and the subscript j, to the j-th expert. The subscript r corresponds to the number of "successes" observed by expert j, and n is the number of data experienced or observed by expert j.)

To briefly compare the two methods, we look at two examples. In the first one, two experts with different experiences provide an estimate of a probability distribution of a fault, call it F_1, centered around 0.5, given a specific problem. The estimate of the first expert is based on 10 observations of that problem, while the estimate of the second expert is based on 100 observations. The combination of both probability distributions under both W-A and N-C methods is schematically shown in Figure 6.7. Note that the N-C method results in a tighter combined probability distribution than the W-A method.

Next, we assume divergent estimates of probability distributions between the two experts. Both experts now claim to have observed 20 occurrences of the problem, but while the first expert finds only 10% of the occurrences were caused by the fault F_1, the second expert claims that 50% of the occurrences were caused by F_1. Under the W-A method, the combination of the two probability distributions results in a bimodal distribution, that can lead to an unstable system. The N-C method, on the other hand, always results in a unimodal probability distribution, which not only leads to a more stable diagnosis system, but also follows our intuition better. Therefore, we have chosen to use the N-C method to combine the probability distribution of the estimates given by our experts.

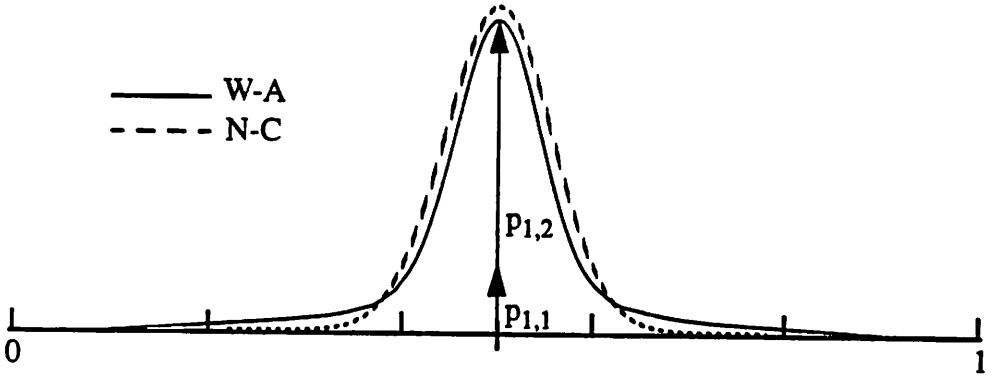


Figure 6.7 W-A and N-C Combinations of $p_{1,1(5, 10)}$ and $p_{1,2(50, 100)}$ with Equal Weights

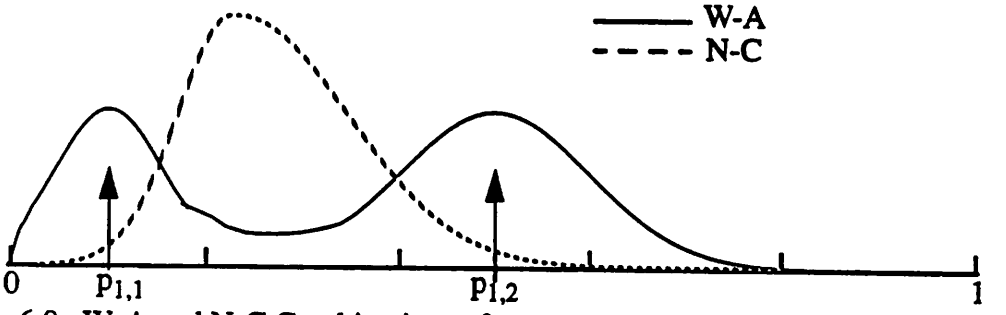


Figure 6.8 W-A and N-C Combinations of $p_{1,1(2, 20)}$ and $p_{1,2(10, 20)}$ with Equal Weights

6.8.3 Application of the Natural Conjugate Combination Method

One disadvantage of using the N-C method is that our data are generated from a multinomial process instead of a binomial process. We were unable to find in the literature, nor could we derive from its definition, the natural-conjugate family related to the multinomial function.

To circumvent this problem, we decompose a multinomial function into a series of binomial functions. After all, a multinomial distribution problem which consists of three probability distributions, $p_{1,j}$, $p_{2,j}$, and $p_{3,j}$, can be divided into three binomial problems. When finding the concensus of $p_{1,j}$ from all the experts, the multinomial distribution problem can be transformed into a binomial distribution problem, by grouping $p_{2,j}$, and $p_{3,j}$ into one single probability $\bar{p}_{1,j}$. Similarly, when finding the concensus of $p_{2,j}$, and $p_{3,j}$ from all the experts, the problem can be solved using the natural conjugate solution of a binomial process, by grouping $p_{1,j}$, and $p_{3,j}$ into $\bar{p}_{2,j}$ in the first case, and $p_{1,j}$, and $p_{2,j}$ into $\bar{p}_{3,j}$ in the second case.

Now, we determine how to combine all the estimates of $p_{1,j}$, for example, given by the experts into one single distribution $p_{1,final}$. This methodology has been developed by Winkler [72], and the reader is refered to that paper, if a more thorough explaination is needed. Here, we only describe the methodology. The natural conjugate function used to combine probabilities generated from a Bernoulli process is the beta distribution function.

$$f_{\beta}(p) = \frac{(n-1)!}{(r-1)!(n-r-1)!} p^{r-1}(1-p)^{n-r-1} \tag{6.47}$$

As mentioned before, n is the number of times a combination of evidence has been experienced by expert "j", and r is the number of times fault F_i has been diagnosed as the cause by expert "j". $f_{\beta}(p)$ gives the probability that $p_{1,final}$ equals the value p. These two parameters, (r, n), describe the beta distribution. To combine multiple beta distributions (r_m, n_m),

such as the case when we want to combine the judgements of multiple experts, we just add the r's and n's together:

$$r_{final} = \sum_{m=1}^{k} r_m \text{ and } n_{final} = \sum_{m=1}^{k} n_m \tag{6.48}$$

where k is the number of experts polled.

To deal with dependent experts, Winkler has introduced a weight for r's and n's, i.e:

$$r_{final} = w\left(\sum_{m=1}^{k} r_m\right) \text{ and } n_{final} = w\left(\sum_{m=1}^{k} n_m\right) \tag{6.49}$$

If the experience of the experts originate from completely separate and distinct samples, w equals 1. On the other extreme, if the experience of the experts originate from the exact same sample, w equals 1/k. Beyond that guideline, the decision on the value of w is purely subjective.

Once we have the distribution of the final concensus of the fault probability $p_{1,final}$, which is of the form of a beta distribution with parameters (r_{final}, n_{final}), we can quantify both the final concensus value $\hat{p}_{1,final}$, and the uncertainty about $\hat{p}_{1,final}$. More specifically, $\hat{p}_{1,final}$ is represented by the average value $E(p_{1,final})$, while the uncertainty is quantified by the variance $Var(p_{1,final})$. Both parameters are given by [77]:

$$\hat{p}_{1,final} = E(p_{1,final}) = \frac{r_{final}}{n_{final}} \text{ and } Var(p_{1,final}) = \frac{r_{final}(n_{final} - r_{final})}{n_{final}^2(n_{final} + 1)} \tag{6.50}$$

Before closing, we would like to warn experts against choosing estimates of conditional probability values of 1.0, because they are degenerate, even though some faults may have such a distinct signature that it is very tempting to give out estimates of conditional probabilities of 1.0. Such an absolute assignment would lead to a very unstable diagnostic system, since a conditional probability of 1.0 means that we are certain of the fault, and that we do not need any training points to confirm it. To guard against such a scenario, our

current diagnostic system automatically replaces any conditional probability values of 1.0 by 0.99 and giving a 1% probability to the No-Fault category.

6.8.4 Linking the Beta Distributions of the Initial Conditional Probability Estimates to the Binomial Distribution of the Conditional Probabilities

The initial estimates of the conditional probabilities' distributions given by the experts are combined into a single beta distribution, whose average value and variance are described by equ. (6.50). As it will be shown in the subsequent section however, conditional probabilities follow a multinomial distribution over time, assuming the faults are independent of each other, and that their probabilities are constants. Therefore, we must transform the beta distributions of the initial estimates into multinomial distributions.

The average value and variance of an initial estimate of a conditional probability distribution are (from equ. (6.50)):

$$\hat{p}_{1,\,final} = E(p_{1,\,final}) = \frac{r_{final}}{n_{final}} \text{ and } Var(p_{1,\,final}) = \frac{\hat{p}_{1,\,final} \cdot (1 - \hat{p}_{1,\,final})}{n_{final} + 1} \qquad (6.51)$$

The average value and variance of a subsequently updated estimate of a conditional probability, which follows a multinomial distribution, are (from equ. (6.58)):

$$E_1 = p_1 \text{ and } V = \frac{p_1 \cdot (1 - p_1)}{N} \qquad (6.52)$$

where p_1 is the probability of fault F_1 and N, the total number of all faults for that combination of evidence. If we equate $p_1 = \hat{p}_{1,\,final}$, the variances of the two distributions are very close to each other. Therefore, we transform the beta distributions of the initial estimates of the conditional probabilities into multinomial distributions simply by equating:

$$p_1 = \hat{p}_{1,\,final}, \text{ and } N = n_{final} + 1 \qquad (6.53)$$

6.9 Analysis of the Accuracy of Fault Probability Values

Often, in addition to the given fault probabilities, operators are also interested in the range of these values, given a specified confidence level. Before addressing the more complex issue of a combination of evidence that leads to multiple faults, we first examine the case when a combination of evidence leads to only two faults, F_1 and F_2. Assuming that they are independent of each other, and that their probabilities, p_1 and p_2, are constant, the number of occurrences of fault F_1, n_1, among N occurrences of faults F_1 and F_2, follows a binomial distribution [21]:

$$P = \frac{N!}{n_1! \cdot n_2!} \cdot p_1^{n_1} \cdot p_2^{n_2} = \frac{N!}{n_1! \cdot (N-n_1)!} \cdot p_1^{n_1} \cdot (1-p_1)^{N-n_1} \tag{6.54}$$

Note that the binomial distribution is completely determined by n_1, and N. The mean and variance of this distribution is given by [21]:

$$E(n_1) = N \cdot p_1 \text{ and } V(n_1) = N \cdot p_1 \cdot (1-p_1) \tag{6.55}$$

In our case though, we are interested in the random variable $\hat{p}_1$, instead of n_1, because $\hat{p}_1$ is an estimate of p_1:

$$\hat{p}_1 = \frac{n_1}{N} \tag{6.56}$$

The probability distribution of $\hat{p}_1$ is easily obtained from that of n_1:

$$\hat{P} = \frac{P}{N} = \frac{(N-1)!}{n_1! \cdot (N-n_1)!} \cdot p_1^{n_1} \cdot (1-p_1)^{N-n_1} \tag{6.57}$$

Its mean and variance are [21]:

$$E(\hat{p}_1) = p_1 \text{ and } V(\hat{p}_1) = \frac{p_1 \cdot (1-p_1)}{N} \tag{6.58}$$

We extend this analysis now to a multi-dimensional space, since a category of evidence can lead to more than two faults. Assuming that the faults F_i (i = 1, ..., k) are independent of each other, and that their probabilities p_i are constant, the number of

occurrences of fault F_i, n_i, among N occurrences of faults follows a multinomial distribution [59]:

$$P = \frac{N!}{n_1! \cdot \ldots \cdot n_k!} \cdot \prod_{i=1}^{k} p_i^{n_i}, i = 1, ..., k \;\&\; \sum_{i=1}^{k} n_i = N \tag{6.59}$$

The mean and covariance matrix of this distribution are given by [59]:

$$\mathbf{E}(\mathbf{n}) = N \cdot \mathbf{p} \text{ and } \mathbf{V}(\mathbf{n}) = N[\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^T] \tag{6.60}$$

where $\mathbf{p}$ is the vector of fault probabilities. As for the estimated fault probability $\hat{p}_i$, i = 1, ..., k, defined as:

$$\hat{p}_i = \frac{n_i}{N}, \text{ with } \sum_{i=1}^{k} n_i = N, \tag{6.61}$$

its probability function is obtained from the multinomial distribution:

$$\hat{P} = \frac{P}{N} = \frac{(N-1)!}{n_1! \cdot \ldots \cdot n_k!} \cdot \prod_{i=1}^{k} p_i^{n_i} \tag{6.62}$$

The mean vector and covariance matrix are given by [59]:

$$\mathbf{E}(\hat{\mathbf{p}}) = \mathbf{p} \text{ and } \mathbf{V}(\hat{\mathbf{p}}) = \frac{1}{N}[\text{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^T] \tag{6.63}$$

Now that the probability density function of the vector $\hat{\mathbf{p}}$ is known, its upper and lower bounds, $\hat{\mathbf{p}}_U$ and $\hat{\mathbf{p}}_L$, can be calculated for any level of confidence C. If the area under the probability density function is normalized to 1.0, C is defined as follows:

$$C = 1 - \Pr(\hat{\mathbf{p}} < \hat{\mathbf{p}}_L) - \Pr(\hat{\mathbf{p}} > \hat{\mathbf{p}}_U) \tag{6.64}$$

where $\Pr(\hat{\mathbf{p}} < \hat{\mathbf{p}}_L)$ corresponds to the area under the probability density function with $\hat{\mathbf{p}}$ being between 0 and $\hat{\mathbf{p}}_L$, and $\Pr(\hat{\mathbf{p}} > \hat{\mathbf{p}}_U)$ corresponds to the area under the probability density function with $\hat{\mathbf{p}}$ being between $\hat{\mathbf{p}}_U$ and 1.0.

Instead of integrating the area underneath the probability density function of $\hat{\mathbf{p}}$, we choose to work with the probability density function of the vector $\mathbf{n}$, because its multino-

mial distribution has been better studied in the literature. For a specific n_j, a component of the vector $\mathbf{n}$, the probability that n_j is less than a number $n_{j,L}$ is given by [77][78]:

$$Pr(n_j < n_{j,L}) = I_{1-p_j}(N - n_{j,L} + 1, n_{j,L}) \tag{6.65}$$

where $I_X(m,n)$ is the incomplete beta function ratio [77]:

$$I_X(m, n) = \frac{1}{B(m, n)} \int_0^X x^{m-1}(1-x)^{n-1} dx \tag{6.66}$$

To calculate this integral, we use the following approximation [77]:

$$I_X(m, n) \cong \Phi(Z) \tag{6.67}$$

where Z is:

$$Z = \frac{d}{|n - 0.5 - N(1-X)|} \cdot \left(\frac{2}{1 + 1/(6N)}\right)^{0.5} \cdot \left[(n-0.5)\log\left\{\frac{n-0.5}{N(1-X)}\right\} + (m-0.5)\log\left\{\frac{m-0.5}{NX}\right\}\right]^{0.5} \tag{6.68}$$

and d is:

$$d = n - \frac{1}{3} - \left(N + \frac{1}{3}\right)(1-X) + 0.2(X/n) - (1-X)/m + \frac{(X-0.5)}{m+n} \tag{6.69}$$

For example, the lower and upper bounds of a fault probability p_j are calculated as follows, given a desired confidence level of 90%. There has been 50 training points for the combination of evidence, i.e. N = 50, and the current estimate of p_j is 0.3.

A 90% confidence level translates into finding Z_1 and Z_2, such that $\Phi(Z_1) = 0.05$ and $\Phi(Z_2) = 0.95$ (equ. (6.64)). These correspond respectively to $Z_1 = -1.64$ and $Z_2 = 1.64$ [21]. Next, we find $n_{j,L}$ and $n_{j,U}$ such that:

$$Pr(n_j < n_{j,L}) = I_{1-0.3}(50 - n_{j,L} + 1, n_{j,L}) = \Phi(Z_1) \tag{6.70}$$

$$Pr(n_j < n_{j,U}) = I_{1-0.3}(50 - n_{j,U} + 1, n_{j,U}) = \Phi(Z_2) \tag{6.71}$$

To solve for $n_{j,L}$ and $n_{j,U}$, we solve equ. (6.68) with (X, m, n, Z) replaced by (0.7, 50 - $n_{j,L}$ +1, $n_{j,L}$, -1.64), and then with (X, m, n, Z) replaced by (0.7, 50 - $n_{j,U}$ +1, $n_{j,U}$, 1.64). We calculate $I_{1-0.3}(50-n_{j,L}+1, n_{j,L})$, varying $n_{j,L}$ between 1 and 50, and do the same for $n_{j,U}$. We have found that $n_{j,L}$ is about 10, and that $n_{j,U}$ is about 21. Therefore, the lower and upper bounds on p_j are:

$$\hat{p}_{j,L} \approx \frac{10}{50} = 0.2 \text{ and } \hat{p}_{j,U} \approx \frac{21}{50} = 0.42 \tag{6.72}$$

6.10 Knowledge Base

The knowledge bases of the three photolithography machines are shown in the figures below.

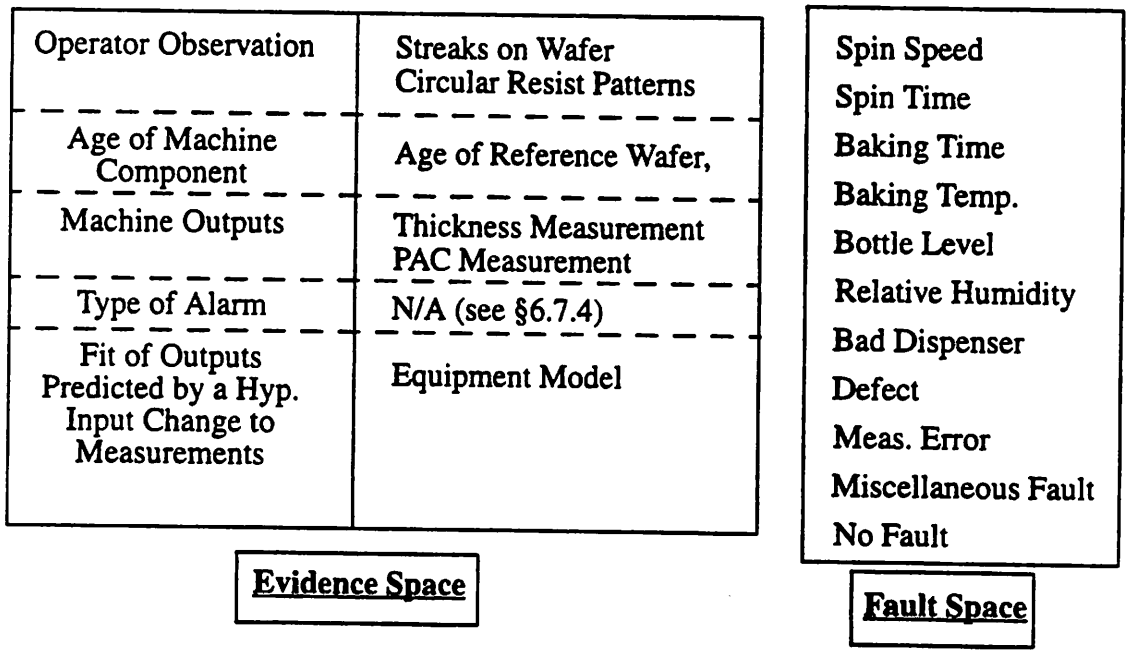


Figure 6.9 Schematic of the Knowledge Base of the Wafer Track

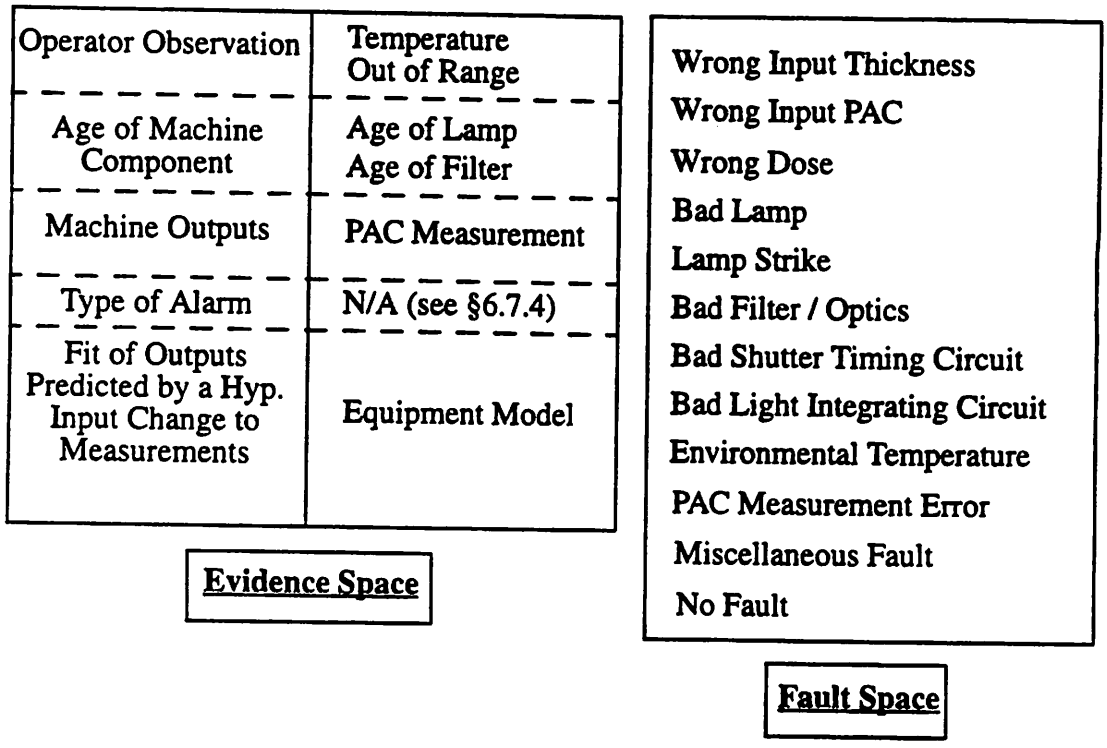


Figure 6.10 Schematic of the Knowledge Base of the Stepper

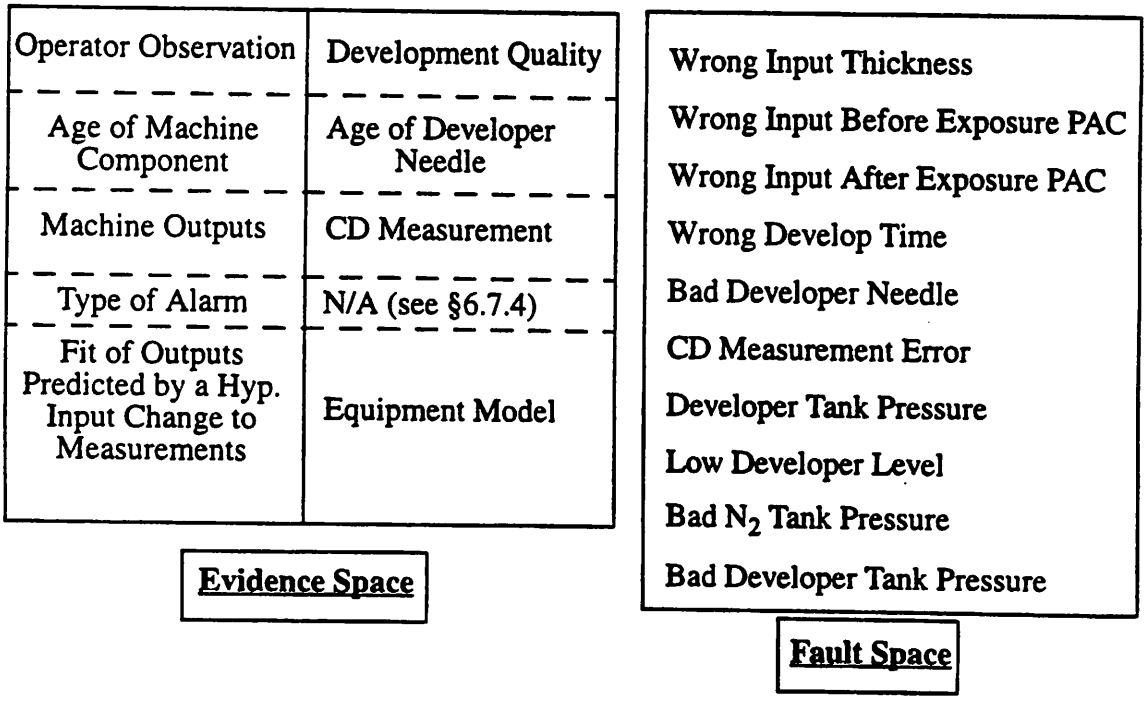


Figure 6.11 Schematic of the Knowledge Base of the Developer

6.11 Conclusion

A practical diagnostic system has been developed for photolithography equipment. Its knowledge base uses both "shallow level"and "deep level" information for evidence, which include operator observations, measurements, maintenance records, alarm type and equipment models. Using basic probability theory, the diagnostic system calculates the probability of all the faults from conditional probabilities, initially supplied by machine experts, and subsequently automatically updated by the system. The procedure for combining the estimates of conditional probabilities, and their convergence properties have been discussed in detail in this chapter. Finally, the diagnostic system also determines from the number of diagnosis cases and the confidence level, specified by the user, the upper and lower bounds of the fault probabilities. A software implementation of the system has been developed and applied on the photolithography equipment in the Microlab. Experimental results are shown in the next chapter.

[This page is intentionally blank]

Chapter 7 Experimental and Simulated Results

7.1 Introduction

While previous chapters present the theory and implementation of the control and diagnostic system, this chapter presents the experimental results of the system. First, we describe how we collect data and screen for outliers. Then we test the capability of the process controller and diagnostic system.

7.2 Data Collection and Screening

While monitoring, not all readings are representative of the process. For example, streaks are produced occasionally during the spin-coat and bake step. If a thickness measurement is performed on the streak, it will be significantly different from the mean thickness on the wafer. Yet that measurement is not representative of the wafer. Wide ranges of measurements within the same sampled wafer are abnormal and are used by our screening procedure to filter outlying measurements. After taking several measurements on the same wafer, we apply them on a Range chart [21] (Figure 7.1). The wafer parameter of interest is the range of the measurements, R:

$$R = y_{max} - y_{min} \tag{7.1}$$

where y_{max} and y_{min} correspond to the highest and lowest measurement values within the sampled wafer. The upper control limit, UCL, of the Range chart with the usual 3-sigma control limit is given by:

$$UCL = \bar{R} + 3\frac{d_3}{d_2}\bar{R} \tag{7.2}$$

where $\bar{R}$ is the average range within all samples. d_3 is the standard deviation of the distribution of the relative range, and d_2, the mean of the distribution of the relative range. Both d_3 and d_2 are well documented functions of the sample size [21].

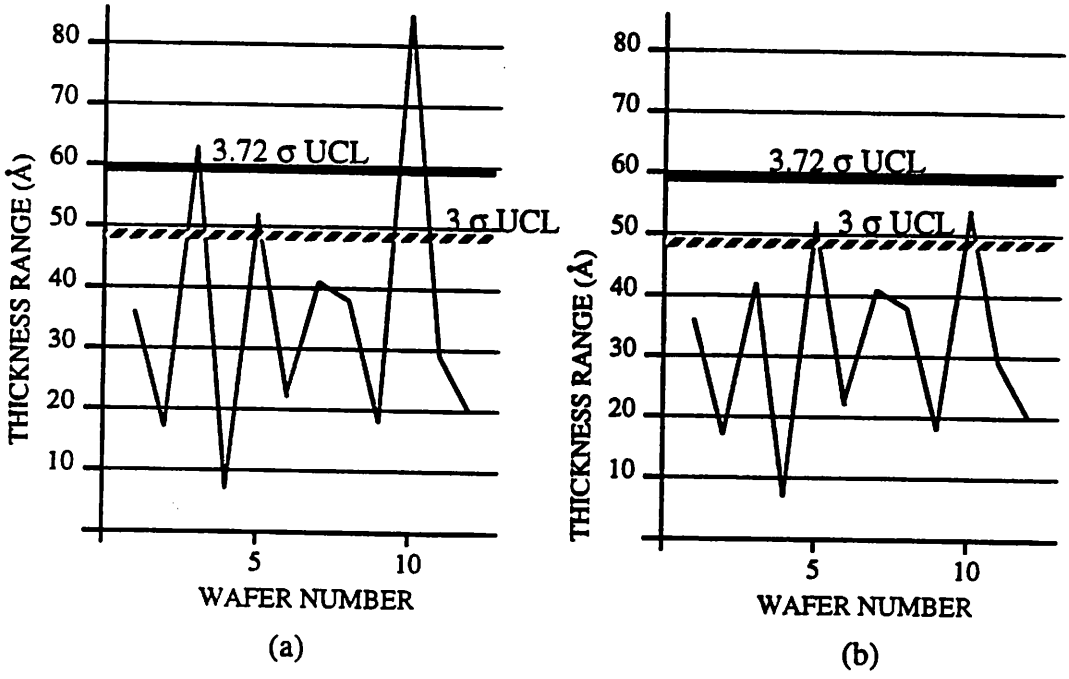


Figure 7.1 Thickness R-chart (a) Before Rejecting Outlying Measurements, and (b) After Rejecting Outlying Measurements.

If the range of a wafer violates the UCL, the screening program finds the measurement that lies furthest away from the others, by going through each measurement, and calculating the range of the remainder measurements. The measurement, which if voided, results in the smallest range, is then replaced by the average of the other measurements. The program then checks the range against the UCL again, and if it is violated, iterates until no more outliers are found. The caveat of this algorithm however is that the range of the measurements could become artificially reduced. To avoid such a case, we use a higher than 3 sigma UCL, such as a 3.3 sigma UCL or a 3.72 sigma UCL instead, which corresponds to a type I error of 0.05% or 0.01%, respectively.

7.3 Experimental Results of the Process Controller

Next, we evaluate the overall effectiveness of the process controller, by applying it to all three pieces of equipment: the wafer track, the stepper and the developer. First, we will apply only the feedback controller and analyze how much it improves the process capabil-

ity. Then, we activate both feed-forward and feedback controllers to see how much improvement feed-forward control brings.

7.3.1 Results of Feedback Control

In this section, we evaluate the overall effectiveness of the feedback controller, by implementing it on photolithography equipment in the Microlab [20]. Our experiment tests the performance of the wafer track, the stepper and the developer, under feedback control and under no control. The experiment consists of processing P-type 4" silicon wafers, coated with 1000Å oxide, through the photolithography sequence of spin-coat and bake, exposure, and develop. Control has been applied on a lot by lot basis instead of on a run-by-run basis, with each lot consisting of three wafers. The historical average of each machine output, when the machine was in-control, was chosen to be the target for the machine output. Each wafer is sampled four times, with the average reading being recorded. 60 wafers were divided into 20 *lots* (i.e, three wafers per lot). These 20 lots were then divided into 2 *groups* (i.e, 10 lots per group). One lot of wafers was processed each day, alternating between an uncontrolled baseline lot, and one subject to feedback control. Details of the experiment, which consist of machine outputs, alarms, and recipe changes, are summarized in the next four figures.

A comparison of the final CD distribution of the two groups of wafers confirms that the feedback controller is very efficient and successful in centering the overall process on target (Figure 7.6). This is largely due to the robustness and accuracy with which the model update algorithm adapts the equipment models to the new process states.

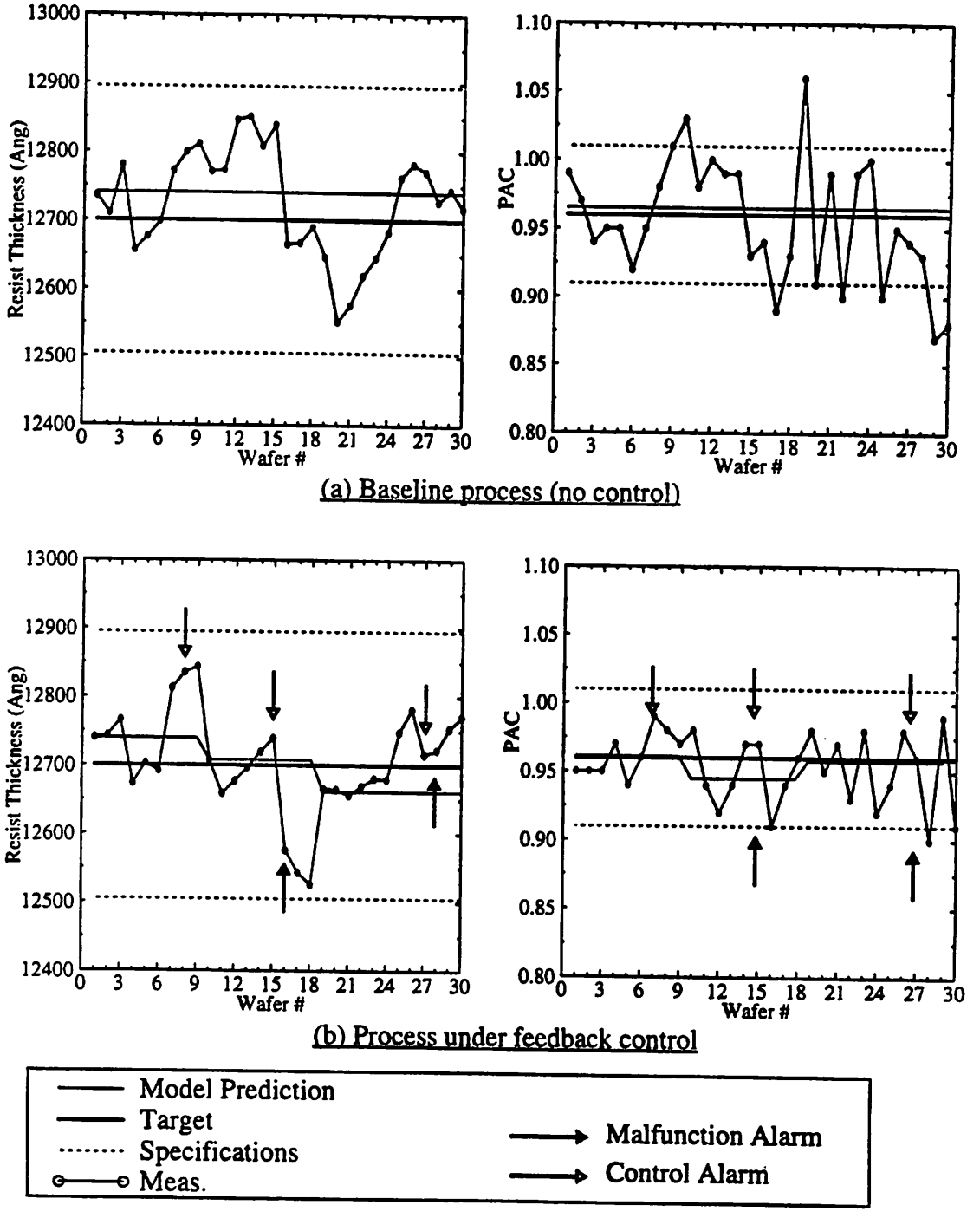


Figure 7.2 Wafer Track Outputs under (a) No Control, (b) Feedback Control

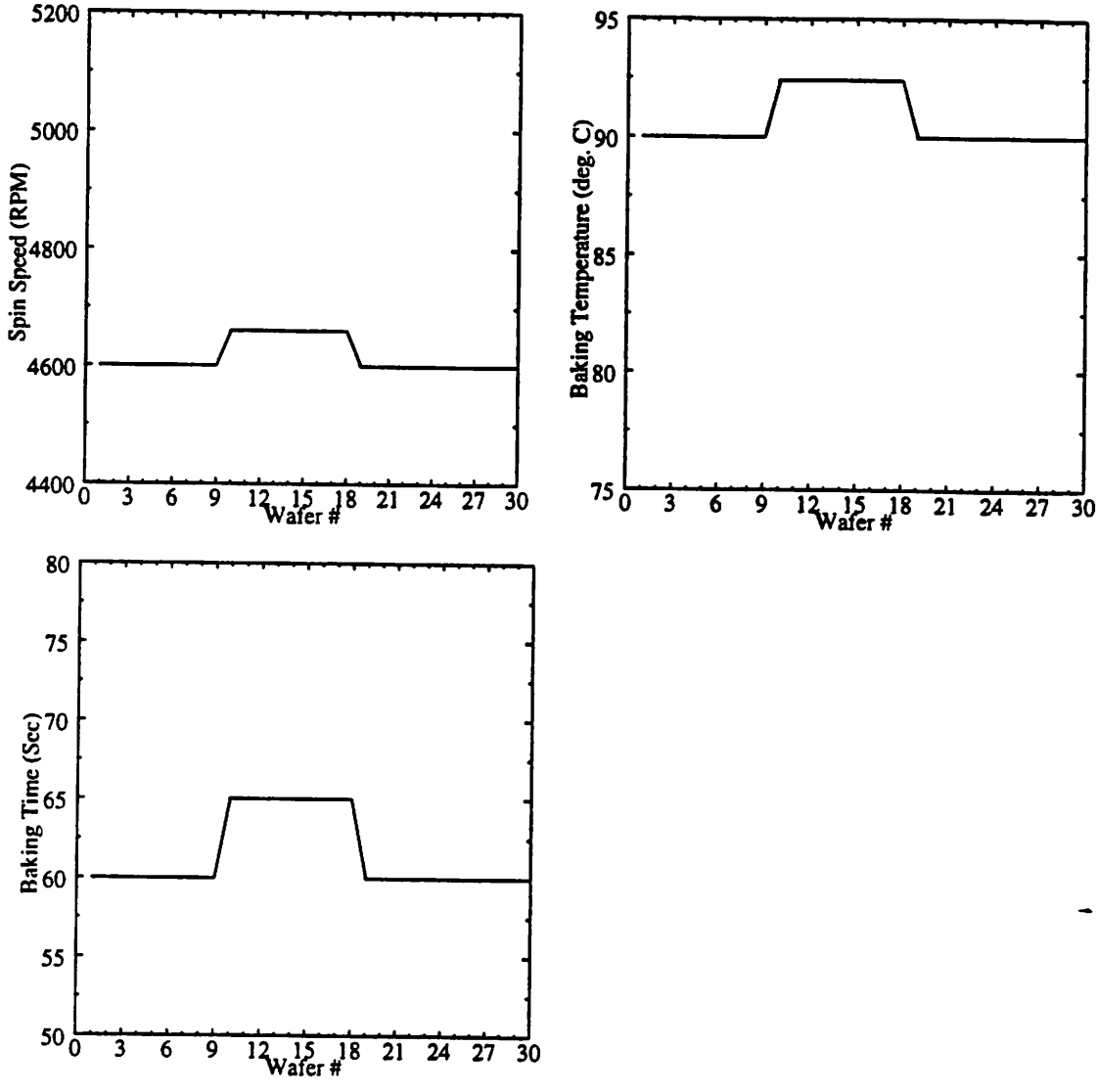


Figure 7.3 Wafer Track Recipe Changes under Feedback Control

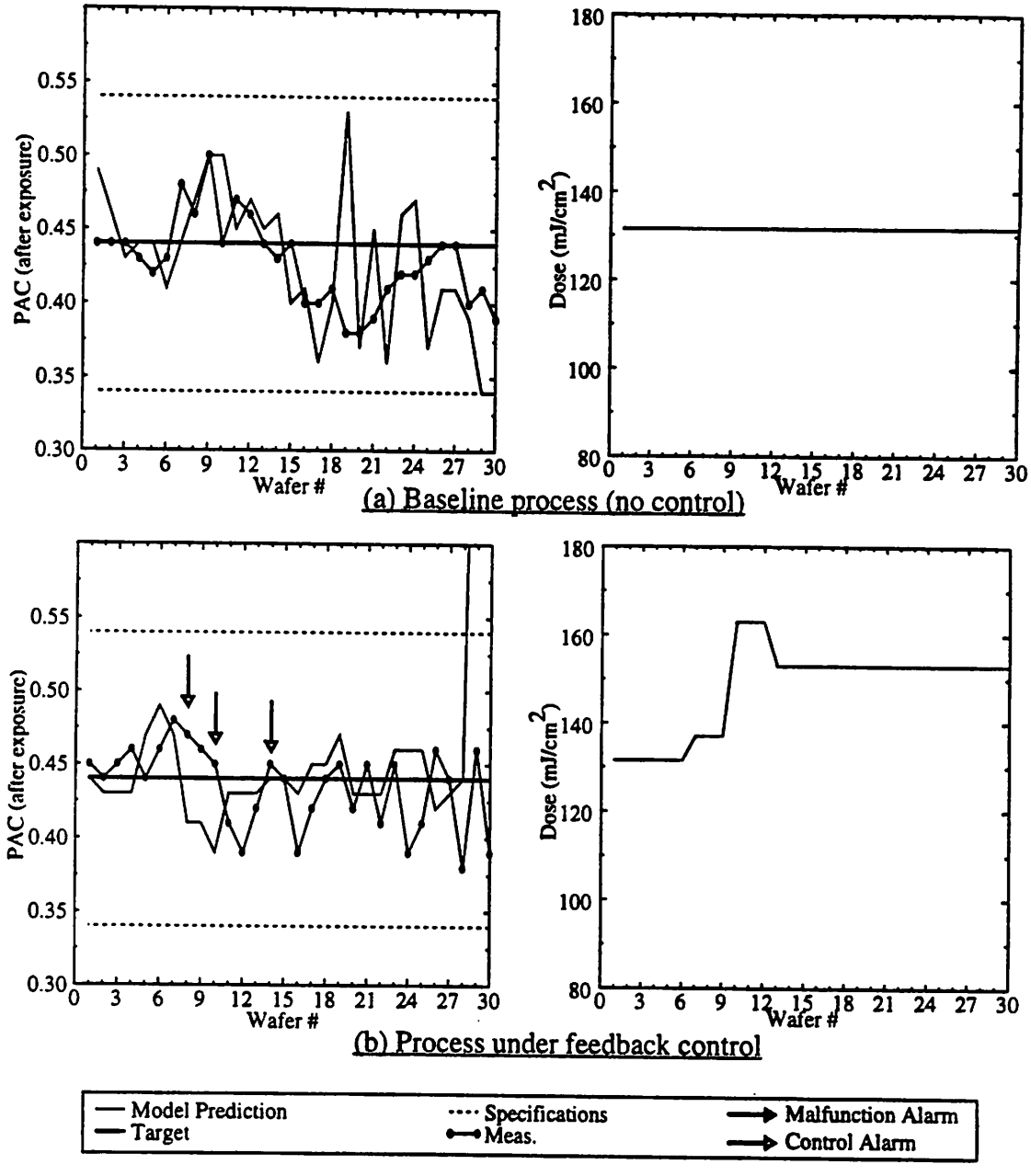
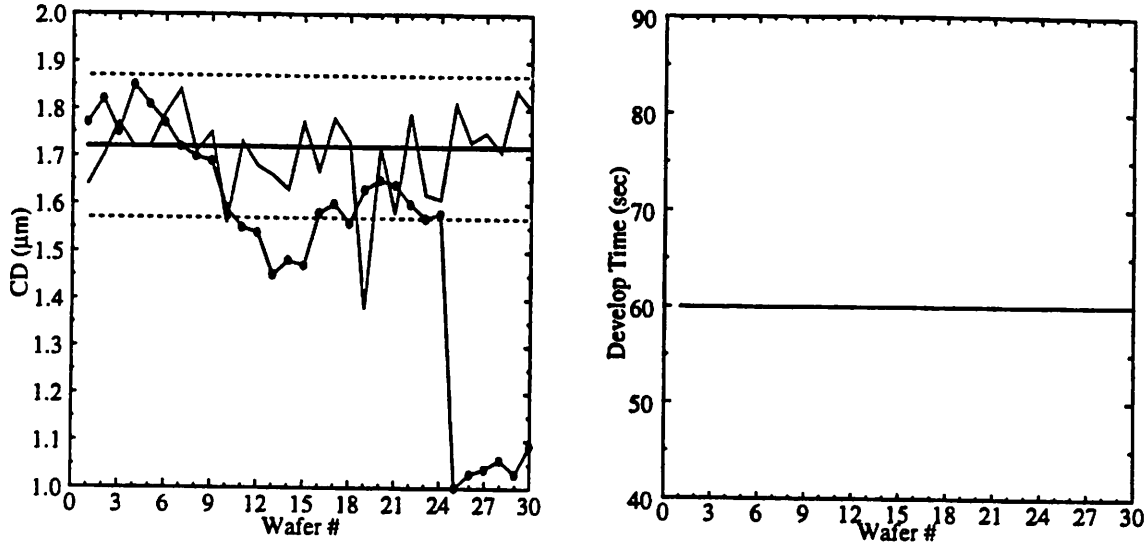
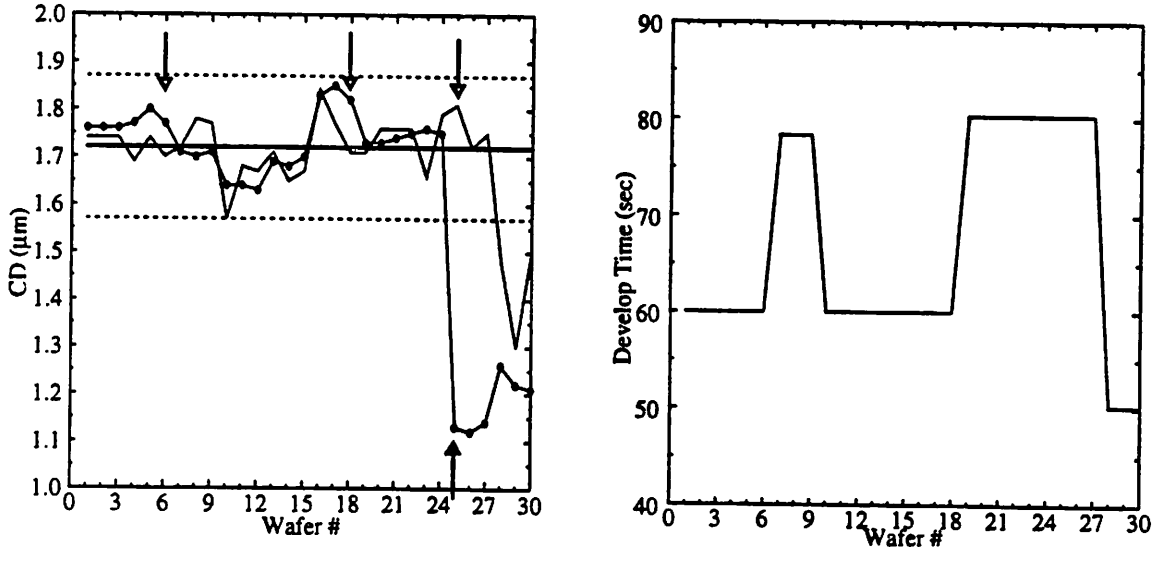


Figure 7.4 Stepper Output & Recipe Changes under (a) No Control, and (b) Feedback Control



(a) Baseline process (no control)



(b) Process under feedback control

Figure 7.5 Developer Output & Recipe Changes under (a) No Control, and (b) Feedback Control

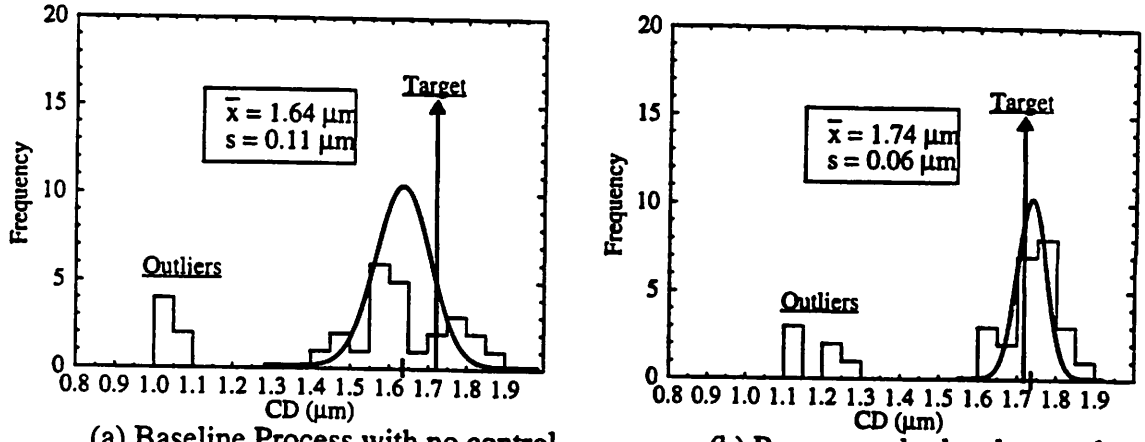


Figure 7.6 CD Distribution for (a) Uncontrolled Baseline Process Sequence, and (b) Process Sequence under Feedback Control

7.3.2 Simulation Results of Feed-Forward & Feedback Control

Next, we evaluate the capability of the process, when subject to both feedback and feed-forward control actions. Due to technical difficulties in the Microlab, we could not perform this experiment with actual wafers. Therefore, we have performed instead computer simulations of the process. In both baseline and controlled processes, we simulated the following drifting process:

	Wafer Track		Stepper	Developer
Wafer #	Thickness Drift	PAC Drift	PAC Drift	CD Drift
1 - 49	0	0	0	0
50 - 99	+4 Å/wafer	0	0	0
100 - 149	-2 Å/wafer	-0.001/wafer	0	0
150 - 199	-1 Å/wafer	0	+0.002/wafer	0
200 - 249	-3 Å/wafer	+0.0005/wafer	+0.0005/wafer	0

Table 7.1 Drift Settings of Simulated Experiment

We did not simulate the developer drifting because the feed-forward controller would be ineffective to correct for that, since the developer is the last machine in the sequence. We also did not simulate sudden shifts in process performances, because our controller was specifically designed to correct process shifts and has been demonstrated to work very well in such cases [11]. We simulated instead process drifts, because first, they pose a bigger challenge to our current version of the controller, which is not explicitly made to handle drifts (since time was not part of any model inputs), and because second, process drifts occur more often than process shifts from our experience in the laboratory. The figures below summarize the complete experiment.

We notice that a few glitches appear among the recipe changes. Upon their investigations, we have concluded that although the theories underlying the control system are sound, the robustness of some of its algorithms, such as the recipe generation algorithm, could be improved.

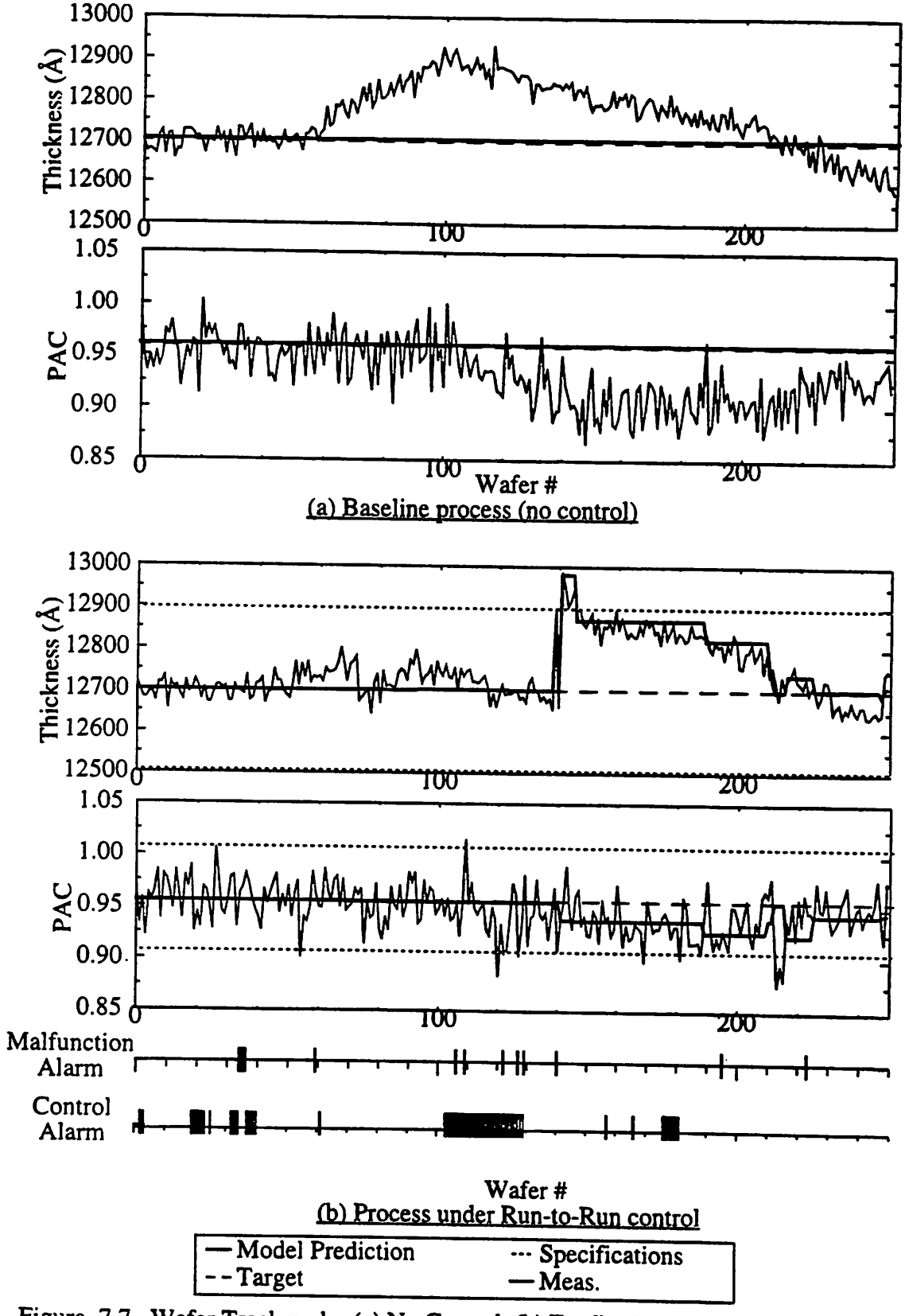


Figure 7.7 Wafer Track under (a) No Control, (b) Feedback & Feed-forward Control

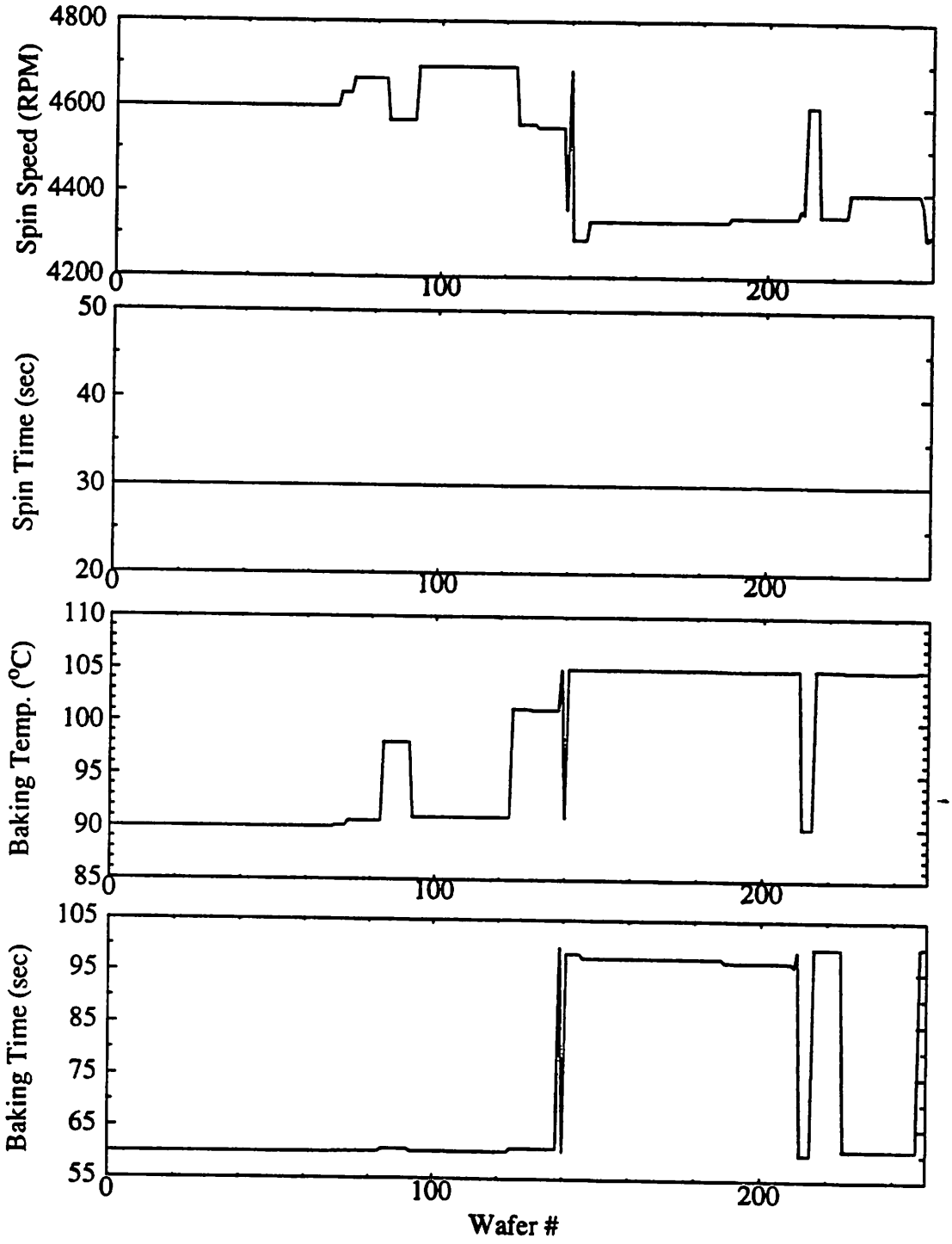
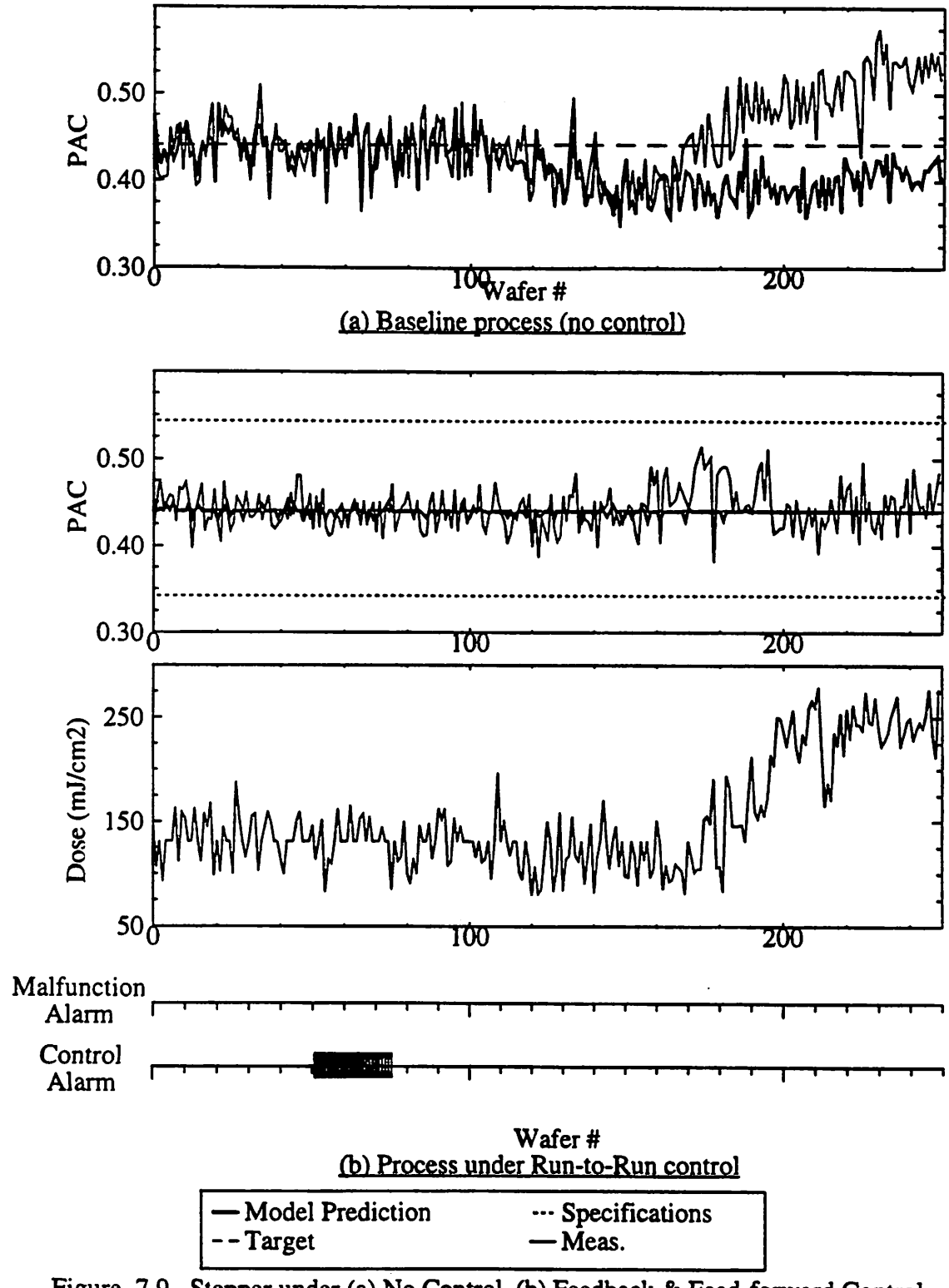


Figure 7.8 Wafer Track Recipe Changes under Feed-forward & Feedback Control



(a) Baseline process (no control)

(b) Process under Run-to-Run control

Figure 7.9 Stepper under (a) No Control, (b) Feedback & Feed-forward Control

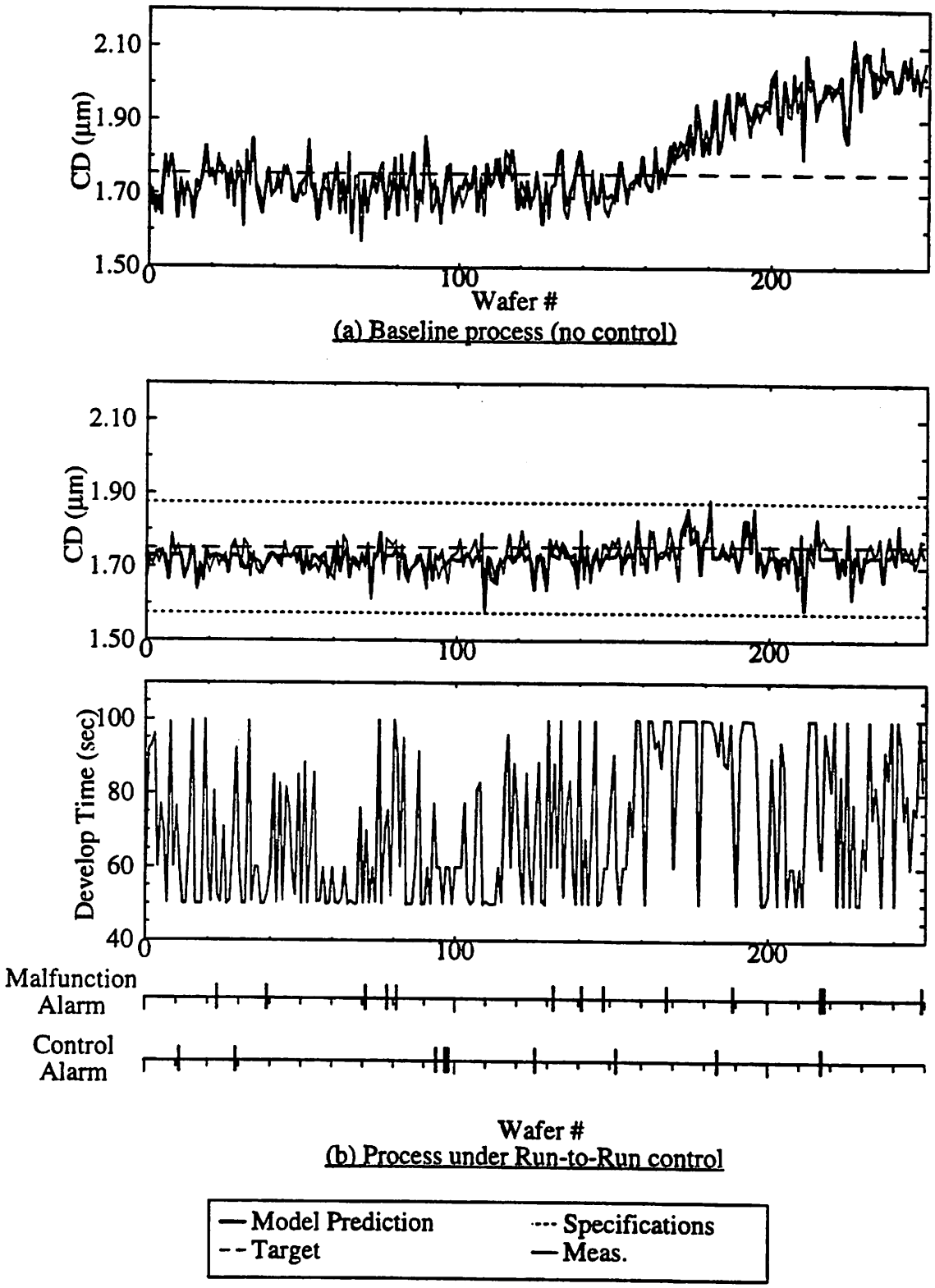


Figure 7.10 Developer under (a) No Control, (b) Feedback Control

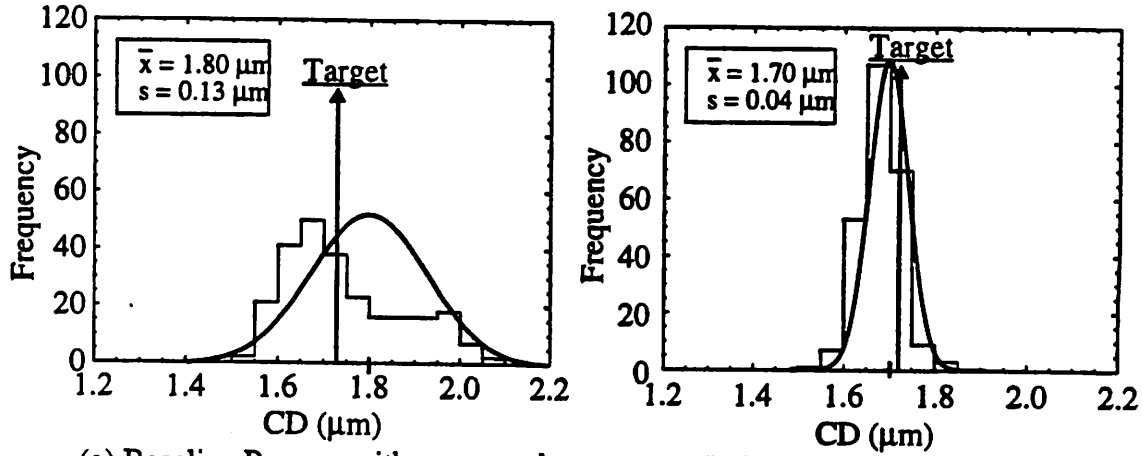


Figure 7.11 CD Distribution for (a) Uncontrolled Baseline Process Sequence, and (b) Process Sequence under Feed-forward and Feedback Control

7.3.3 Summary

The experimental results of the feedback and feed-forward controllers are very encouraging. The feedback controller is very efficient in detecting drifts and in bringing the process back on target, while the complementing feed-forward controller is very efficient in reducing process variations. The combination of both controllers has proven to be a powerful tool in increasing the process capability of the photolithography sequence.

7.4 Experimental Results of the Diagnostic System

7.4.1 Software Implementation of the Diagnostic System

The diagnostic system is activated only upon a control or malfunction alarm. The reason is because the sensitivity of any diagnostic system is generally limited: if diagnosis were performed on every single wafer, a great number of misdiagnosis would occur, causing people to lose trust in the diagnostic system. For that reason, we have decided to diagnose only suspicious wafers, which are currently defined as those that have generated either a control or a malfunction alarm.

Upon an alarm, the diagnostic system calculates the probability of all the faults and presents the result to the operator. The operator is then expected to confirm the problem and enter the real fault into the computer, after which the diagnostic system updates the file containing the conditional probability of faults for all combinations of evidence, and the frequency of all faults. If the *Miscellaneous Faults* category becomes significant, i.e the ratio of miscellaneous faults relative to the total number of faults exceeds a specified threshold, the system suggests the user to update the fault list. It is very important for the diagnostic system to be adaptive, since only time and experience can improve its effectiveness.

The initial estimates of conditional probabilities come from the FAULT database [55] and personal experience. They are not well tuned yet, for several reasons. First, not all the faults described by the diagnostic system are fields in the FAULT database. Some, such as relative humidity or N_2 pressure, are considered part of the regular process variations in the laboratory. Second, we only have information on one machine of each kind, and we have estimates of only one expert (me), besides the database. However, since the system is adaptive, it will ultimately converge onto the correct conditional probabilities, as more diagnosis cases get logged.

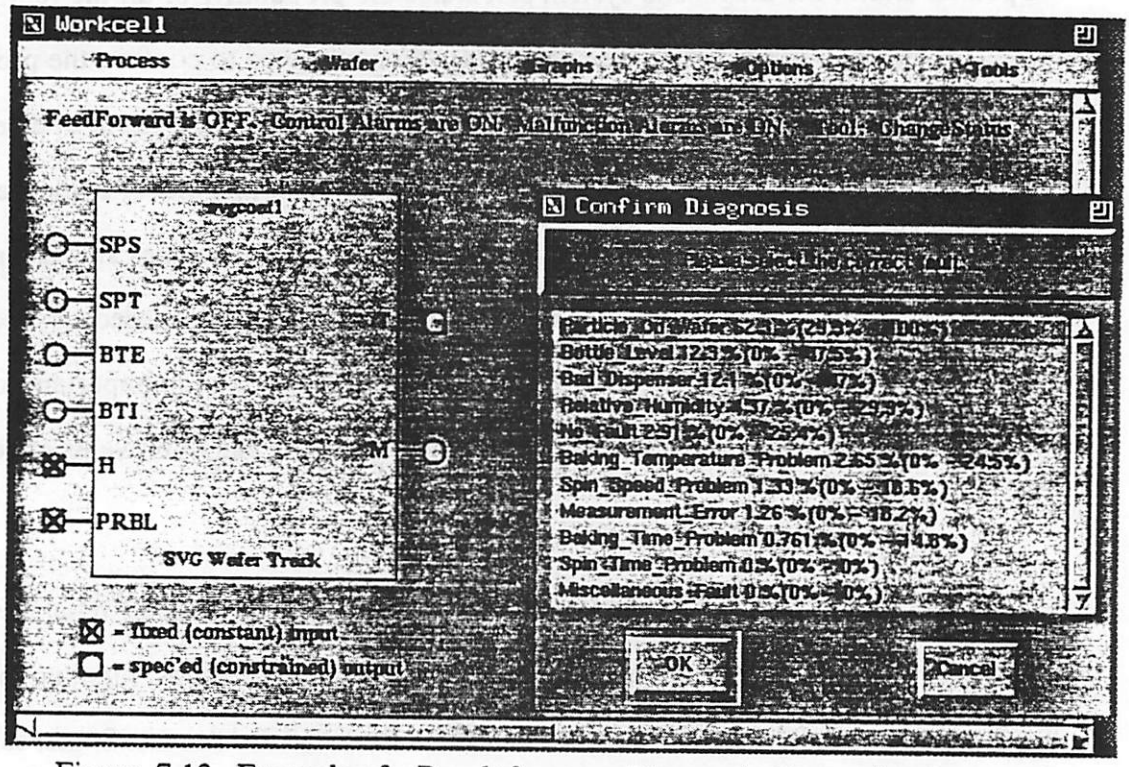


Figure 7.12 Example of a Result from our Diagnostic System Implementation

7.4.2 Some Diagnosis Examples

We present now four real cases of alarms that have been diagnosed by a technician. We will run those four examples through our diagnostic system to see if it would have helped the technician identify the real problem.

7.4.2.1 Diagnosis Example #1

In the first example, a control alarm with a type I and II errors of 5% and 20% was triggered on the stepper. The output PAC drifted up, and triggered the control alarm with a value of 0.43, while the system predicted a value of 0.32 instead. The input thickness and PAC were within specifications at 13115Å, and 0.96 respectively. The input dose was specified at the standard recipe of 167 mJ/cm^2. The environmental temperature was within tolerance. The ages of the lamp and filter optics were 67 days and 50 days respectively, while their characteristic lives were 45 days and 120 days respectively. The technician in

charge traced the problem to a weak lamp and replaced it with a new one. The diagnostic system estimated the fault probabilities as follows:

Table 7.2 Fault Probabilities of Diagnostic Example #1

Fault	Probability (%)	Probability Range (Confidence Level = 90%)
Wrong Input Thickness	3.19%	0.65% - 7.23%
Wrong Input PAC	3.73%	0.94% - 8.03%
Wrong Input Dose	9.06%	4.35% - 15.19%
Bad Lamp	**34.73%**	**26.14% - 44.21%**
Bad Environmental Temperature	1.29%	0% - 4.13%
Bad Lamp Strike	14.51%	8.49% - 21.83%
Damaged Filter Optics	0.22%	0% - 1.71%
Bad Shutter Timing Circuit	7.78%	3.44% - 13.55%
Bad Light Integrating Circuit	12.81%	7.15% - 19.80%
PAC Measurement Error	11.26%	5.97% - 17.92%
Miscellaneous Fault	1.60%	0% - 4.67%
No Fault	1.36%	0% - 4.26%

7.4.2.2 Diagnosis Example #2

In the second example, a malfunction alarm with a type I error of 5% was triggered on the wafer track. Streaks of photoresist were observed on the wafer. No other pattern was noted though. The inputs to the machine consisted of the normal recipe: a spin speed of 4600 RPM, a spin time of 30 seconds, a baking temperature of 90 °C, and a baking time of 60 seconds. The output thickness and PAC were 13658Å and 0.99 respectively, and their predicted values were 13092Å and 0.97 respectively. The reference wafer had been cleaned 1 day ago, and its characteristic life between cleanings is 7 days. The technician in charge traced the problem to bubbles in the photoresist dispenser tube, which was then cleaned. The diagnostic system estimated the fault probabilities as follows:

Table 7.3 Fault Probabilities of Diagnostic Example #2

Fault	Probability (%)	Probability Range (Confidence Level = 90%)
Wrong Input Spin Speed	7.24%	0% - 37.22%
Wrong Input Spin Time	0%	0% - 0%
Wrong Input Baking Temperature	3.14%	0% - 26.17%
Wrong Input Baking Time	13.49%	0% - 49.52%
Different Relative Humidity	2.38%	0% - 23.45%
Different Bottle Level	5.28%	0% - 32.42%
Dust Particle	6.70%	0% - 35.98%
Bad Photoresist Dispenser	**57.36%**	**24.29% - 100%**
Measurement Error	0.03%	0% - 0.05%
Miscellaneous Fault	0%	0% - 0%
No Fault	4.40%	0% - 29.94%

7.4.2.3 Diagnosis Example #3

In the third example, a control alarm was triggered on the wafer track. Its type I and type II errors were set at 5% and 20%, respectively. There was no discernible pattern nor any streak on the wafer. Actually, the laboratory users did not notice anything wrong with the machine at all. The control alarm was triggered however, because the thickness and PAC drifted to 13346Å and 0.99 respectively from their expected values of 13102Å and 0.97. The reference wafer was again just cleaned 2 days ago, and its characteristic life, i.e, period between cleaning, is 7 days. We suspect the cause to be relative humidity and the technician in charge agrees that relative humidity is the most probable cause. It was noted from the sensor log that the relative humidity had indeed changed from values around 25% to values around 50%. The diagnostic system's estimates for this alarm are:

Table 7.4 Fault Probabilities of Diagnostic Example #3

Fault	Probability (%)	Probability Range (Confidence Level = 90%)
Wrong Input Spin Speed	1.26%	0% - 18.24%
Wrong Input Spin Time	0%	0% - 0%
Wrong Input Baking Temperature	3.19%	0% - 26.33%
Wrong Input Baking Time	10.16%	0% - 43.38%
Different Relative Humidity	**63.76%**	**30.82% - 100%**
Different Bottle Level	15.07%	0% - 52.24%
Dust Particle	0%	0% - 0%
Bad Photoresist Dispenser	3.11%	0% - 26.09%
Measurement Error	0.11%	0% - 8.12%
Miscellaneous Fault	0%	0% - 0%
No Fault	3.34%	0% - 26.84%

7.4.2.4 Diagnosis Example #4

Finally, in the fourth example, a malfunction alarm occurred on the wafer track, on the first batch of the day. The type I and type II errors of the malfunction alarm were 5% and 20% respectively. The thickness and PAC were 12032Å and 0.94, which are significantly different from their expected values of 13112Å and 0.97. There was no discernible pattern, nor any streak on the wafer. The reference wafer was in need of cleaning, since it has been 10 days since its last cleaning, and its characteristic life is 7 days. The diagnostic system calculated the following fault probabilities:

Table 7.5 Fault Probabilities of Diagnostic Example #4

Fault	Probability (%)	Probability Range (Confidence Level = 90%)
Wrong Input Spin Speed	**32.54%**	**0% - 92.88%**
Wrong Input Spin Time	0%	0% - 0%

Table 7.5 Fault Probabilities of Diagnostic Example #4

Fault	Probability (%)	Probability Range (Confidence Level = 90%)
Wrong Input Baking Temperature	8.66%	0% - 51.92%
Wrong Input Baking Time	3.53%	0% - 36.31%
Different Relative Humidity	25.20%	0% - 82.42%
Different Bottle Level	1.12%	0% - 22.55%
Dust Particle	0.10%	0% - 0.15%
Bad Photoresist Dispenser	1.17%	0% - 23.04%
Measurement Error	8.32%	0% - 51.06%
Miscellaneous Fault	17.27%	0% - 69.66%
No Fault	2.10%	0% - 29.52%

When we checked the recipe of the machine, we confirmed that the diagnosis was indeed correct. Somebody changed the spin speed of the recipe and forgot to change it back to its default value.

7.4.3 Simulated Example of a False Diagnosis Converging to a Correct One

In section 6.9, we analyzed the rate of convergence of the conditional probabilities. To improve our understanding, we have run a simulated experiment to see how many diagnosis are necessary to make the system converge from an incorrect set of conditional probabilities to the correct one.

The experiment consists of using the same identical set of evidence on the diagnostic system 300 times, and recording the resulting fault probabilities. We have used the following evidence on the wafer stepper: a control alarm was triggered, under a type I error of 5% and type II error of 20%. The input thickness, PAC, and dose were 13115Å, 0.96, and 167mJ/cm^2 respectively. The expected PAC output was 0.312, but the actual output turned out to be 0.286. The environmental temperature of the chamber was within tolerance. The ages of the lamp and filter optics were 45 and 60 days respectively, and their characteristic

lives are 45 days and 120 days, respectively. This evidence fitted the following category of evidence best:

Table 7.6 Evidence Space

Evidence Category	Value
Possibly wrong input thickness	True
Possibly wrong input PAC	True
Possibly wrong input dose	True
Out-of-range environmental temperature	False
Output PAC	Below target
Alarm type	Control alarm
Lamp age	Old
Filter age	New

The original estimates of the fault probabilities for the fault space are listed in the table below, along with their ranges calculated at a 90% level of confidence:

Table 7.7 Original Estimates of Fault Probabilities

Fault Name	Fault Probability (%)	Fault Prob. Range (% - %)
Wrong input thickness	13.68	0 - 34.27
Wrong input PAC	18.24	0 - 40.69
Wrong input dose	31.91	12.68 - 57.50
Bad lamp	23.51	6.91 - 47.53
Bad environmental temperature	0	0 - 0
Bad lamp strike	2.45	0 - 14.04
Damaged filter optics	3.69	0 - 17.01
Bad shutter timing circuit	0.86	0 - 9.04
Bad light integrating circuit	0.86	0 - 9.04
PAC measurement error	3.55	0 - 16.67
Miscellaneous fault	0.51	0 - 7.33
No fault	0.74	0 - 8.57

Now, we have assumed that this database is being used on a "new" machine, with no pre-recorded history, that actually has a different set of problems. In other words, the fault conditional probabilities in the database are incorrect for this machine. Given the same set of evidence, we have assumed that the "true" fault probabilities of the new machine are instead:

Table 7.8 True Estimates of Fault Probabilities

Fault Name	Fault Probability (%)
Wrong input Thickness	2.00
Wrong input PAC	0.25
Wrong input Dose	7.00
Bad lamp	31.00
Bad environmental temperature	0.75
Bad lamp strike	18.00
Damaged filter optics	0
Bad shutter timing circuit	11.00
Bad light integrating circuit	13.00
PAC measurement error	15.00
Miscellaneous fault	1.00
No fault	1.00

To force these fault probabilities onto the diagnostic system, we have generated 300 random numbers between 0 and 1.0, which ultimately represent 300 diagnosis. We have related these numbers to the various types of faults by binning the value of the random number into categories, whose width is defined by the "true" probability of the fault. Therefore, when we run the diagnosis case described previously 300 times, we use these random numbers to simulate 300 faults with well defined probabilities. The database gets updated following each diagnosis, and the converging fault probabilities are shown in Fig-

ure 7.13. The precision of the probability estimates improve with experience. The following four figures show the ranges of each fault probability given a confidence level of 90%.

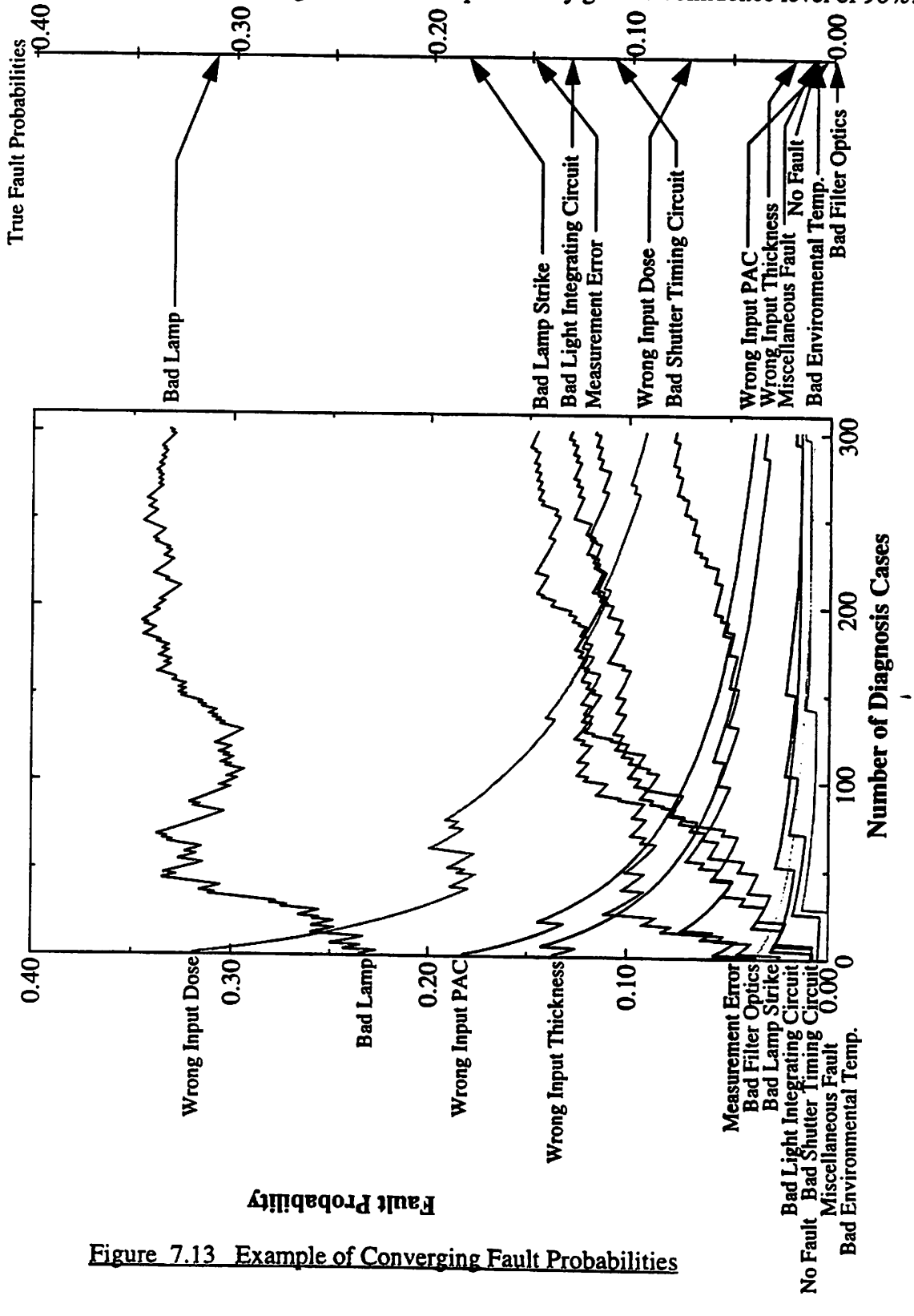


Figure 7.13 Example of Converging Fault Probabilities

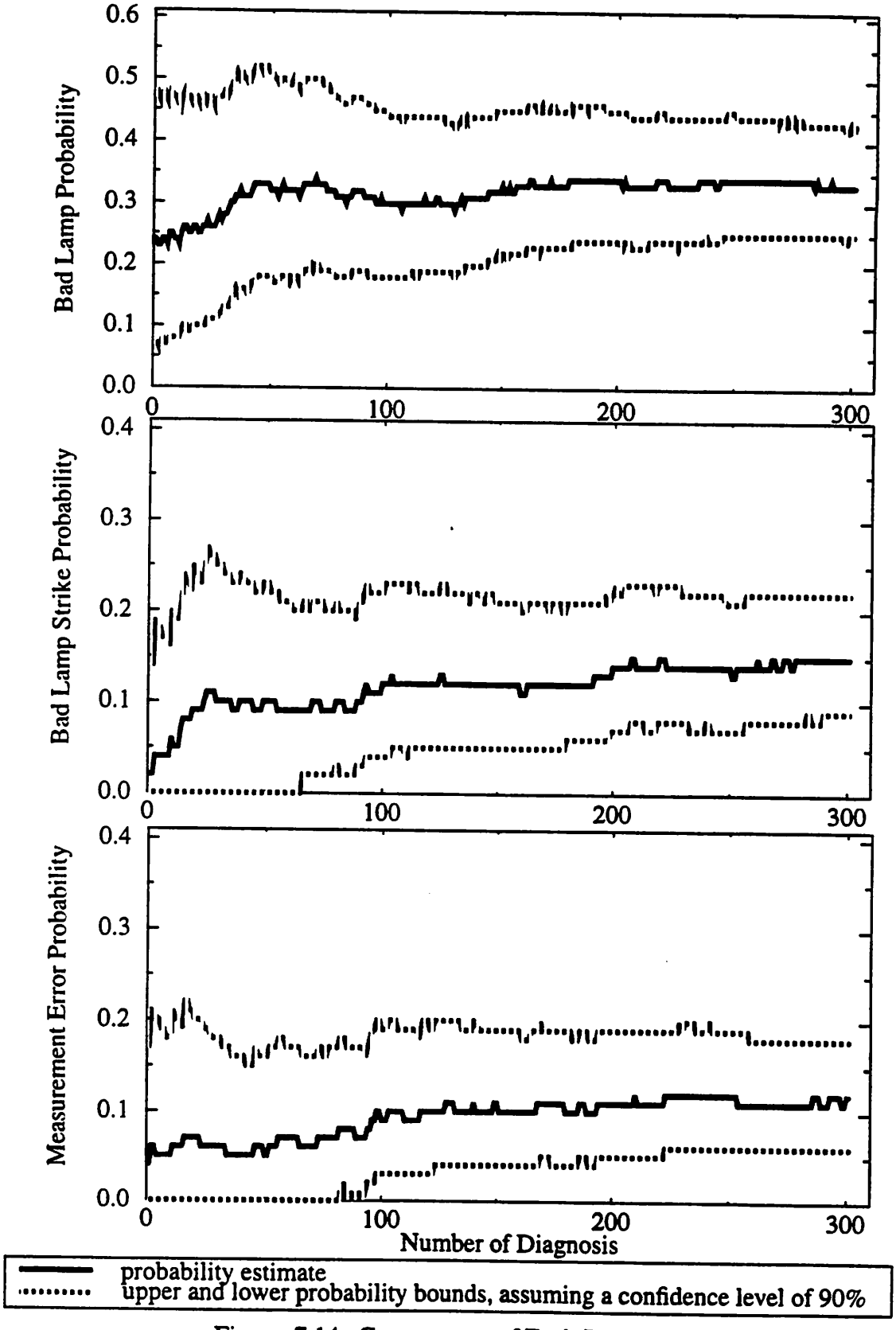


Figure 7.14 Convergence of Fault Probabilities

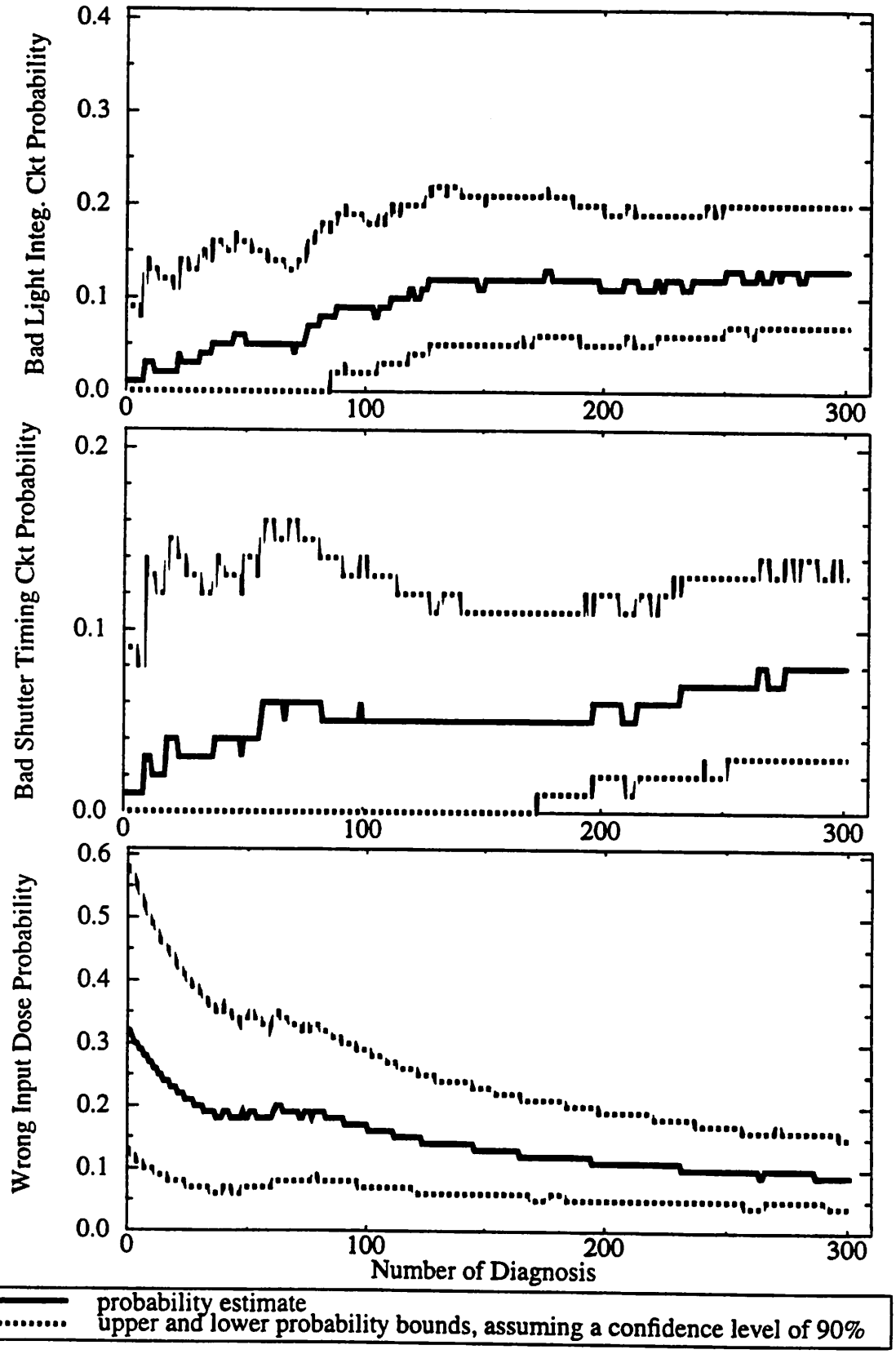


Figure 7.15 Convergence of Fault Probabilities (Continued)

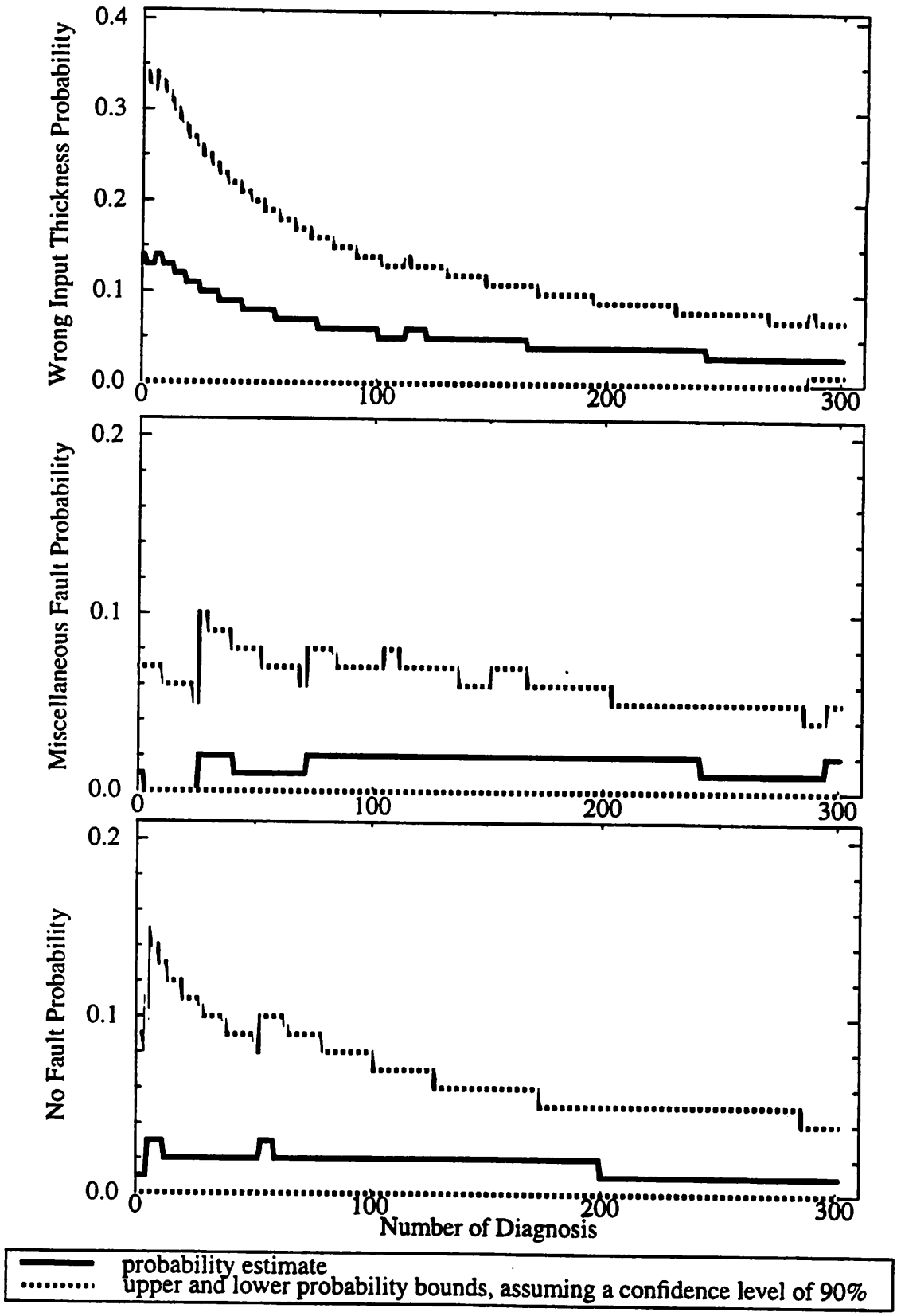


Figure 7.16 Convergence of Fault Probabilities (Continued)

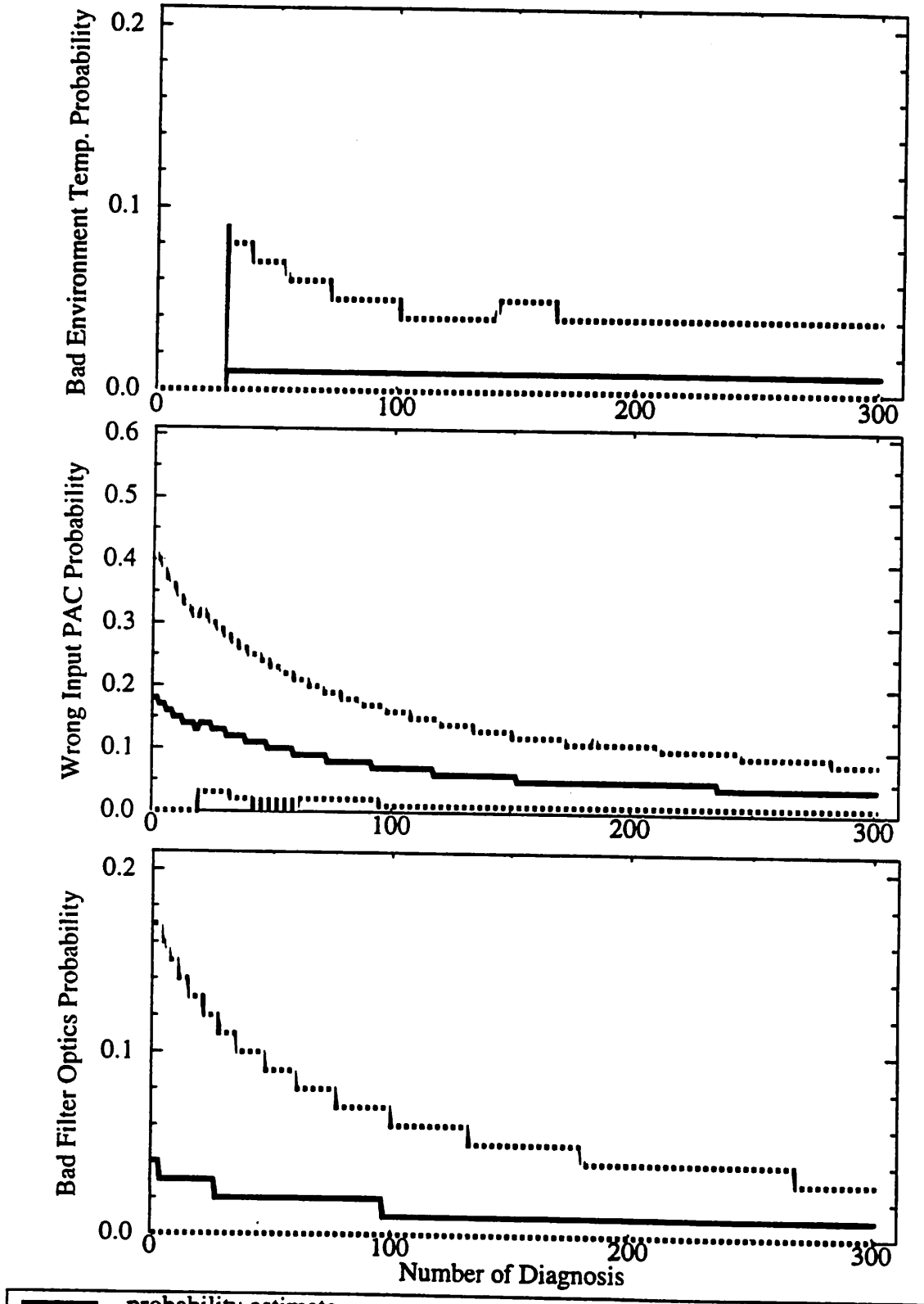


Figure 7.17 Convergence of Fault Probabilities (Continued)

7.5 Summary

In conclusion, we have tested both control and diagnostic systems on real and simulated equipment in the Microlab. The results are very promising. The combination of the feedback and feed-forward controllers proves itself very effective at keeping the targets of each machine within specifications, by detecting and correcting process drifts as they occur. Meanwhile, the diagnostic system has proven itself capable of homing into the correct problem, and of adapting itself from an incorrect set of conditional probabilities to a correct one.

Chapter 8 Conclusions

We have presented in this dissertation a methodology for developing a generic control and diagnostic system for a sequence of interrelated processes. The goal of the thesis is to provide an economical way of increasing the process capability of any sequence of interrelated process steps through innovative use of statistical techniques and probability theory.

The control system consists of a feedback loop and a feed-forward loop. The feedback loop tracks the performance of the present machine, using adaptive equipment models, and keeps the outputs of the machine centered around their respective target. It corrects process drifts by detecting them and by generating new recipes to counter any significant trends. After each process run, the feed-forward loop checks if standard settings on subsequent process steps would result in a correctly processed wafer. If the process outputs are predicted to be off-target, it will correct for the shortcomings of the present machine by generating customized recipes at the next process step. Together, the feedback and feed-forward loop have been proven to significantly improve the process capability of the photolithography sequence, resulting in photoresist patterns which are closer to target and have twice smaller variance.

The control mechanisms used in the control system are themselves not novel, but the way they are used is. We have purposefully chosen to use well known statistical techniques, instead of heuristics, to detect the process disturbances, and well known optimization techniques to generate recipes and update the equipment models. Thus, the resulting control methodology can be applied to any machine, and its accuracy can be properly quantified. If the equipment models were more complex, the control methodology would still be valid, although better optimization algorithms may be needed for the recipe generation and model updating algorithms.

In summary, the theory is clearly applicable to any sequence of interrelated process steps, and we have chosen to use the photolithography sequence purely as a demonstration vehicle. If future photolithography processes change from the one we have described in this thesis, some of the monitoring parameters will change, but the control methodology can still be applied to the new processes.

The control methodology can however be further improved, by fixing only the final output(s), instead of keeping every intermediary machine target fixed. In such a control methodology, the outputs of the intermediary processes would be dynamically adjusted, optimizing the final process output. Such a scheme has been investigated and has resulted in better process control. Another future direction for better process control is to actually model the time dependencies directly, so that process drifts can be corrected more accurately.

We have also implemented a diagnostic system to complement this control system. After each fault detection, the diagnostic system is activated to assist the operator in finding the cause of the decreased performance.

As in the case of the controller, the structure of the diagnostic system is also generic and can be applied on any machine. The diagnostic system is based on conventional probability theory, because its mathematical foundations are rigorous, and its assumptions are valid in our domain. The main novelty of our diagnostic system is that it incorporates both shallow and deep level information as evidence, so that any evidence can be used to diagnose faults. Typically, current diagnostic systems only handle one type of information (i.e, either shallow level or deep level only), which prevents them from gathering all necessary evidence in order to properly diagnose the fault. Furthermore, it also limits their diagnosis capabilities, since some faults can only be diagnosed from deep level information, whereas some others can only be diagnosed from shallow level information. Currently, we incorporate five sources of evidence: operator observations, sensor information, machine

maintenance data, process alarms, and equipment models. From this data, and from the conditional probabilities of faults initially supplied by machine experts (and subsequently updated by the system), the fault probabilities and their bounds are calculated, given a specified confidence level. The convergence of the fault probabilities has been derived in detail in the thesis, and the procedure for combining the estimates of conditional probabilities given by the machine experts has also been described in detail. We have implemented a software version of the diagnostic system, and tested it on real photolithography equipment malfunctions and drifts.

As in the control methodology, the methodology for combining the estimates of the conditional probabilities is not new, but comes from well known mathematical theories. We have purposefully chosen to use them, because they lead to a robust diagnostic system, that can be applied to any machine, and whose accuracy can be easily quantified.

Finally, although it is often successful in diagnosing the correct fault, the diagnostic system can use further inputs from machine experts' experiences. Other possible research directions to improve diagnosis include better fault signature filtering, a more efficient way of obtaining and managing the conditional probabilities of faults, and a better methodology for the machine to learn conditional probabilities [86].

[This page is intentionally blank]

APPENDIX 1 Program Documentation

We present now a brief documentation of all the files used in the diagnostic system.

- diagnosis.cc:

This is the main program of the diagnostic system. The inputs to this program consist of (1) the file "*machine_name*_fault_matrix", from which diagnosis.cc reads the fault conditional probabilities, the list of faults, and the list of evidence, and (2) internal outputs of the process controller, from which diagnosis.cc reads the type of alarm triggered, its type I and type II errors, measurement values, and predicted output values. For example, the input file to the stepper, named "gcaws_fault_matrix" looks as follows:

```
1.  gcaws 12 5 3

2.  Thickness_Problem 0
3.  PAC_Before_Exposure_Problem 0
4.  Dose_Problem 2
5.  Bad_Lamp 33
6.  Environment_Temperature 10
7.  Bad_Lamp_Strike 1
8.  Damaged_Filter_Optics 1
9.  Bad_Shutter_Timing_Circuit 3
10. Bad_Light_Integrating_Circuit 2
11. PACxp_Measurement_Error 9
12. Miscellaneous_Fault 2
13. No_Fault 3

14. Physical_observations 1
15. Chamber_Temp_Out_Of_Range 2
16. False True

17. Output_measurements 1
18. PACxp 2
19. Below_target Above_target

20. Alarm_type 1
21. Alarm 3 562 31 102 10 8
22. Malfunction Control False

23. Machine_component_age 2
```

24. Lamp_Age 2 45.0 40.0 3.5
25. New Old
26. Filter_Age 2 60.0 100.0 2.5
27. New Old

28. Equipment_model_measurement 3
29. thickness 2
30. Not_Fit Fit
31. PAC 2
32. Not_Fit Fit
33. Dose 2
34. Not_Fit Fit

35. 7 0.000 0.000 0.714 0.000 0.000 0.000 0.000 0.000 0.000 0.286 0.000 0.000
36. 9 0.000 0.000 0.710 0.000 0.000 0.000 0.000 0.000 0.000 0.290 0.000 0.000
37. 8 0.000 0.286 0.510 0.000 0.000 0.000 0.000 0.000 0.000 0.204 0.000 0.000
38. 3 0.125 0.167 0.292 0.134 0.134 0.000 0.134 0.000 0.000 0.000 0.000 0.000
39. etc..

Line #1 contains the *machine name*, the *number of faults* in the fault space, the *number of pieces of evidence*, and the *number of inputs to the machine*.

Lines #2 - #13 contain the name of a fault, and the number of times it has occurred.

Lines #14 - #34 contain the evidence data. There are five sets of pieces of evidence, as mentioned on line #1. The *name of each piece of evidence* is listed first, followed by the *number of variables* of the piece of evidence. Let n_{evi} be the number of variables, there are afterwards n_{evi} pairs of lines. The first of the two lines lists the *name of the variable* and its *number of values*, while the second line lists the *values* of the variable.

For the pieces of evidence related to the age of machine components and the type of alarm, there are additional data. For the evidence concerning the age of machine components, three numbers follow the number of values of the variable. The three numbers correspond to the life of the component, its characteristic life, and its shape factor (Please refer to §6.7.2 for details on these parameters). In the future, these numbers should be obtained directly from the maintenance database. For the evidence concerning the type of alarm, three numbers follow the number of variables. They correspond to the number of

processed wafers, the number of malfunction alarms, and the number of control alarms that have been triggered. The type I and II errors of the malfunction and control alarms are specified by the user through the controller software.

Each line from line #35 to the end of the file stores the number of occurrences of a particular combination of evidence (first number), followed by the conditional probability of each fault. The numbers are listed in the same order as the faults listed between line #2 - #13.

diagnosis.cc then calculates the probability of all combinations of evidence, the probability of all the faults and their range, given a specified confidence level, which is specified by the user interactively through the controller software.

- inputcp.cc:

This program asks the machine expert for his estimates of fault conditional probabilities. It goes through each combination of evidence and asks the user to enter the number of times s/he has seen that particular combination of evidence. Then it goes through the list of faults and asks the user how many times s/he estimates that particular fault was the cause. From that information, it calculates the fault conditional probabilities, and stores it in a file designated by the user.

- join_cp.cc:

This program joins all the conditional probability files from all the machine experts into one file, using the theory described by equation (6.49).

[This page is intentionally blank]

Chapter 9 References

[1] S.M. Sze, "VLSI Technology", 2nd Edition, McGraw Hill Book Co., 1988.

[2] Silicon Valley Group, Inc., 2240 Ringwood Ave., San Jose, CA 95131- 1716.

[3] GCA, 3111 Coronado, Santa Clara, CA 95054.

[4] SC Technology, Inc., 51 Whitney Place, Fremont, CA 94539.

[5] Sun Microsystems, Inc., 2550 Garcia Ave., Mountain View, CA 94043-1100.

[6] Nanometrics Inc., 690 East Arques Ave., Sunnyvale, CA 94086, 1990.

[7] Zhi-min Ling, Sovarong Leang, and Costas Spanos, "A Lithography Workcell Monitoring and Modeling Scheme", Micro-Electronique 1990, Leuven, Belgium, Sept. 1990.

[8] Zhi-min Ling, Sovarong Leang and Costas Spanos, "In-Line Supervisory Control in a Photolithographic Workcell", SPIE, Symposium on Microelectronic Processing Integration, Santa Clara, Sept. 1990.

[9] Sovarong Leang and Costas Spanos, "Application of a Supervisory Control to a Photolithography Sequence", Advanced Semiconductor Manufacturing Conference, Boston, Oct. 1992.

[10] Sovarong Leang and Costas Spanos, "Statistically Based Feedback Control of Photoresist Application", Advanced Semiconductor Manufacturing Conference, Boston, Oct. 1991.

[11] Sovarong Leang and Costas Spanos, "Application of Feed-Forward Control to a Lithography Stepper", International Semiconductor Manufacturing Science Symposium, San Francisco, June 1992.

[12] Sovarong Leang, "Supervisory Control System for a Photolithographic Workcell", M.S. Thesis, University of California at Berkeley, Memorandum No. UCB/ERL M92/

70, July 1992.

[13] Andrew Neureuther et al, "SAMPLE", University of California at Berkeley, Memorandum No. UCB/ERL M82, 1982.

[14] F. H. Dill, W. P. Hornberger, P. S. Hauge, and J. M. Shaw, "Characterization of Positive Photoresist", IEEE Transactions on Electron Devices, Vol. ED-22, No.7, July 1975.

[15] F.H. Dill et al, "Modeling Projection Printing of Positive Photoresist", IEEE trans. on Electron Devices, vol. ED-22, No. 7, p.456, July 1975.

[16] Edward D. Palik, "Handbook of Optical Constants of Solids", Vol I & II, Academic Press, Maryland, 1991.

[17] Max Born and Emil Wolf, "Principles of Optics - Electromagnetic Theory of Propagation, Interference and Diffraction of Light", 6th Edition, Pergamon Press, 1980.

[18] M.J.D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations.", Proceedings of Dundee Conference on Numerical Analysis, 1977.

[19] OCG Chemicals Company, Address.

[20] Shang-Yi Ma, "Experimental Verification of the Sequential Optimization Methodology", M.S. Thesis, University of California Memorandum No. UCB/ ERL M93, 1993.

[21] Douglas C. Montgomery, "Introduction to Statistical Quality Control", 2nd ed., New York: John Wiley & Sons, 1990.

[22] G.Box, W. Hunter & S. Hunter, "Statistics for Experimenters: an Introduction to Design, Data Analysis, and Model Building" 1st ed., New York, John Wiley & Sons, 1978.

[23] SAS Institute Inc., JMP Version 2.0.2, Box 8000, Cary, NC 27512, 1989.

[24] Richard Guldi, C.D. Jenkins, G.M. Damminga, T.A. Baum, and T.A. Foster, "Process Optimization Tweaking Tool (POTT) and its Application in Controlling Oxidation Thickness", IEEE trans. on Semiconductor Manufacturing, Vol.2, No. 2, pp. 54-59,

May 1989.

[25] Emanuel Sachs, R.S. Guo, S. Ha, and A. Hu, "Process Control System for VLSI Fabrication", IEEE Trans. on Semiconductor Manufacturing, Vol. 4, No. 2, pp. 134-144, May, 1990.

[26] ULTRAMAX, Version 4.1, by ULTRAMAX Corp., 1990.

[27] B.J. Mandel, "The Regression Control Chart", Journal of Quality Technology, Vol.1, No.1, pp. 1-9, Jan. 1969.

[28] Richard Harris, "A Primer of Multivariate Statistics", Academic Press, 1975.

[29] Sherry Lee, "A Strategy for Adaptive Regression Modeling of LPCVD Reactors", Special Issues in Semiconductor Manufacturing, pp.69-80, University of California, Berkeley / ERL M90/8, January 1990.

[30] Ronald Crosier, "Multivariate Generalizations of Cumulative Sum Quality-Control Schemes", Technometrics, Vol. 30, No. 3, Aug. 1988.

[31] William Woodall and Matoteng Ncube, "Multivariate CUSUM Quality-Control Procedures", Technometrics, Vol. 27, No. 3, Aug. 1985.

[32] Mario Perez-Wilson, "Machine/Process Capability Study", Advanced Systems Consultants P.O. Box 1176, Scottsdale, AZ 85252-1176, 1989.

[33] Crid Yu, "A Multivariate Exponentially Weighted Moving Average Control Scheme", Special Issues in Semiconductor Manufacturing III, University of California at Berkeley, UCB Memorandum No. UCB/ERL M92/84, 1992.

[34] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, "Numerical Recipes in C", 2nd Edition, Cambridge University Press, 1992.

[35] Bart Bombay, "The BCAM Control and Monitoring Environment", University of California at Berkeley, M.S. Thesis, Memorandum No. UCB/ERL M92/113, 1992.

[36] Gene H. Golub, Charles F. Van Loan, "Matrix Computations", 2nd Edition, John Hopkins University Press, Baltimore, 1989.

[37] Keung-Chi Ng and Bruce Abramson, "Uncertainty Management in Expert Systems", IEEE Expert, Vol. 5, No. 2, April 1990.

[38] D.A. Waterman, "A Guide to Expert Systems", Addison-Wesley, Reading, MA, 1986.

[39] J. Pearl, "Probabilistic Reasoning in Intelligent Systems", Morgan Kaufmann, Palo Alto, CA, 1988.

[40] P. Hart, "Directions for AI in the Eighties", Fairchild Technical Report No. 612, 1982.

[41] Y. Pan, "Qualitative Reasonings with Deep-Level Mechanism Models for Diagnoses of Dependent Failures", Ph.D dissertation, University of Illinois at Urbana-Champaign, CSL Report T-132, 1983.

[42] B.G. Buchanan and E.H. Shortliffe, "Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project", Addison-Wesley, Reading, MA, 1984.

[43] R.A. Miller, H.E. Pople, Jr., and J.D. Myers, "Internist-1: An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine", New England J. Medicine, Vol. 307, No. 8, 1982, pp. 468-476.

[44] Jeff Yung-Choa Pan and Jay M. Tenenbaum, "P.I.E.S: An Engineer's "Do-It-Yourself" Knowledge System for Interpretation of Parametric Test Data", Proceedings of the 5th National Conference on Artificial Intelligence, pp. 836-843, 1986.

[45] B. de Finetti, "Theory of Probability", John Wiley & Sons, New York, NY. 1974.

[46] Glenn Shafer, "A Mathematical Theory of Evidence", Princeton University Press, 1976.

[47] L.A. Zadeh, "Fuzzy Sets as a Basis for a Theory of Possibility", Fuzzy Sets and Systems, Vol. 1, No.1, pp. 3-28, 1978.

[48] M. A. Kramer, "Malfunction Diagnosis Using Quantitative Models with Non-Boolean Reasoning in Expert Systems", Journal of the American Institute of Chemical Engi-

neers, Vol. 33, No. 1, January 1987.

[49] B.N. Grosof, "Evidential Confirmation as Transformed Probability: On the Duality of Priors and Updates", in "Uncertainty in AI", L.N. Kanal and J.F. Lemmer, eds., Elsevier, New York, NY, pp. 137-152, 1986.

[50] D. E. Heckerman and E.J. Horvitz, "On the Expressiveness of Rule-Based Systems for Reasoning under Uncertainty", Proceedings of 6th National Conference on AI, Morgan Kaufmann, Palo Alto, CA, pp. 121-126, 1987.

[51] R.O. Duda, P.E. Hart, and N.L. Nilsson, "Subjective Bayesian Methods for a Rule-Based Inference System", Proceedings of National Computer Conference, Vol. 45, pp. 1075-1082, 1976.

[52] Norman H. Chang and Costas J. Spanos, "Continuous Equipment Diagnosis Using Evidence Integration: an LPCVD Application", IEEE Transactions on Semiconductor Manufacturing, Vol. 4, No. 1, February 1991.

[53] Gary Stephen May, "Automated Malfunction Diagnosis of Integrated Circuit Manufacturing Equipment", Ph.D dissertation, University of California at Berkeley, Memorandum No. UCB/ERL M91/33, April 1991.

[54] G.A. Barnard, "Control Charts and Stochastic Processes", Journal of the Royal Statistical Society, (B), Vol. 25, 1959.

[55] David Mudie and Norman Chang, "FAULTS: An Equipment Maintenance and Repair System Using a Relational Database", Proceedings of the 1990 IEEE/CHMT International Electronics Manufacturing Technology Symposium, October, 1990.

[56] R. P. Lippman, "An Introduction to Computing with Neural Nets", IEEE ASSP Magazine, April 1987.

[57] A. Rege and A. M. Agogino, "Knowledge-based Expert Systems for Manufacturing" ASME-PED Vol. 24, pp. 67-83, 1986.

[58] Pamela Tsai, "A Neural System for Calibrating a Lithography Wafer Stepper", M.S

Thesis, UC Berkeley, CA, June 1994.

[59] K.V. Mardia, J.T. Kent, and J.M. Bibby, "Multivariate Analysis", Academic Press, 1979.

[60] Warren Flack, David Soong, Alexis Bell, and Dennis Hess, "A Mathematical Model for Spin Coating of Polymer Resists", Journal of Applied Physics, Vol. 56, pp. 1199-1206, Aug. 1984.

[61] K.C. Hickman, S. M. Gaspar, K. P. Bishop, S. S. H. Naqvi, J. R. McNeil, G. D. Tipton, B. R. Stallard, and B. L. Draper, "Use of Diffracted Light from Latent Images to Improve Lithography Control", Journal of Vacuum Science & Technology, Vol. B10, p. 2259, June 1992.

[62] R. Azari et al, "Dynamic Statistical Process Control", SPIE vol.921, p.258, 1988

[63] W.G. Oldham et al, "A General Simulator for VLSI Lithography and Etching Process", IEEE trans. Electron Devices, vol. ED-27, p.717, 1979.

[64] Costas J. Spanos, Sovarong Leang, and Sherry F. Lee, " A Control and Diagnosis Scheme for Semiconductor Manufacturing", American Control Conference, Vol. 3, pp. 3008-3012, June 1993.

[65] Peter A. Morris, "Combining Expert Judgements: A Bayesian Approach", Management Science, Vol. 23, No. 7, pp. 679 - 693, March 1977.

[66] Robert F. Bordley, "A Multiplicative Formula for Aggregating Probability Assessments", Management Science, Vol. 28, No. 10, pp. 1137 - 1148, October 1982.

[67] Christian Genest and Mark J. Schervish, "Modeling Expert Judgments for Bayesian Updating", The Annals of Statistics, Vol. 13, No. 3, pp. 1198 - 1212, 1985.

[68] S.R. Dalal and W.J. Hall, "Approximating Priors by Mixtures of Natural Conjugate Priors", Journal Royal Statistical Society Series B, Vol. 45, No. 2, pp. 278 - 286, 1982.

[69] Patrick D.T. O'Connor, "Reliability and Quality Engineering", Encyclopedia of Physical Sciences and Technology, Vol. 12, Academic Press, Inc., pp. 128-146, 1987.

[70] Harry F. Martz, "Reliability Theory", Encyclopedia of Physical Sciences and Technology, Vol. 12, Academic Press, Inc., pp. 147-163, 1987.

[71] Max B. Mendel and Thomas B. Sheridan, "Filtering Information from Human Experts", IEEE Transactions on Systems, Man and Cybernetics, Vol. 36, No.1, pp. 6-16, January/February 1989.

[72] Robert Winkler, "The Concensus of Subjective Probability Distributions", Management Science, Vol. 15, No. 2, pp. B-61 - B-75, October 1968.

[73] Edmund Eisenberg and David Gale, "Concensus of Subjective Probabilities: The Pari-Mutuel Method", Annals of Mathematical Statistics, Vol. 30, 1959, pp. 165-168.

[74] Frederick Sanders, "On Subjective Probability Forecasting", Journal of Applied Meteorology, Vol. 2, No. 2, pp. 191-201, April 1963.

[75] Robert Winkler, "The Assessment of Prior Distributions in Bayesian Analysis", Journal of the American Statistical Association, Vol. 62, pp. 776-800, 1967.

[76] Howard Raiffa and Robert Schlaifer, Applied Statistical Decision Theory, Harvard University, Division of Research, Graduate School of Business Administration, Boston, 1961.

[77] Norman L. Johnson and Samuel Kotz, Continuous Univariate Distributions, Vol. 1 & 2, Houghton Mifflin Co., Boston, 1970.

[78] N. L. Johnson, "An approximation to the multinomial distribution: some properties and applications", Biometrika, Vol. 47, pp. 93-102, 1960.

[79] J. McGehee, J. Hebley, and J. Mahaffey, "The MMST Computer Integrated Manufacturing System Framework", IEEE Transaction on Semiconductor Manufacturing, Vol. 7, p.107, May 94.

[80] Michael Sullivan, Stephanie Watts Butler, Judith Hirsch, and C. Jason Wang, "A Control-to-Target Architecture for Process Control", IEEE Transaction on Semiconductor Manufacturing, Vol. 7, p.134, May 94.

[81] Stephanie Watts Butler and Jerry A. Stefani, "Supervisory Run-to-Run Control of Polysilicon Gate Etch Using In Situ Ellipsometry", IEEE Transaction on Semiconductor Manufacturing, Vol. 7, p.193, May 1994.

[82] A. Hu, E. Sachs. A/ Ingolfsson, and P. Langer, IEEE/SEMI Int. Semiconductor Manufacturing Science Symposium, pp. 73 - 78, 1992.

[83] Sik-Lam Wong, "Measurement of Refractive Index / Film Thickness by Ellipsometry", M.S Dissertation, University of California at Berkeley, 1973.

[84] Sherry Lee, "Semiconductor Equipment Analysis and Wafer State Prediction System Using Real Time Data", Ph.D thesis Dissertation, University of California at Berkeley, Memorandum No. UCB/ERL M94/104, 1994.

[85] S. Saxena, and A. Unruh, "Diagnosis of Semiconductor Manufacturing Equipment and Processes", IEEE Transaction on Semiconductor Manufacturing, Vol. 7, p.220, May 1994.

[86] Alice M. Agogino, Ming-Lei Tseng,and Punit Jain, "Integrating Neural Networks with Influence Diagrams for Power Plant Monitoring and Diagnostics", Neural Network Computing for the Electric Power Industry: Proceedings of the 1992 INNS Workshop (International Neural Network Society), Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, pp. 213-216.