SPECIAL ISSUES IN SEMICONDUCTOR MANUFACTURING

I

EECS 290W Class Project Reports, Fall 1989

Professor:

Costas J. Spanos

Students:

Eric D. Boskin, Yupin K. Fong, Tom Garfinkel,
Haifang Guo, Christopher J. Hegarty,
Timothy H. Hu, Sherry F. Lee, Tom L. Luan,
Gary S. May, Jaime Ramirez, Elyse Rosenbaum

Memorandum No. UCB/ERL M90/8

13 January 1990

SPECIAL ISSUES IN SEMICONDUCTOR MANUFACTURING

I

EECS 290W Class Project Reports, Fall 1989

Professor:

Costas J. Spanos

Students:

Eric D. Boskin, Yupin K. Fong, Tom Garfinkel,
Haifang Guo, Christopher J. Hegarty,
Timothy H. Hu, Sherry F. Lee, Tom L. Luan,
Gary S. May, Jaime Ramirez, Elyse Rosenbaum

Memorandum No. UCB/ERL M90/8

13 January 1990

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Preface

This document contains the final reports of the projects that were completed during the first run of EECS 290W ("Special Issues in Semiconductor Manufacturing") in the fall semester of 1989. In this semester we covered a wide area of basic manufacturing topics, including statistical process control, design of experiments, and circuit design for manufacturability. The diversity of these subjects is reflected in the projects that are included in this report.

The first seven projects focus on Statistical Process Control (SPC). The application of SPC in semiconductor manufacturing is meant to ensure that the parameters of the equipment, as well as the product, remain on target during long production runs. This is accomplished by the early identification of damaging deviations in critical preformance measures. From the many SPC schemes that are available, chapters 1 and 2 address the evaluation of Shewhart control charts with arbitrary "runs rules", i.e. rules that characterize normal production. Chapter 3 is a short study on establishing non-parametric rules of deviation, by teaching some of the abnormal production patterns to simulated neural nets. Chapter 4 describes the implementation of a simulator, written to evaluate the general characteristics of Cumulative Sum (CUSUM) charts.

The next SPC subject is the variable sampling interval (VSI) chart. This chart has good sensitivity, yet it is more economical than comparable control schemes, since it requires fewer measurements. This property makes the VSI chart a promising candidate for applications such as photolithography control, where measurements can be very expensive. The potential economical benefits and other characteristics of this chart are examined in chapters 5 and 6.

Modern sensor technologies, combined with the proliferation of hardware communication protocols on the factory floor, greatly facilitate the collection of multiple real-time diagnostic readings. These readings contain valuable information about the process, yet due to their strong cross-correlation, their interpretation is not straightforward. Chapter 7 focuses on the application of a multivariate control scheme for the reliable generation of alarms from cross-correlated data. This scheme was used for the analysis of real-time data collected from a plasma etcher. Also in the context of equipment control, chapter 8 describes the implementation of an adaptive regression strategy for modeling equipment that change over time. This strategy has been built around the concept of the regression control chart and it has been applied on sample data from a Low Pressure Chemical Vapor Deposition reactor.

The next two topics focus on the application of statistical experimental designs in semiconductor production. The objective is the generation of empirical models of products and production equipment. Chapter 9 describes the application of non-linear transformation techniques in the analysis of oxide reliability data from oxides that have been grown on off-axis silicon substrates. The subject discussed in chapter 10 is the design and implementation of a two-staged statistical experiment. This two staged experiment was employed for the extraction of empirical models of several critical performance measures of a plasma etcher.

Finally, chapter 11 is a circuit manufacturability study that analyzes and compares two EPROM designs. The manufacturability of each design is evaluated with the help of formal statistical techniques, that predict the spread of parametric performances under given variations of the fabrication process.

The analysis of some of these topics required the development of special routines written in C and Fortran, and also procedures developed within special statistical analysis packages such as BLSS and RS/1. Some of the experimental design topics involved the collection of sizable amounts of raw data. This information is not included in this document but it is available from C. Spanos.

I want to thank the students whose names appear in this report, and the others who, by contributing their helpful comments, made this course a valuable experience.

Costas J. Spanos

January, 1990

Table of Contents

Simulation of Shewhart Control Charts with Supplementary Runs Rules

Tom L. Luan

Abstract

This report describes a program for the evaluation of the performance of Shewhart control charts with supplementary runs rules. This program was implemented in Fortran.

1. Introduction

The Shewhart Control Charts are often used with supplementary runs rules to detect small shifts and trends. These supplementary rules increase the sensitivity of the Shewhart control charts, while reducing their Average Run Lengths (ARLs) Various runs rules have been postulated and practically used since the 1950s [1], and, among them, a particularly popular set is known as the "Western Electric Rules". In general, the runs rules may be stated as follows: An out-of-control signal is given if k of the last m standardized sample means fall in the interval (a, b), where $k \leq m$ and $a < b$.

Champ and Woodall [1] have suggested an algorithm in which a Markov chain approach is adopted to evaluate the ARL of the Shewhart control charts. Although they have only applied the method to Shewhart $\overline{X}$ control charts, the method is general and can be applied to other types of control charts such as the R and p charts.

The objective of this project is to test and implement this algorithm. I have picked up two relatively simple run rules to test this method. A FORTRAN program is written for this purpose. However, I would like to point out that, with some effort, this code can be generalized to all runs rules presented in Champ and Woodall's paper, as well as to other runs rules if people choose to define their own.

2. Methodology

2.1. Runs Rules and Probability Distributions

A combination C_{12} of the following two runs rules has been considered in this project report:

Rule1: $C_1 = T(1,1,-\infty,-3),\ T(1,1,3,\infty)$

Rule2: $C_2 = T(2,3,-3,-2),\ T(2,3,2,3)$

For example, Rule 2 signals an alarm if two of the last three samples fall into (-3, -2) or if two out of the last three fall into (2, 3). $C_{12} = C_1 \cup C_2$ is the combination of these two rules. There are 5 critical regions for the C_{12} runs rule: $R_1 = (-\infty,-3)$, $R_2 = (-3,-2)$, $R_3 = (-2,2)$, $R_4 = (2,3)$, and $R_5 = (3,\infty)$, as shown in Fig. 1. The probabilities in each region are represented by p_1, p_2, p_3, p_4 and p_5, respectively. If the mean μ_o shifts, p_i (i=1,...,5) changes accordingly. I have calculated the probability in each region as a function of the shift of μ_o, the results are listed in Table 1.

2.2. The Markov chain Representation

The states of the Markov chain indicate the status of the chart with respect to each runs rule. There is one absorbing state that corresponds to the out-of control signal. The 8 states required in the Markov chain representation for chart C_{12} are listed in Table 2. The first two coordinates of each vector representing a particular state correspond to the rule T(2,3,–3,–2) and the next two correspond to the rule T(2,3,2,3). For example, state 5, represented by (01 10), tells us that the past two observations are in the intervals (2, 3) and (-2, -3), respectively. If the next observation is in either of the two intervals, or in $(-\infty,-3)$ or $(3,\infty)$, according to the rule C_{12}, an out-of-control alarm will be signaled.

The Markov chain transition matrix P is defined as $P = [P_{i,j}]$, where $P_{i,j}$ is the probability of the transition of state i to state j. The set of required transient states can be determined iteratively. For a given initial state, one can determine the state resulting from each of the region possibly containing the first sample point. This process is repeated for each new transient state until no new states can be generated [1]. To illustrate this, let's consider state 5, represented by (01 10), as an example. If this state is

occupied and the next observation is in R_3=(–2,2), the resulting state is (00 01), which is state 6. If, instead, the observation is in any other region, the resulting state is the absorbing state. We can then do the same thing for state 6, the resulting new states are state 2 and state 1, represented by (10 00) and (00 00) respectively. This process can thus be repeated iteratively until no new states can be generated. In practice, state 1, represented by (00 00) is usually used as the initial state. The transition matrix P can be deduced from Table 2.

3. Implementation

First we define the run length probability vector:

$$L_h = [Pr(N_1 = h), ..., Pr(N_{s-1} = h)]^T \tag{1}$$

where N_i is the run length of the chart with initial state i. To calculate the run length probabilities, the following recursive numerical method is used [2]

$$L_1 = (I-Q)1$$

$$L_h = L_{h-1}Q, \quad h = 1,2,3, \cdots \tag{2}$$

where 1 is a column vector of 1s and Q is the matrix obtained by deleting the last row and column from the transition matrix P. This method of calculating the run-length probability gives simple recursive formulas to calculate the run length probabilities. For the control chart C_{12} considered in this report, the recursive formulas are:

$$Pr(N_1 = h) = p_3\, Pr(N_1 = h-1) + p_2\, Pr(N_2 = h-1) + p_4\, Pr(N_4 = h-1)$$

$$Pr(N_2 = h) = p_3\, Pr(N_3 = h-1) + p_4\, Pr(N_5 = h-1)$$

$$Pr(N_3 = h) = p_3\, Pr(N_1 = h-1) + p_4\, Pr(N_4 = h-1)$$

$$Pr(N_4 = h) = p_3\, Pr(N_6 = h-1) + p_2\, Pr(N_7 = h-1)$$

$$Pr(N_5 = h) = p_3\, Pr(N_6 = h-1)$$

$$Pr(N_6 = h) = p_3\, Pr(N_1 = h-1) + p_2\, Pr(N_2 = h-1)$$

$$Pr(N_7 = h) = p_3\, Pr(N_3 = h-1) \tag{3}$$

When the run length probabilities are calculated using these formulas, the average run length (ARL) can be calculated using the formula given by Woodall and Reynolds [3]:

$$ARL = E(N) \approx \sum_{h=1}^{n} Pr(N = h) + \lambda\, Pr(N=n)\left[\frac{n}{(1-\lambda)} + \frac{1}{(1-\lambda)^2}\right] \tag{4}$$

where

$$\lambda = \frac{[1-\sum_{h=1}^{n} Pr(N = h)]}{[1 - \sum_{h=1}^{n-1} Pr(N = h)]} \tag{5}$$

and N, n are the run length and the number of steps required to converge, respectively. For the control chart C_{12}, n = 10 is sufficient.

4. Results

4.1. ARL as a Function of Mean Shift

The results of ARL calculation as a function of the shift (d) of μ_o are plotted in Fig. 2, for the control chart C_{12}. And the numerical values are listed in Table 3. The result shows that the supplementary runs rule increases the sensitivity of the Shewhart control chart, especially at small shifts. The Supplementary runs rule also reduces the ARL at the target value $\mu = \mu_o$. Although any desired ARL at the

target value could be obtained by changing the control limits, however, the increased sensitivity, obtained by using supplementary runs rules to identify small shifts and trends, cannot be obtained by simply narrowing the control limits of the original Shewhart control chart.

4.2. Comparison with Champ and Woodall's Results

My Fortran Program calculation results are almost exactly the same as the results obtained by Champ and Woodall (compare with Table 1 of their paper), essentially at all shifts (d). Therefore, independent numerical implementations confirm the correctness of the Markov chain approach and the efficiency of the numerical approximation of Eq. (4) described above.

5. Example

The program calculates the run length probabilities and ARL for given probability in each regions R_i ($i = 1,...,5$). the following example is for the shift $d = 0.0$ and $d = 3.0$ cases.

```
argon% a.out
read in p1,p2,p3,p4,p5 and nr from prob_d?.dat
INPUT FILE NAME:
prob_d0.dat

********The Run Length Probabilities Are:***********
 2.70E-03  2.41E-02  2.41E-02  2.42E-02  4.56E-02  2.42E-02  4.56E-02
 3.61E-03  2.40E-02  3.09E-03  2.40E-02  2.30E-02  3.09E-03  2.30E-02
 4.48E-03  3.45E-03  3.96E-03  3.45E-03  2.95E-03  3.96E-03  2.95E-03
 4.42E-03  3.85E-03  4.35E-03  3.85E-03  3.78E-03  4.35E-03  3.78E-03
 4.39E-03  4.23E-03  4.30E-03  4.23E-03  4.15E-03  4.30E-03  4.15E-03
 4.37E-03  4.20E-03  4.28E-03  4.20E-03  4.11E-03  4.28E-03  4.11E-03
 4.35E-03  4.17E-03  4.26E-03  4.17E-03  4.08E-03  4.26E-03  4.08E-03
 4.33E-03  4.15E-03  4.24E-03  4.15E-03  4.06E-03  4.24E-03  4.06E-03
 4.31E-03  4.13E-03  4.22E-03  4.13E-03  4.05E-03  4.22E-03  4.05E-03
 4.29E-03  4.11E-03  4.20E-03  4.11E-03  4.03E-03  4.20E-03  4.03E-03
******** The Last Three ARLs Are: ******************
   8   2.250E+02
   9   2.250E+02
  10   2.250E+02
FORTRAN STOP
argon%
argon% a.out
read in p1,p2,p3,p4,p5 and nr from prob_d?.dat
INPUT FILE NAME:
prob_d15.dat

********The Run Length Probabilities Are:***********
 5.00E-01  5.00E-01  5.00E-01  8.41E-01  8.41E-01  8.41E-01  8.41E-01
 3.66E-01  3.66E-01  3.66E-01  1.34E-01  1.34E-01  7.94E-02  7.94E-02
 1.04E-01  1.04E-01  1.04E-01  1.26E-02  1.26E-02  5.82E-02  5.82E-02
 2.08E-02  2.08E-02  2.08E-02  9.23E-03  9.23E-03  1.65E-02  1.65E-02
 6.44E-03  6.44E-03  6.44E-03  2.61E-03  2.61E-03  3.29E-03  3.29E-03
 1.91E-03  1.91E-03  1.91E-03  5.23E-04  5.23E-04  1.02E-03  1.02E-03
 4.82E-04  4.82E-04  4.82E-04  1.62E-04  1.62E-04  3.04E-04  3.04E-04
 1.32E-04  1.32E-04  1.32E-04  4.82E-05  4.82E-05  7.65E-05  7.65E-05
 3.74E-05  3.74E-05  3.74E-05  1.21E-05  1.21E-05  2.09E-05  2.09E-05
 1.01E-05  1.01E-05  1.01E-05  3.32E-06  3.32E-06  5.93E-06  5.93E-06
******** The Last Three ARLs Are: ******************
   8   1.676E+00
   9   1.676E+00
  10   1.676E+00
FORTRAN STOP
argon%
```

6. Conclusions

The Markov chain approach proposed by Champ and Woodall [1] can be used to model supplementary runs rule used with Shewhart control chart. Independent numerical implementation of this method in this project confirm that supplementary runs rules cause the Shewhart chart to be more sensitive to small shifts than the original.

References

[1] Champ, C.W., and Woodall, W.H. "Exact Results for Shewhart Control Charts with Supplementary Runs Rules," Technometrics, Vol. 29, No. 4, 393-399, 1987.

[2] Brook, D., and Evans, D. A. "An Approach to the Probability Distribution of CUSUM Run Length," Bulletin in Applied Statistics, 5, 113-128, 1978.

[3] Woodall, W.H., and Rynolds, M.R. Jr., "A Discrete Markov Chain Representation of the Sequential Probability," Communications in Statistics, Vol. 2, No. 1, 27-44, 1983.

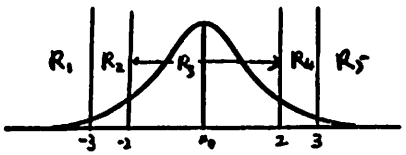


Figure 1a

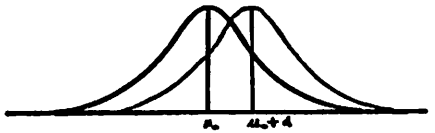


Figure 1b

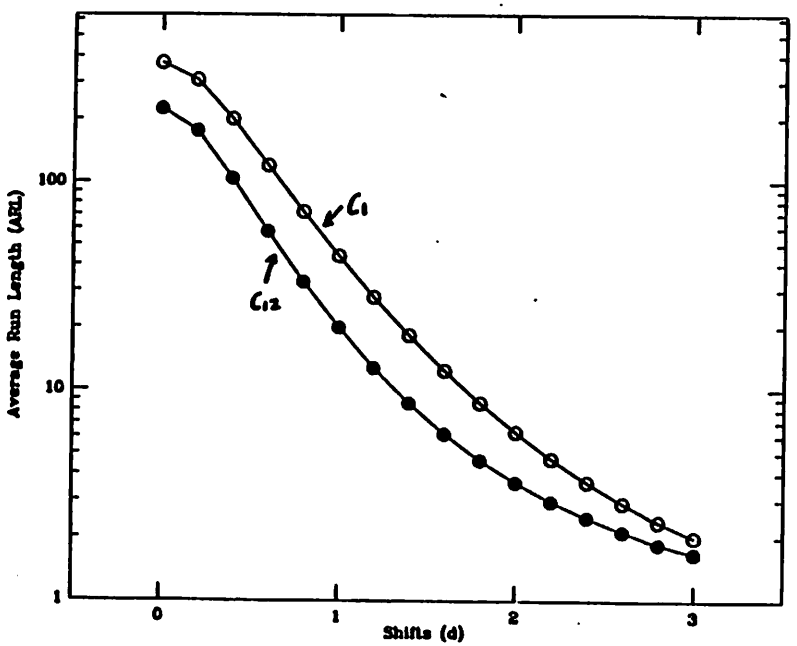


Figure 2

TABLE 1 SHIFTS AND PROBABILITY DISTRIBUTION

Shift d	Probabilities p1	p2	p3	p4	p5
0.0	.0013	.0214	.9544	.0214	.0013
0.2	.0007	.0132	.9502	.0333	.0026
0.4	.0003	.0079	.9370	.0501	.0047
0.6	.0002	.0045	.9145	.0726	.0082
0.8	.0001	.0025	.8823	.1012	.0139
1.0	.0000	.0013	.8400	.1359	.0228
1.2	.0000	.0007	.7874	.1760	.0359
1.4	.0000	.0003	.7254	.2195	.0548
1.6	.0000	.0002	.6552	.2638	.0808
1.8	.0000	.0001	.5792	.3056	.1151
2.0	.0000	.0000	.5000	.3413	.1587
2.2	0000	.0000	.4207	.3674	.2119
2.4	.0000	.0000	.3446	.3811	.2743
2.6	.0000	.0000	.2743	.3811	.3446
2.8	.0000	.0000	.2119	.3674	.4207
3.0	.0000	.0000	.1587	.3413	.5000

TABLE 2 MARCOV-CHAIN REPRESENTATION

	Present state	Next state				
No	representation	R1	R2	R3	R4	R5
1	(00 00)	8	2	1	4	8
2	(10 00)	8	8	3	5	8
3	(01 00)	8	8	1	4	8
4	(00 10)	8	7	6	8	8
5	(01 10)	8	8	6	8	8
6	(00 01)	8	2	1	8	8
7	(10 01)	8	8	3	8	8
8	absorbing	8	8	8	8	8

TABLE 3 CALCULATION OF ARL

shift d	ARL C1	C12
0.0	370.4	225.0
0.2	308.4	176.1
0.4	200.0	104.5
0.6	118.7	57.8
0.8	71.5	33.1
1.0	43.9	20.0
1.2	27.8	12.8
1.4	18.3	8.69
1.6	12.4	6.21
1.8	8.69	4.66
2.0	6.30	3.65
2.2	4.72	2.96
2.4	3.65	2.48
2.6	2.90	2.17
2.8	2.38	1.87
3.0	2.00	1.68

Average Run-Lengths of Shewhart Control Charts with Supplementary Runs Rules

Yupin K. Fong

Abstract

A C program which determines the average run-lengths (ARL) of Shewhart control charts with supplementary runs rules has been developed. By representing the supplementary runs rules with a Markov chain, the exact ARL can be easily determined [1]. This program reduces the number of initial states in the Markov chain (as compared to [1]) resulting in a significant saving of computing time for most supplementary runs rules. The ARL's calculated using Markov chains agrees with ARL's simulated using a normally distributed random number generator.

1. Introduction

Shewhart control charts with supplementary runs rules (SCC/SRR) not only determine if the measured parameter of a process is out-of-control by 3σ, they can also signal trends or shifts in the process. For example, if the last eight measurements are all larger than the expected value, this might signify a shift which requires further investigation. With each additional runs rule, the average number of measurements before a Type I [2] error (average run-length, ARL) will decrease. A Type I error is when a signal is generated due to the inherent randomness of the measured parameter and not due to a shift in the process. Thus the trade-off is between the number of trends or shifts that can be monitored versus the ARL.

Recently, Champ and Woodall [1] proposed a simple and efficient method to determine the ARL's of SCC/SRR using Markov chains. This method is significant because it allows the calculation of the ARL when many runs rules are used simultaneously. The implementation of Champ and Woodall's method in a C program is the subject of this project. Section II of this report will describe the use of Markov chains to determine the ARL's of SCC/SRR. Section III discusses the specifics of the C program including the algorithms used to 1) reduce the number of initial states in the Markov chain and 2) generate normally distributed random numbers. Finally, Section IV demonstrates the functionality of this ARL C program including the simulation of ARL's using a normally distributed random number generator. Insight into the factors which determine the ARL computation time are also discussed in this section.

2. Markov Chain Representation of SCC/SRR

The purpose of this section is to give a short description on what is a Markov chain representation of SCC/SRR and how the ARL's can be calculated from this Markov chain. A clear understanding of this topic is required to follow the specifics of the C program discussed in the next section. A more general and mathematical treatment of this problem can be found in [1,3].

Fig. 1a shows an example of a SCC/SRR. Each rule (k,m,a,b) consists of four parameters. If m of the last k measurements fall within a and b, then an out-of-control signal is generated. a and b are in terms of normalized σ's. Rules 1 and 2 describe the usual Shewhart 3σ control chart. Rules 3 and 4 are the supplementary runs rules. Rule 3 states that if 2 of the last 3 measurements are within (-3,-2) then a signal should occur. Similarly, Rule 4 is for 2 of the last 3 measurements being within (2,3). The state table for the Markov chain representation of this SCC/SRR is shown in Fig. 1b. Each row is a different state of the Markov chain while each column corresponds to a different region which the measurement can be within. In this example, the regions are $(-\infty,-3)$, (-3,-2), (-2,2), (2,3), and $(3,\infty)$ corresponding to all possible measurement values between $-\infty$ and $+\infty$. The entries in the table point to the next state for a given present state and a measurement within a particular region.

The initial state represented by (0/0/00/00) is the first state. Each section in (0/0/00/00) corresponds to one of the four rules, the first section corresponds to rule 1, the second corresponds to rule 2, and so on. For any k=m rule, the section will contain one digit representing how many consecutive measurements have been in (a,b) for that rule. For any k<>m rule, the section will contain m-1 ones and zeros representing the time evolution of the measurements and how they correspond to that

rule. For example, 10 represents that the last measurement was within (a,b) while the next to last measurement was not. Similarly, 01 represents that the next to last measurement was within (a,b) while the last measurement was not. Additional states are created when a new state representation is generated. State 0 is the "absorbing" [1] or signal state. For example, if a new measurement is within (–3,–2) and the present state is state 2, 4, 5, or 6, then the next state will be state 0 (a signal generating state) because 2 of the last 3 measurements were within (-3,-2).

The ARL is the expectation value of the number of measurements before an out-of-control *signal* is generated,

$$ARL = E(N) = \sum_{h=1}^{\infty} h \Pr(N = h), \tag{1}$$

where Pr(N = h) is the probability that the number of measurements is h. This probability can be determined recursively with the use of the state table (Fig. 1b). For h > 1,

$$\Pr(N_{S1} = h) = p_{R2} \Pr(N_{S2} = h - 1) + p_{R3} \Pr(N_{S1} = h - 1) + p_{R4} \Pr(N_{S3} = h - 1), \tag{2}$$

$$\Pr(N_{S2} = h) = p_{R3} \Pr(N_{S4} = h - 1) + p_{R4} \Pr(N_{S5} = h - 1), \tag{3}$$

and similarly for $\Pr(N_{S3} = h)$, $\Pr(N_{S4} = h)$, ..., where N_{S_i} is the run length of the chart with initial state i and p_{Rj} is the probability that the measurement is within region j. Note that $N = N_{S1}$ since the initial state is the first state. For h = 1,

$$\Pr(N_{S1} = 1) = 1 - p_{R2} - p_{R3} - p_{R4}. \tag{4}$$

$$\Pr(N_{S2} = 1) = 1 - p_{R3} - p_{R4}, \tag{5}$$

and similarly for $\Pr(N_{S3} = 1)$, $\Pr(N_{S4} = 1)$, ...

The ARL in Eq.(1) can be approximated for run-length probabilities which are geometrically limited [1] by,

$$ARL \simeq \sum_{h=1}^{n^*} h \Pr(N = h) + \frac{\lambda}{(1-\lambda)^2} \Pr(N = n^*) [n^*(1-\lambda) + 1], \tag{6}$$

where

$$\lambda = \frac{[1 - \sum_{h=1}^{n^*} \Pr(N = h)]}{[1 - \sum_{h=1}^{n^*-1} \Pr(N = h)]} \tag{7}$$

and n^* is expected to be less than 25 for most SCC/SRR. Fig. 1c shows how n^* affects the calculated ARL for the rules of Fig. 1a. The ARL converges to 225.4384 for $n^* = 9$.

3. Specifics of C Program for ARL Calculation

The method described in Section II to determine the ARL of SCC/SRR was implemented in C because C is the standard programming language for UNIX environments in addition to being easily portable to other computers. This section will discuss the specifics of this C program. The ARL program consists of four main sections as shown in Fig. 2. They are 1) Input processing, 2) Initialization of parameters, 3) Creation of states, and 4) ARL calculation.

Input processing is handled in the procedure "Get_rules" which scans the input deck (Fig. 3a) for key words. Runs rules are signify by the word 'rule' followed by the parameters k,m,a,b. 'nstar' is the number used for n^* in Eq. (6) and (7). 'shift' is the amount of shift (normalized to σ) of the normal distribution. 'combine' and 'print' are flags to combine certain states in the initial state table and to print out the final state table, respectively. 'random' is the number of ARL simulations used to determine the simulated ARL. 'end' is required for the last line of the input deck.

Initialization of parameters includes three procedures which do most of the bookkeeping of the program. "Regions" determines the different regions by calling a *pick sort* routine with all the values

of a's and b's. Pick sort is the fastest sorting algorithm for sorting less than 50 numbers (i.e., less than 25 rules). The probability for the measurement to be in each region is calculated using the C math library *erf* function. A shift in the normal distribution will affect this probability by shifting the limits of the regions. "In_rule" determines which rules correspond to each region while "State_rep" determines the length required for the state representation.

Creation of states includes both the generation of new states and also the user option of combining certain states. New states generation takes place in the procedures "Create_states" and "Next_state". The procedure "Combine_states" combines all states with identical rows to a single state resulting in a smaller state table.

The method to determine if a new state is generated follows that described in Section II. Given a present state and a new measurement within a particular region, "In_rule" is used to determine which rules correspond to this region. A check is first made to see if an out-of-control signal should be generated; otherwise, the state representation is updated. This new state representation is reduced, if possible, to a basic state representation which contains all the required information as the original new state representation. The basic state representation is then used to compare to previous state representations to avoid duplicate states.

State representation reduction is achieved by realizing that some of the information in the state representation can be disregarded. For example, a rule (5,6,a,b) with state representations of 10011, 10010, 10001, and 10000 are all equivalent because two measurements which were not within (a,b) have been made (the next to last measurement and the one before the next to last measurement). This state representation reduction greatly reduces the number of initial states in the Markov chain especially when the permutations of many rules are involved. Examples of this will be shown in the next section. This process of creating new states and determining the entries of the state table continues until no new states are generated.

Combining states with identical rows is not required to determine the ARL. It does reduce the size of the state table which reduces the amount of computing time needed to calculate the recursive run-length probabilities. However, this time saved might be offset by the time spent combining the states in the first place because of the extensive amount of bookkeeping needed. Entries in the state table will have to be updated to point to the the new combined states. Also whenever states are combined, the entire new state table has to be checked again to see if any identical rows were generated during this process. Again, examples of this will be shown in the next section including the fact that for runs rules which do not overlap (rules (k1,m1,1,3) and (k2,m2,2,3) do overlap), the state representation reduction algorithm generates a state table with no identical rows.

The calculation of the ARL takes place in the procedure "Find_ARL". "Prob_table" is first called to determine the p_{Rj} N_{S_i}'s of Eqs. (2)-(5). The ARL is then calculated for n^*, n^*-1, and n^*-2 using Eqs. (6) and (7). Comparing the ARL's can determine if a larger n^* is required for a more accurate ARL. The procedure "Random_ARL" determines the ARL using a normally distributed random number generator.

Normally distributed random numbers can be generated [4] using a uniform random number generator available in the C math library. Two random numbers, R_1 and R_2, which are between 0 and 1 (i.e., normalized to the maximum number that can be generated) are used to calculate $\theta = 2\pi R_1$ and $R = (-2 \ln(R_2))^{1/2}$. A pair of normally distributed random numbers $N_1 = R\cos(\theta)$ and $N_2 = R\sin(\theta)$ can then be generated. Any shift in the normal distribution is added to N_1 and N_2. The state table is used to determine when an out-of-control signal should be generated due to a sequence of these random numbers.

4. ARL Examples and Analysis

The SCC/SRR input deck of Fig. 3a is used to demonstrate the functionality of the ARL C program. Fig. 3b shows the output generated using this input deck. 29 states, not including the 0 state, are created. The calculated ARL is 166.05 while the simulated ARL is 166.82 showing good agreement between the two results. Fig. 3c is a table comparing the ARL of this SCC/SRR against the ARL of just the first two rules in Fig. 3a, i.e., a regular Shewhart control chart. The table shows that a shift in

the distribution is detected much more quickly for the SCC/SRR as expected. This advantage decreases for larger shifts because an out-of-control due to the regular Shewhart controls is more likely. Of course, the drawback of using SCC/SRR is the shorter ARL for a zero shift in the distribution.

The SCC/SRR in Fig. 3a is an example of runs rules which do not have overlapping regions. The state representation reduction algorithm generated a state table with no identical rows, i.e., a state table which do not have any states to combine. This is always true for SCC/SRR which do not have overlapping regions. Fig. 3d shows that 79 initial states are generated when this algorithm is not used. These 79 states are combined to form 39 states and then further combined to form the final 29 states. Remember even after the 29 states are determined, one last check had to be performed to make sure no new identical rows were generated. The ARL computing time is also shown in Fig. 3d. Combining identical states did not reduce the computation time as compared to the case when all 79 initial states were used. Note that these computation times do not include the time needed to calculate the simulated ARL.

The amount of computing time required to calculate the ARL depends on the number of rules in the SCC/SRR, the k of each rule, and if the rules have overlapping regions. More rules, larger k, and more overlapping regions all increase the number of initial states in the Markov chain resulting in an increase in the ARL computation time. By adding two rules to the SCC/SRR of Fig. 3a. the computing time for the SCC/SRR of Fig. 4a increases from 57 msec to 260 msec (Fig. 4b). If the state reduction algorithm is used, the number of initial states increases to 95; otherwise there will be 845 initial states and an ARL computation time of 12,803 msec.

5. Conclusions

A C program which determines the ARL of SCC/SRR using a Markov chain has been developed. This program includes an algorithm which reduces the number of initial states in the Markov chain thus significantly reducing the ARL computation time for most SCC/SRR. The ARL's calculated using this program agrees well with ARL's simulated using a normally distributed random number generator.

References

[1] C.W. Champ and W.H. Woodall, "Exact Results for Shewhart Control Charts with Supplementary Runs Rules," Technometrics, Vol. 29, No. 4, pp. 393-399, 1987.

[2] D.C. Montgomery, "Introduction to Statistical Quality Control," John Wiley & Sons, New York, 1985.

[3] W.H. Woodall and M.R. Rynolds, Jr., "A Discrete Markov Chain Representation of the Sequential Probability," Communications in Statistics, Vol. 2, No. 1, pp. 27-44, 1983.

[4] W.H. Press et al., "Numerical Recipes in C," Cambridge University Press, Cambridge, 1988.

(a) rule 1 (1,1,-∞,-3)
rule 2 (1,1,3,+∞)
rule 3 (2,3,-3,-2)
rule 4 (2,3,2,3)

(b)

		R1	R2	R3	R4	R5
1	0/0/00/00	0	2	1	3	0
2	0/0/10/00	0	0	4	5	0
3	0/0/00/10	0	6	7	0	0
4	0/0/01/00	0	0	1	3	0
5	0/0/01/10	0	0	7	0	0
6	0/0/10/01	0	0	4	0	0
7	0/0/00/01	0	2	1	0	0

(c) nstar=12: ARL=2.254384e+02
nstar=11: ARL=2.254384e+02
nstar=10: ARL=2.254384e+02
nstar= 9: ARL=2.254384e+02
nstar= 8: ARL=2.254382e+02
nstar= 7: ARL=2.254368e+02
nstar= 6: ARL=2.254527e+02
nstar= 5: ARL=2.255061e+02
nstar= 4: ARL=2.246600e+02
nstar= 3: ARL=2.228728e+02
nstar= 2: ARL=2.766332e+02
nstar= 1: ARL=3.703983e+02

Figure 1: Sample runs rules (a), states (b) and ARL as a function of nstar (c)

I) Input processing
  a) Get_rules

II) Initialization of parameters
  a) Regions
  b) In_rule
  c) State_rep

III) Creation of states
  a) Create_states
  b) Next_state
  c) Combine_states

IV) ARL calculation
  a) Find_ARL
  b) Prob_table
  c) Random_ARL

Figure 2: The structure of the ARL program

(a)
```
rule 1 1 -100 -3      * -100 and 100 were used instead of -∞ and +∞
rule 1 1   3  100
rule 4 5  -3   -1
rule 4 5   1    3
nstar 20
shift 0.0
combine
print
random 1000
end
```

(b)

1	0/0/0000/0000	0	2	1	3	0	16	0/0/0000/1010	0	7	8	25	0
2	0/0/1000/0000	0	4	5	6	0	17	0/0/1000/0110	0	4	5	26	0
3	0/0/0000/1000	0	7	8	9	0	18	0/0/0000/0110	0	2	1	27	0
4	0/0/1100/0000	0	10	11	12	0	19	0/0/0000/1110	0	28	29	0	0
5	0/0/0100/0000	0	13	1	3	0	20	0/0/0111/0000	0	0	1	3	0
6	0/0/0100/1000	0	14	8	9	0	21	0/0/0111/1000	0	0	8	9	0
7	0/0/1000/0100	0	4	5	15	0	22	0/0/1011/0000	0	0	5	6	0
8	0/0/0000/0100	0	2	1	16	0	23	0/0/1011/0100	0	0	5	15	0
9	0/0/0000/1100	0	17	18	19	0	24	0/0/1101/0000	0	0	11	12	0
10	0/0/1110/0000	0	0	20	21	0	25	0/0/0000/1101	0	17	18	0	0
11	0/0/0110/0000	0	22	1	3	0	26	0/0/0100/1011	0	14	8	0	0
12	0/0/0110/1000	0	23	8	9	0	27	0/0/0000/1011	0	7	8	0	0
13	0/0/1010/0000	0	24	5	6	0	28	0/0/1000/0111	0	4	5	0	0
14	0/0/1010/0100	0	24	5	15	0	29	0/0/0000/0111	0	2	1	0	0
15	0/0/0100/1010	0	14	8	25	0							

nstar=20: ARL=1.660545e+02 random ARL=1.668210e+02
nstar=19: ARL=1.660545e+02
nstar=18: ARL=1.660545e+02

(c)

	Rules 1,2		Rules 1,2,3,4	
shift	ARL	SIM ARL	ARL	SIM ARL
0.0	370.40	362.20	166.05	166.82
0.4	200.08	189.80	63.88	65.67
0.8	71.55	74.09	19.78	20.22
1.2	27.82	28.30	8.84	8.64
1.6	12.38	12.39	5.24	5.36
2.0	6.30	6.29	3.68	3.72
2.4	3.65	3.55	2.78	2.77
2.8	2.38	2.35	2.14	2.12

(d) (I) with state representation reduction algorithm
(II) without algorithm, but combine states
(III) without algorithm, do not combine states

	(I)	(II)	(III)
states	29	79 to 39 to 29	79
computation time (DECstation 3100)	57 msec	153 msec	148 msec

Figure 3: Simple Chart specification (a), states (b), performance (c) and simulation cost (d).

(a)
```
rule 1 1 -100 -3
rule 1 1  3  100
rule 4 5 -3  -1
rule 4 5  1   3
rule 5 6 -1   0
rule 5 6  0   1
nstar 20
shift 0.0
combine
end
```

(b) (I) with state representation reduction algorithm
(II) without algorithm, but combine states
(III) without algorithm, do not combine states

	(I)	(II)	(III)
states	95	845 to 253 to 121 to 95	845
computation time (DECstation 3100)	260 msec	13,380 msec	12,803 msec

Figure 4: Complex Chart specification (a) and simulation cost (b).

Using Neural Nets to Reconize Non-random Patterns in Control Charts

Timothy H. Hu

Abstract

Often, statistical process control depends on the recognition of special non-random patterns in the data. Since it is not practical to have a trained statistician to inspect all charts for non-randomness, it has been proposed to apply an automated pattern recognition procedure to this task. In this report, a simulated "neural net" that can be trained to identity special non-random patterns is used to recognize shifts and trends in a noisy univariate control chart. The method is compared to the Shewhart Control Chart with Western Electric Rules.

1. Introduction

Often, statistical process control depends on the recognition of special non-random patterns in the data. An example is shown in fig.1 where the process is well within the control limits but clearly there exists a cyclic pattern from data 10 to data 50. Since it is not practical to have a trained statistician to inspect all charts for non-randomness, it has been proposed to apply an automated pattern recognition procedure to this task.

The Shewhart Control Chart has been used by the industry for a long time together with the Western Electric Rules to detect non-random patterns on the control chart. The rule concluding that the process is out-of-control if either;

1. One point plots outside the 3-sigma control limits.
2. Two out of three consecutive points plot beyond the 2-sigma warning limits.
3. Four out of five consecutive points plot at a distance of 1-sigma or beyond from the center line.
4. Eight consecutive points plot on one side of the center line.

Those rules apply to one side of the center line at a time.

The problem with this approach is that while it can detect large shifts and runs effectively, it is completely useless for detecting small variations within 1-sigma. While statistically this may not be significant, but early detection of runs with small increments is useful in modern robust process control. The average run length (ARL) for small incremental changes is usually too large (> 10).

A simulated "neural net" that can be trained to identify special non-random patterns is used to recognize shifts and trends in a noisy univariate control chart. This method is then compared to the Shewhart Control Chart with Western Electric Rules to see the advantages and disadvantages. The ARL for the neural net is bounded above by the window size (W) which is 7 in this report and is much more sensitive than the Shewhart Charts with run rules.

2. Methodology

2.1. Neural Net

A neural net can be trained to reconize non-random patterns. Given a set of input patterns and the corresponding outputs as training set, the neural net can be trained to adapt to this "mode" of thinking by adjusting the weights of its nodes. When it is given a new pattern, the net will then look at the patterns that it learned and try to adjust the output values to give a best fit of the new patterns to the patterns it learned.

2.2. Training Set and Windowing

Using this property of neural network, we can give a set of artificially generated patterns for the net to learn and then give outputs to reflect the input pattern. it would also be nice if one pattern is reconized, the other pattern should be suppressed. With this in mind, the following patterns are used.

Pattern	Output
In Control	0 0 0 0
Increasing with different slopes and starting point	1 0 0 0
Decreasing with different slopes and starting point	0 1 0 0
Shift Up with different step sizes and starting point	0 0 1 0
Shift Down with different step sizes and starting point	0 0 0 1

Since number of inputs in the training set is fixed, we have to do the same for the control chart. A running window of size W is opened to sampled the process data for testing. Here an immediate tradeoff is what value of W one should use. Using a long window, the net is more immune to noise but the response will be delayed.

Once the window size is determined, the training set is of great importance. For example, choosing too small a slope to train will make it too susceptible to noise while choosing too large will make it insensitive to small changes.

2.3. Determination of Alarm

With the above training set, there can be four different indicators (I_i's), namely the four outputs. It is obvious that a full detection of a particular pattern will give a value closed to 1 while the others will be suppressed. Should an alarm be sounded for a certain output greater than a certain threshold value? What about false alarm? Also it is tedious to look at 4 outputs, is there a way to combine the outputs and give one combined output? There is no best answer to the above problems.

The method used in this report is filtering the different indicators and pooling the filtered outputs to give a combined warning signal X. Low pass filtering is used for smoothing the four outputs and eliminates false alarm triggered by pulses. The pooled output is a combined information to give an indication of in-control or out-of-control. Alarm is sounded if X passes through some threshold region.

3. Implementation

3.1. Neural Net

A neural net simulator written by Fariborz Nadi is used to simulate the "neural net". The program takes in input patterns and corresponding outputs and generate a multi-layer "neural net" with structure specified by the user. The number of input is the same as the window size and the number of output is four in this case. The less degree of freedom the neural net is given, provided the training patterns converge to the desired output, the better. Since using just the input and output layer didn't give a converging output, a three layers structure is used with the minimum hidden nodes just enough for convergence.

3.2. Training Data and Neural Net Specification

A window size (W) of 7 is chosen to balance the tradeoff between response time and noise immunity. The network thus is specified to be three layers with 7 input nodes, 10 hidden nodes as a second layer and 4 output nodes. The parameters file used is shown in fig.2.

The training set used is shown in fig.3 and the corresponding output in fig.4. Set 1 is for in-control operation; set 2 to 7 is for increasing slopes; 8 to 13 is for decreasing slopes; 14 to 22 is for step up and 23 to 31 is for step down.

From the above data, one may suggest that if we take the differentials between successive data points, then we should eliminate the use of different starting points and for each slope it will be a different constant level for training. Also, for the step data, it will become an impulse. There are several drawbacks of this approach. First, the above method is like taking derivatives from a noisy data and is very susceptible to noise. Also the neural net is not so good in recognizing different constant levels as it is trying to fit the input pattern to some reference pattern and every constant patterns look the same to the neural net and will confuse the net.

3.3. Input and Test Data

The input data (D_i's) has a mean (m) of 0.5 and a sigma (σ) of 0.05. The net takes in inputs from 0 to 1, therefore the samples in the window (S_i's) are scaled according to the following

$$S_i = \frac{D_i - m}{k\sigma} \times 0.5 + 0.5$$

k is a design parameter for the sensitivity of control chart. In this report, a value of 3 is used for k to give full scale data when the input is 3 sigma form the mean.

3.4. Filtering of Output and Warning Generation

A running average is an equivalent of low passing the indicators. A running size of 3 is chosen for smoothing out the indicators. The warning signal (X) is the combination of the filtered indicators.

$$X = FINC + FSUP - FDEC - FSDW$$

where:

FINC = filtered increasing indicator
FDEC = filtered decreasing indicator
FSUP = filtered step up indicator
FSDW = filtered step down indicator

If X is too positive, the process is out-of-control and is increasing. If X is too negative, the data is out-of-control and is decreasing. A value of ± 0.4 is chosen as the threshold. This is reasonable as if there is a strong indication of a trend, the indicator will be a full 1 and the filtered value will be closed to 0.33 while the others are suppressed, so a value of 0.4 is right above the noise level but sensitive enough for significant small changes.

3.5. Input Generation and Experiment

The Input is generated by BLSS and two sets of data are generated and compared with the performance of Shewhart Chart with Western Electric Rules. One set contains 180 data points with large variation of the mean. The second is generated with 60 data points and small variations of the mean. The sigma is controlled to be constant throughout the experiment.

4. Examples and Results

4.1. Small Variations

In fig.5, the data with small variations is shown and the warning signal with the alarm of the Western Electric Rules are shown. In fig. 6 are the pattern indicators. A 1 on the WE line means alarm by the WE rules and a value larger than 0.4 on the FOUT line indicates an alarm by the neural net. The data is set up such that it is random from 1 to 10 and then the mean changes as a sine wave of period 10 and amplitude sigma (0.05). The last 10 data is also random with mean back to 0.5.

The weakness of WE rules is completely exposed in this case. Since the process is varying in a small amount around the mean, the WE rules failed to detect most of the variations while the neural net picked up all the changes and made alarm in 17, 28, 36 and 47 while the WE rules can only detect it at 50. The pattern indicator can explain the reason for alarm and the X values shows which way the process is out of control right away.

Note in here, the net can reconize the small changes only after it moves the full window into the increasing data. Since the window size is 7, that explains why it sounds alarm around nx10+7. For large changes, there is no need for the window to fall completely on the increasing data as shown in the next example.

4.2. Large Variations

A large example, with 180 data points are generated with some large shifts and trends, is shown in fig.7 together with its warning signals. It is expanded into subsequent figures.

In fig.8 are data points 1 to 30 and fig.9 its indicators. The pattern is completely random here. WE gives no alarm at all but the net gives an alarm at 21. By looking at the indicator, there is a strong indication of step up shown in 21 which indicates a step up in 21 - 3 = 18. The stepping is a special indicator. It is delayed by about half of the window size (W) and in this case it is 3. This is so because in the training set the step point is approximately in the middle of the window. Several attempts has been made to shift the step to the front of the window to remove the delay. One experiment was tried by putting different weights on the sampled data within the window. This was not successful becasue by putting more weights on the new samples, one can ensure the early detection of the step but it also gives a lot of false alarm or wrong indications. As a result, the step has to be around the middle of the window and a delay of about W/2 is the biggest draw back of this approach.

Fig 10 and 11 are for data from 31 to 60. Here a big step up occurred at 32 and WE picked it up immediately while the net picked it up at 34. Note that because of the big step, the net should reconize it as a step in 35 but the indication of increasing is strong, an early alarm due to increasing mean will be sound a data point earlier! After 47, the process is back to normal again.

Fig 12 and 13 are for data from 61 to 90. A gradual increase of the mean occurred in here. The net picked it up at 66 for a single alarm and then alarm from 75 on while the WE rules sounded alarm from 78 on. The net is doing a lot better than the WE rules if the change is gradual even if the change of mean is big.

Fig 14 and 15 are for data from 91 to 120. A gradual decrease of the mean occurred. The net picked it up in 109 and then on. The WE rules sounded alarm from 110 on. Again the net is doing better.

Fig 16 and 17 are for data from 121 to 150. The mean kept on falling till 130 and then rise again from that point on. The net picked it up in 146 while the WE rules in 150. The net is doing better again.

Fig 18 and 19 are for data from 151 to 180. The mean kept on increasing and then immediately dropped with a sharp transition from increasing to decreasing and then back to normal. the net picked it up at 161 while the WE rules picked up in 162. The net is a step earlier again.

4.3. Average Run Length

Though a formal calculation of the ARL is hard to calculate, it is obvious that the ARL is bounded above by the window size (W) as for the small variations, the net can detect it once it moves the whole window within the runs. With large variations, the ARL is going to be less than W becasue the net will try to fit the large variation with the corresponding trained patterns when enough data points with the large variation moved into the window. Therefore, the ARL is approximately bounded by W/2 and by W.

5. Conclusions

The neural net approach is much better than the traditional approach of Shewhart Chart with Western Electric Rules in several ways. it is more sensitive to small changes while the WE rules failed to detect. The ARL is much shorter for small variations except for a step across the 3-sigma line. When the data jumped across the 3-sigma line, the WE rules sounds an alarm immediately while the net is delayed by half the window width. That is one reason for choosing a small window size and use filtering to eliminate the noise problem. The neural net also gives information of what kind of change occurs as it reconizes the trained patterns while the WE rules can only give indication of out-of-control. Also to avoide the above problem, one can combine the run rules with the "neural net" to give the best performance.

The results we have achieved so far show a huge potential for developing the "neural net" control chart. Further experiments should be done to include other patterns. e.g. cyclic pattern. The net can use running data to train itself adaptively for recognizing cyclic patterns instead of using some pre-

determined training patterns. Optimal size of filtering window and the window size should be found. A better training set may even give better sensitivity. Varying k, the scaling parameter adaptively to change the sensitivity for small changes and large changes of mean should be interesting.

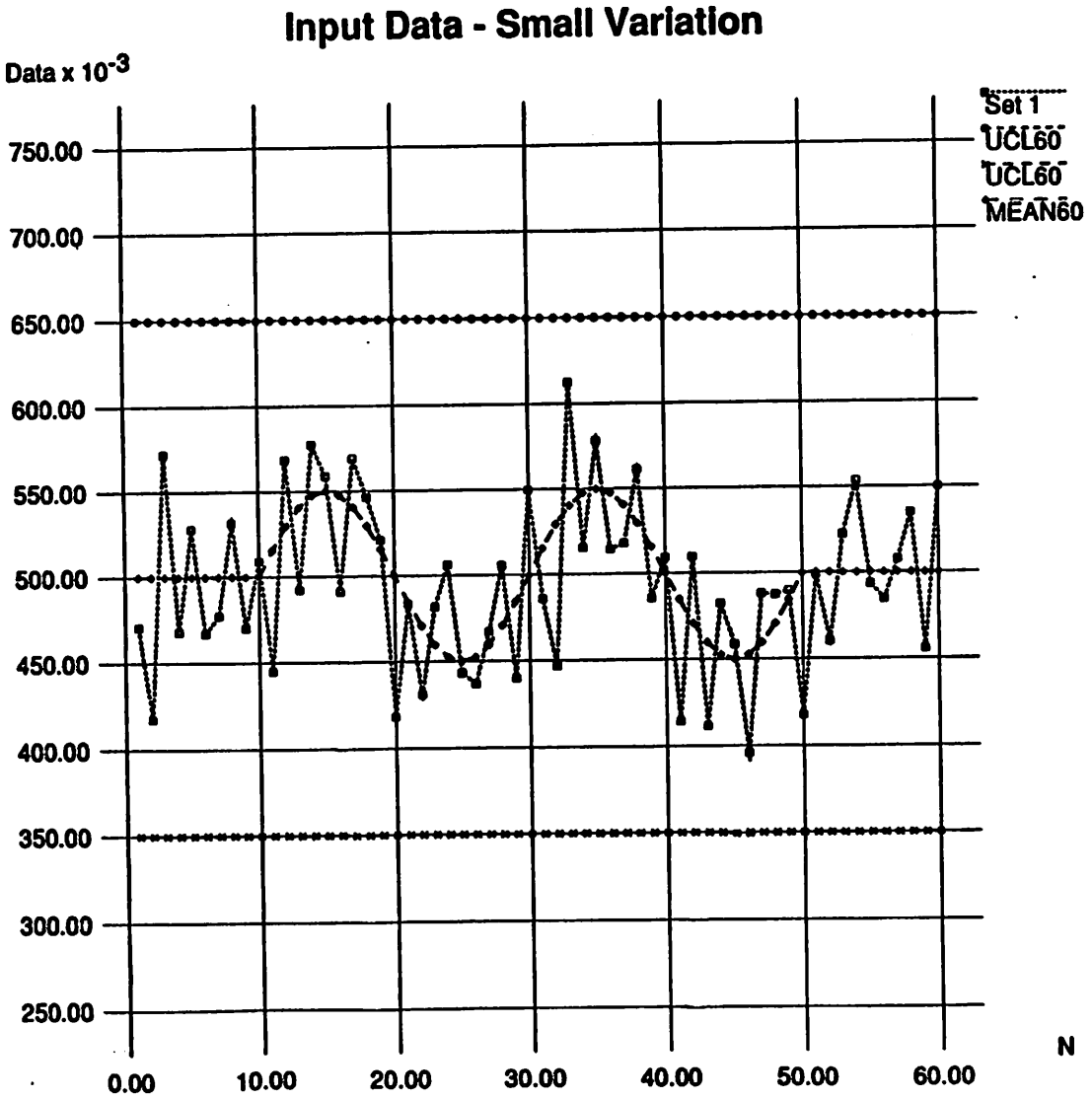


Figure 1

```
::::::::::::::
parameter
::::::::::::::
3
31
7 10 4
97
10000 1 0.0 1.0e-06 1.0e-05 1.0 1.0
train.in
1.0 1.0 1.0 1.0 1.0 1.0 1.0
train.out
actout6
weights6
error6
```

Figure 2

```
::::::::::::::
train.in
::::::::::::::
 1 0.50 0.50 0.50 0.50 0.50 0.50 0.50
 2 0.40 0.50 0.60 0.70 0.80 0.90 1.00
 3 0.40 0.48 0.57 0.65 0.73 0.82 0.90
 4 0.40 0.47 0.53 0.60 0.67 0.73 0.80
 5 0.50 0.58 0.67 0.75 0.83 0.92 1.00
 6 0.50 0.57 0.63 0.70 0.77 0.83 0.90
 7 0.60 0.67 0.73 0.80 0.87 0.93 1.00
 8 0.60 0.50 0.40 0.30 0.20 0.10 0.00
 9 0.60 0.52 0.43 0.35 0.27 0.18 0.10
10 0.60 0.53 0.47 0.40 0.33 0.27 0.20
11 0.50 0.42 0.33 0.25 0.17 0.08 -0.00
12 0.50 0.43 0.37 0.30 0.23 0.17 0.10
13 0.40 0.33 0.27 0.20 0.13 0.07 -0.00
14 0.40 0.40 0.40 1.00 1.00 1.00 1.00
15 0.40 0.40 0.40 0.90 0.90 0.90 0.90
16 0.40 0.40 0.40 0.80 0.80 0.80 0.80
17 0.40 0.40 0.40 0.70 0.70 0.70 0.70
18 0.50 0.50 0.50 1.00 1.00 1.00 1.00
19 0.50 0.50 0.50 0.90 0.90 0.90 0.90
20 0.50 0.50 0.50 0.80 0.80 0.80 0.80
21 0.60 0.60 0.60 1.00 1.00 1.00 1.00
22 0.60 0.60 0.60 0.90 0.90 0.90 0.90
23 0.60 0.60 0.60 0.00 0.00 0.00 0.00
24 0.60 0.60 0.60 0.10 0.10 0.10 0.10
25 0.60 0.60 0.60 0.20 0.20 0.20 0.20
26 0.60 0.60 0.60 0.30 0.30 0.30 0.30
27 0.50 0.50 0.50 0.00 0.00 0.00 0.00
28 0.50 0.50 0.50 0.10 0.10 0.10 0.10
29 0.50 0.50 0.50 0.20 0.20 0.20 0.20
30 0.40 0.40 0.40 0.00 0.00 0.00 0.00
31 0.40 0.40 0.40 0.10 0.10 0.10 0.10
```

Figure 3

```
::::::::::::::
train.out
::::::::::::::
 1 0 0 0 0
 2 1 0 0 0
 3 1 0 0 0
 4 1 0 0 0
 5 1 0 0 0
 6 1 0 0 0
 7 1 0 0 0
 8 0 1 0 0
 9 0 1 0 0
10 0 1 0 0
11 0 1 0 0
12 0 1 0 0
13 0 1 0 0
14 0 0 1 0
15 0 0 1 0
16 0 0 1 0
17 0 0 1 0
18 0 0 1 0
19 0 0 1 0
20 0 0 1 0
21 0 0 1 0
22 0 0 1 0
23 0 0 0 1
24 0 0 0 1
25 0 0 0 1
26 0 0 0 1
27 0 0 0 1
28 0 0 0 1
29 0 0 0 1
30 0 0 0 1
31 0 0 0 1
```

Figure 4

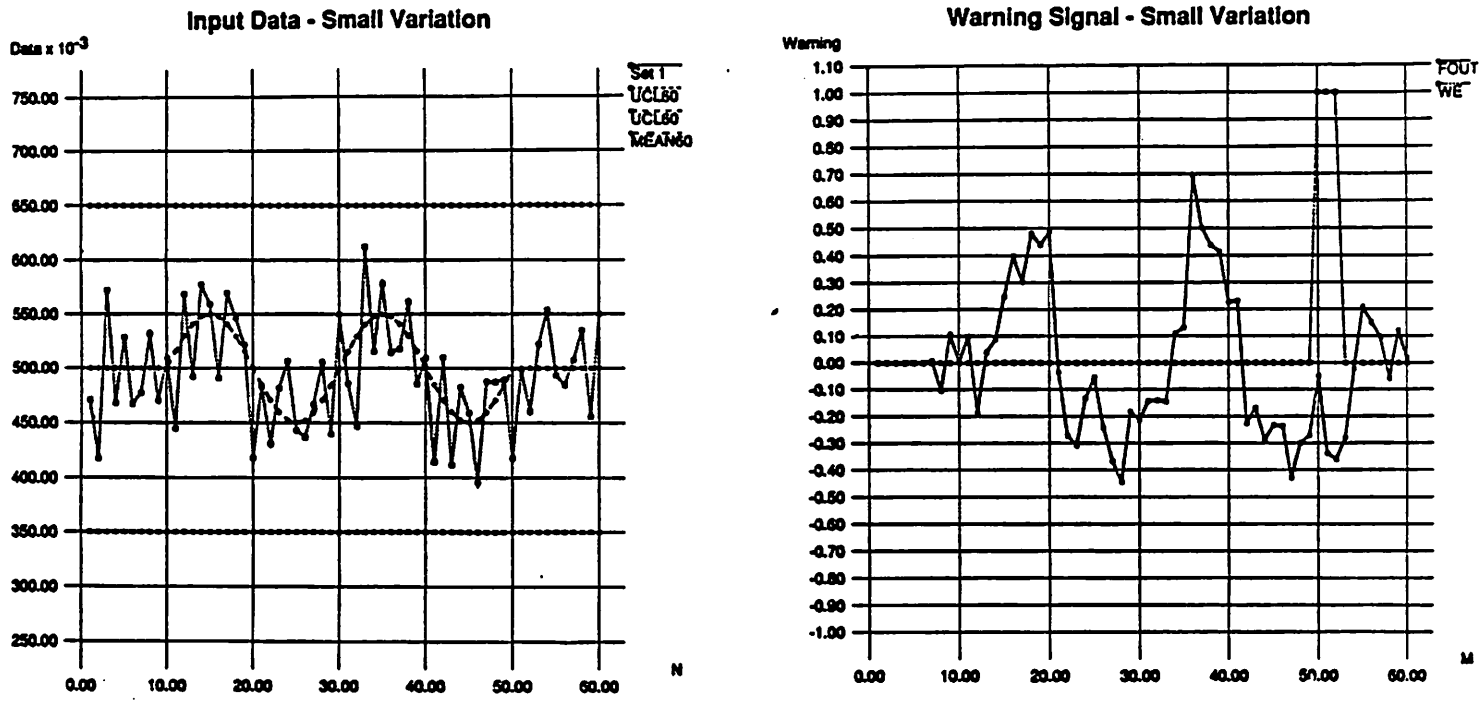


Figure 5

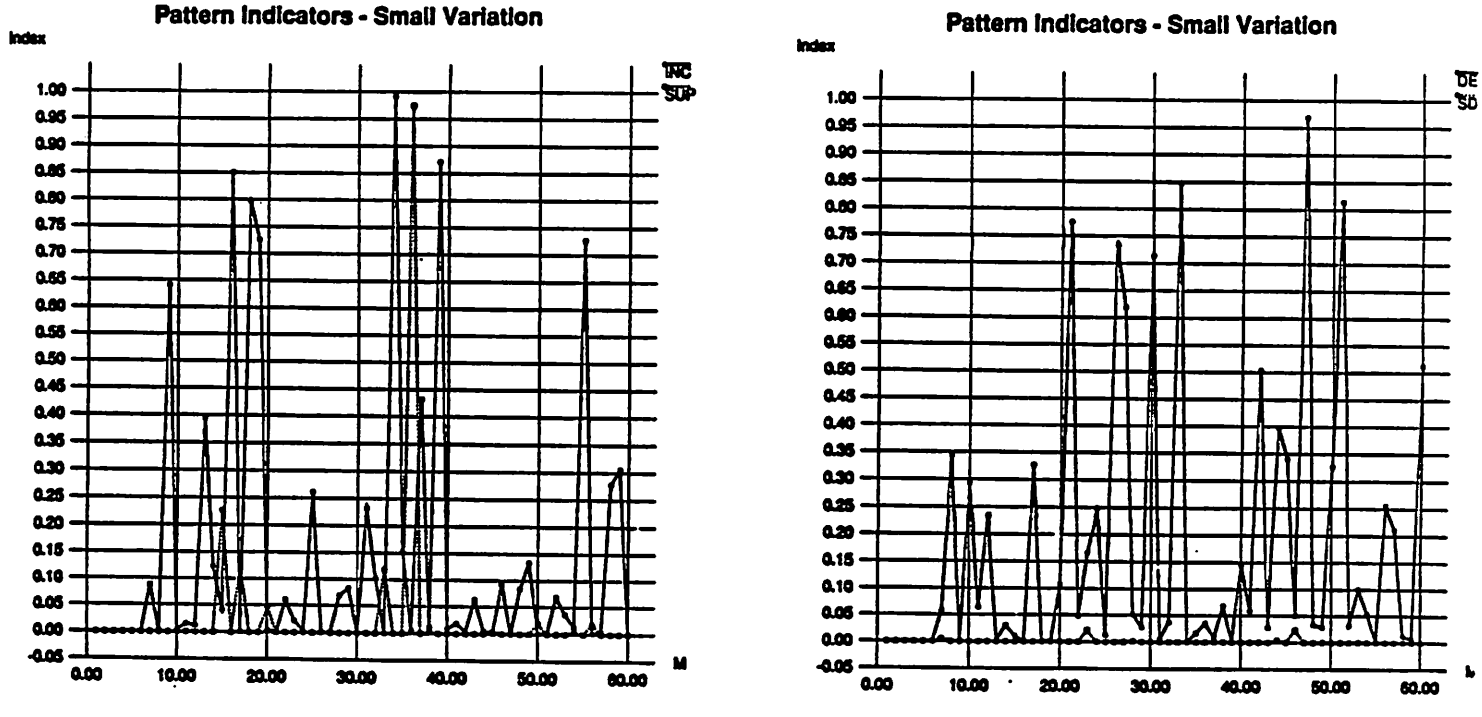


Figure 6

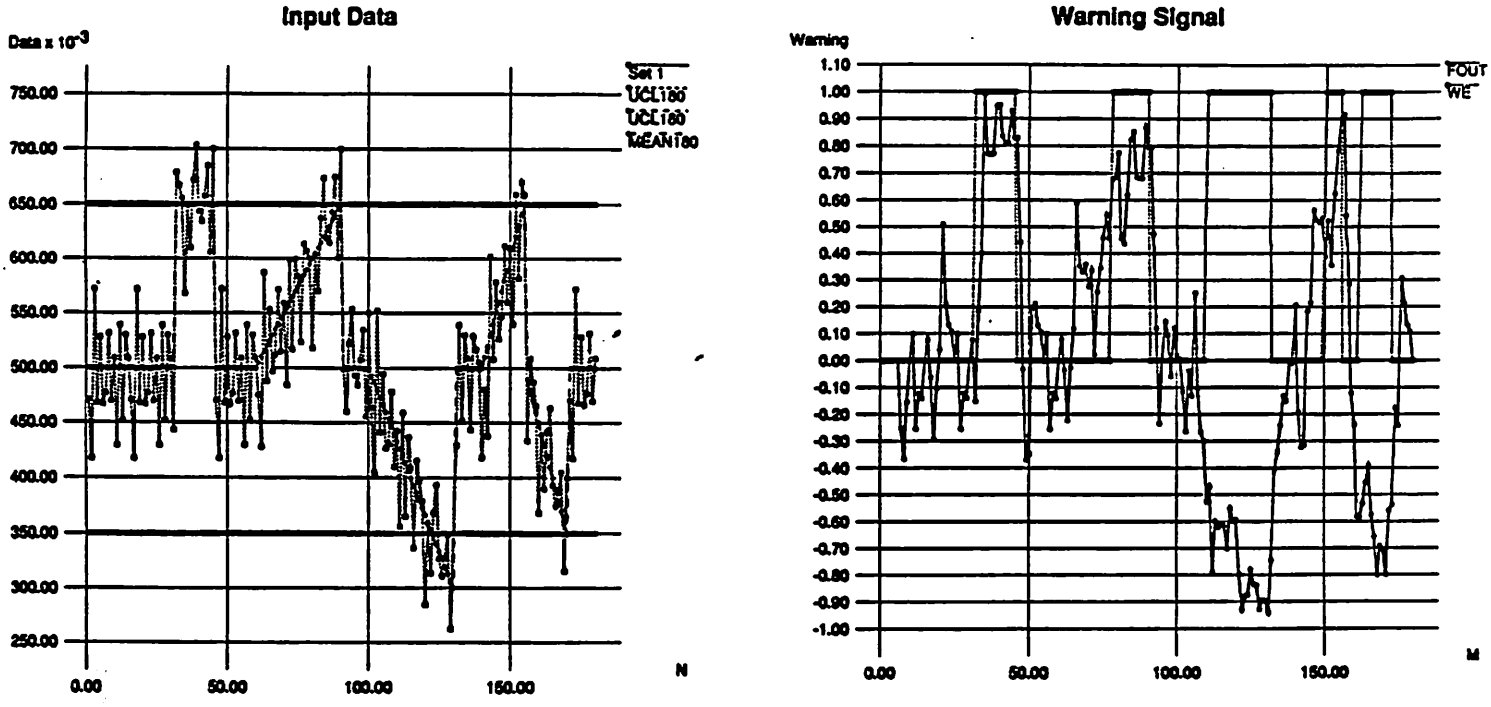


Figure 7

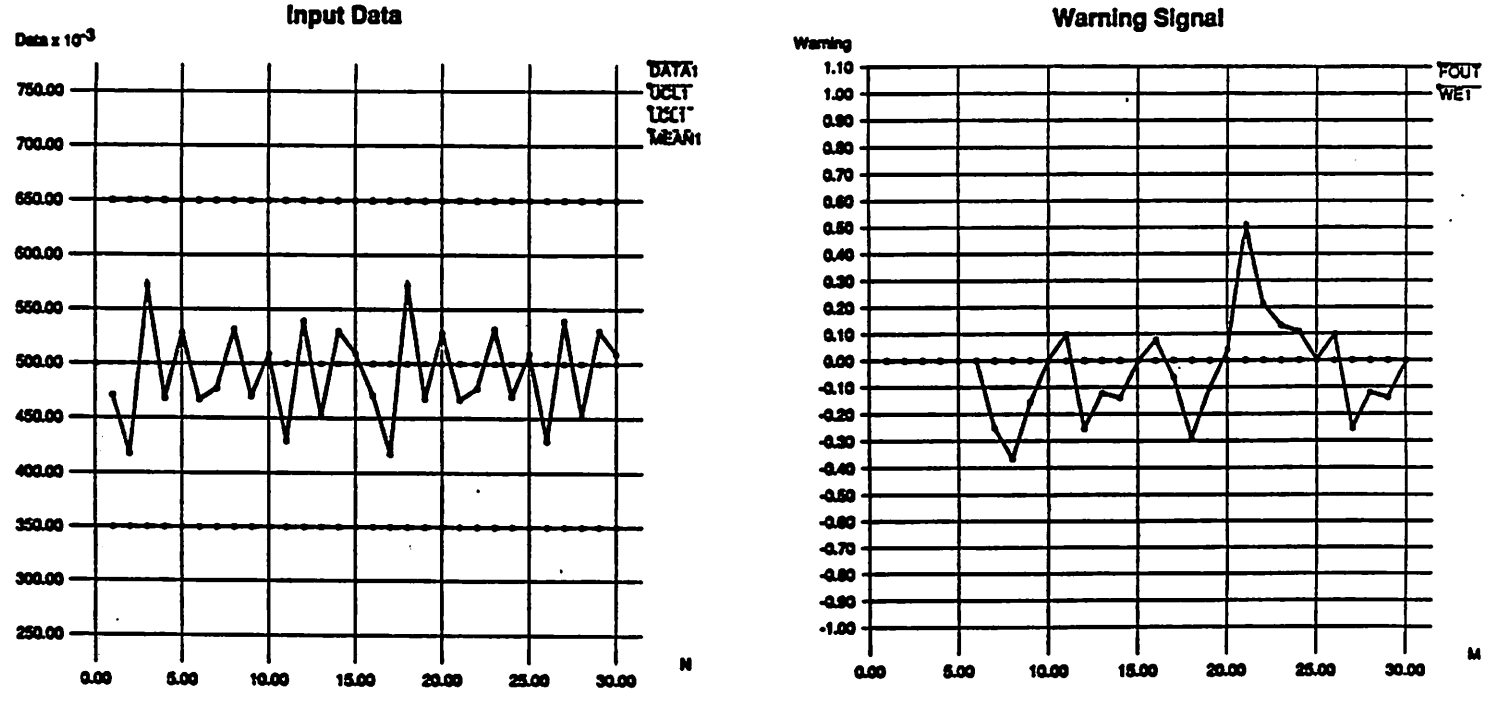


Figure 8

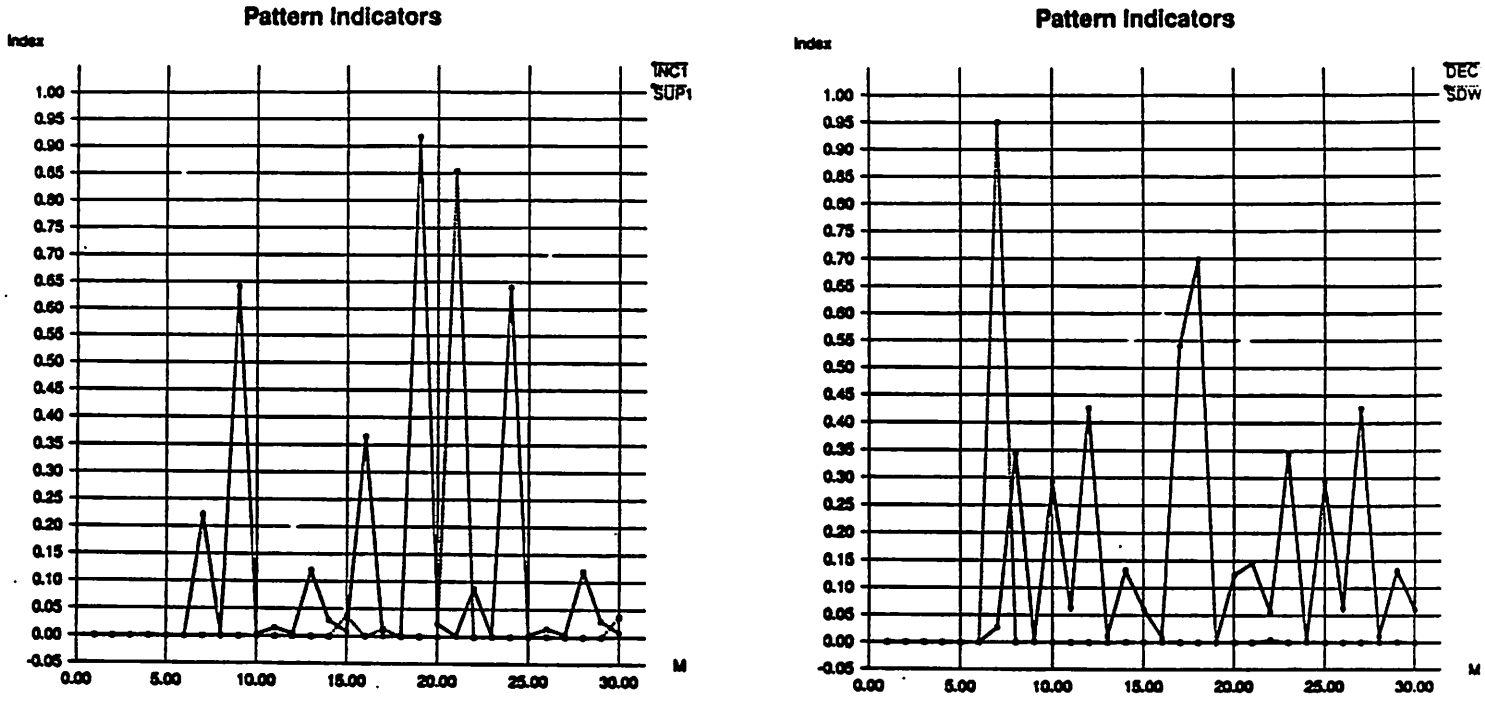


Figure 9

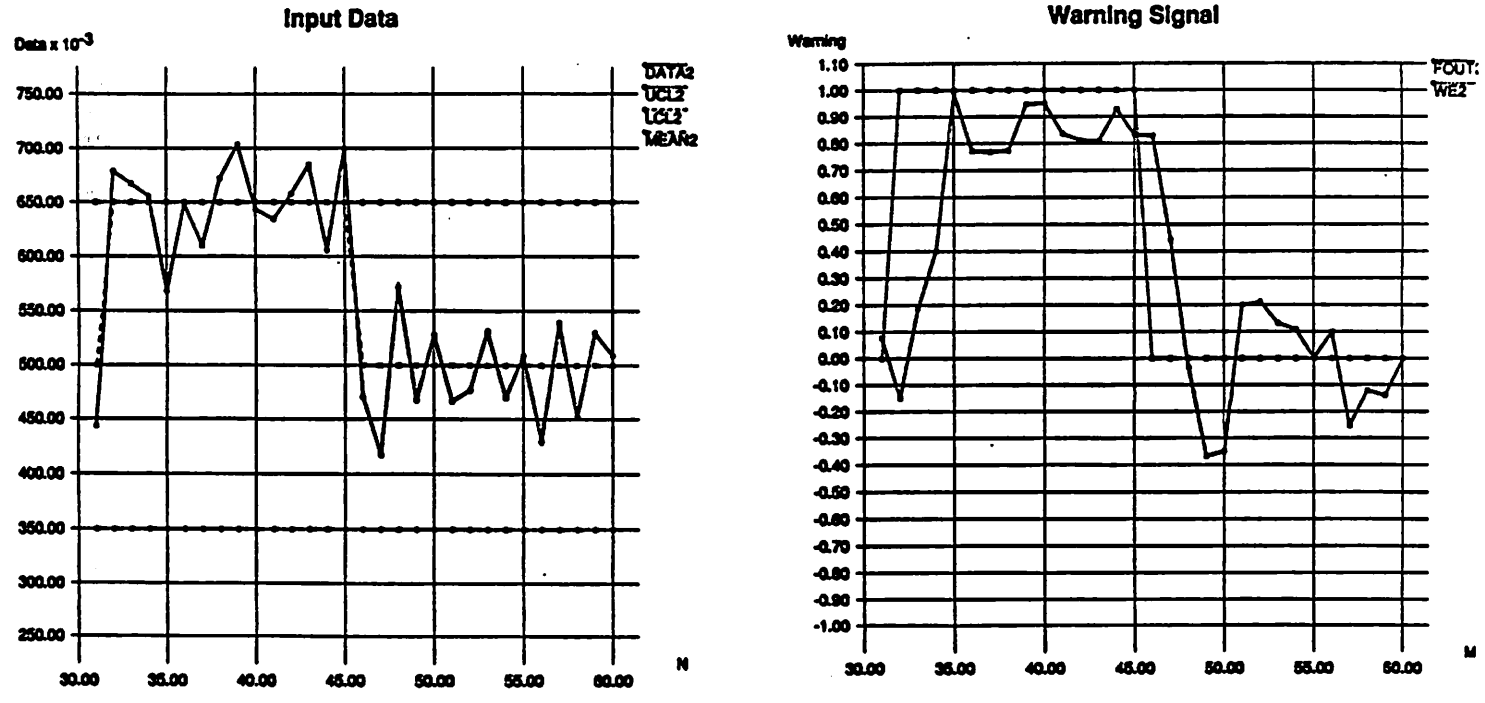


Figure 10

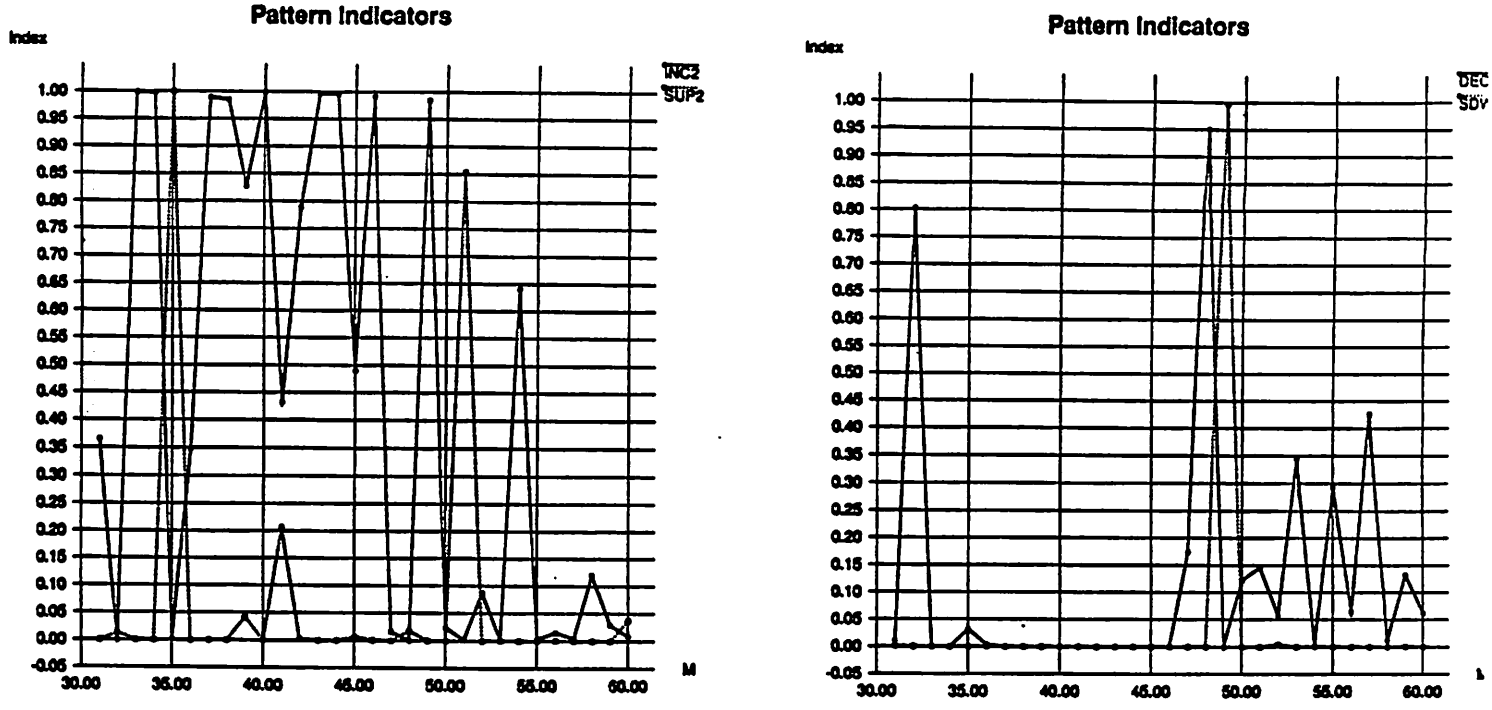


Figure 11

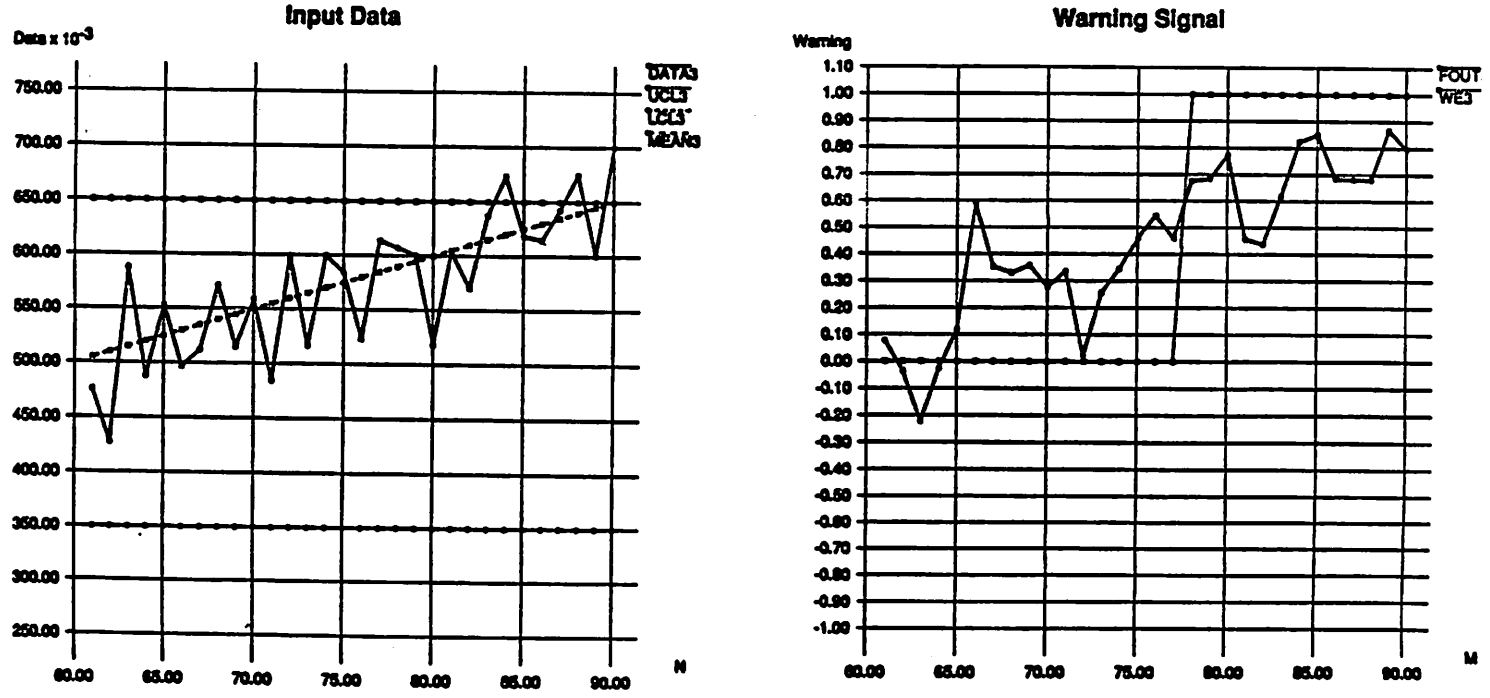


Figure 12

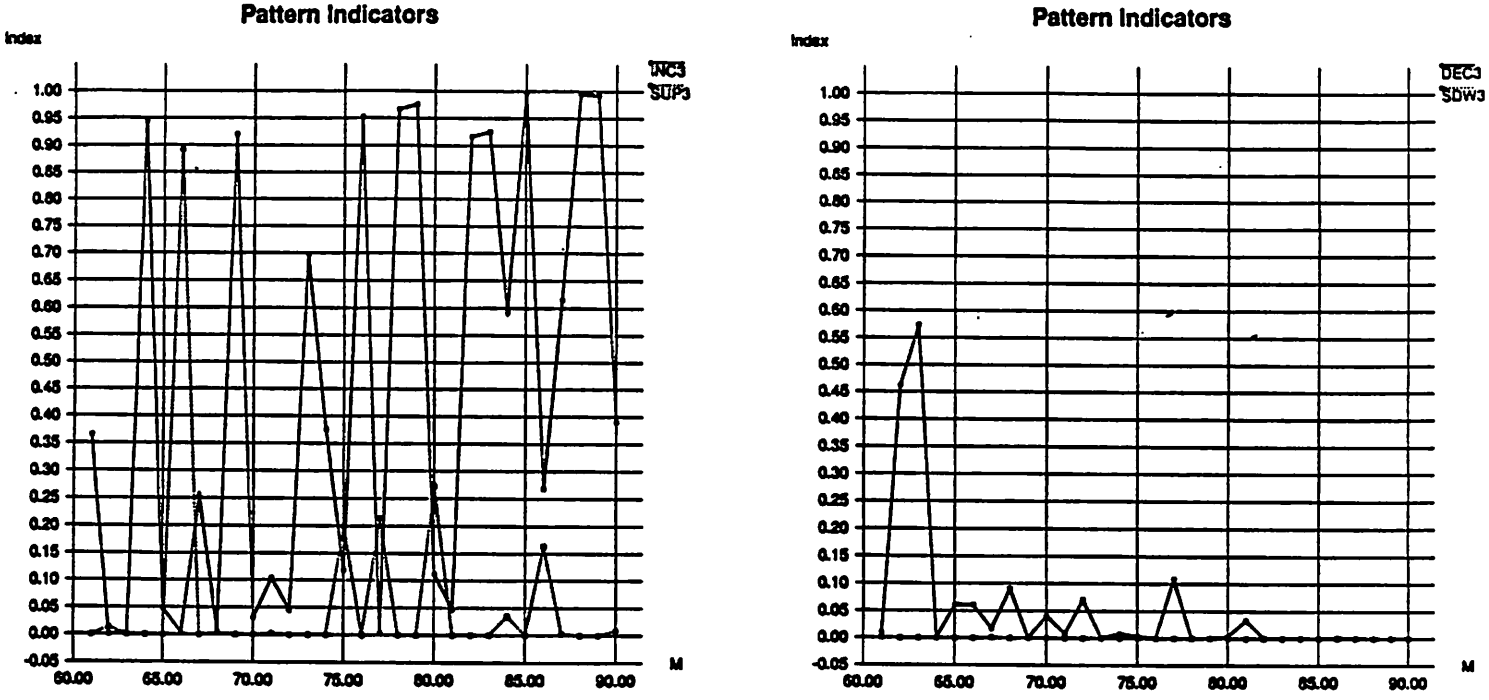


Figure 13

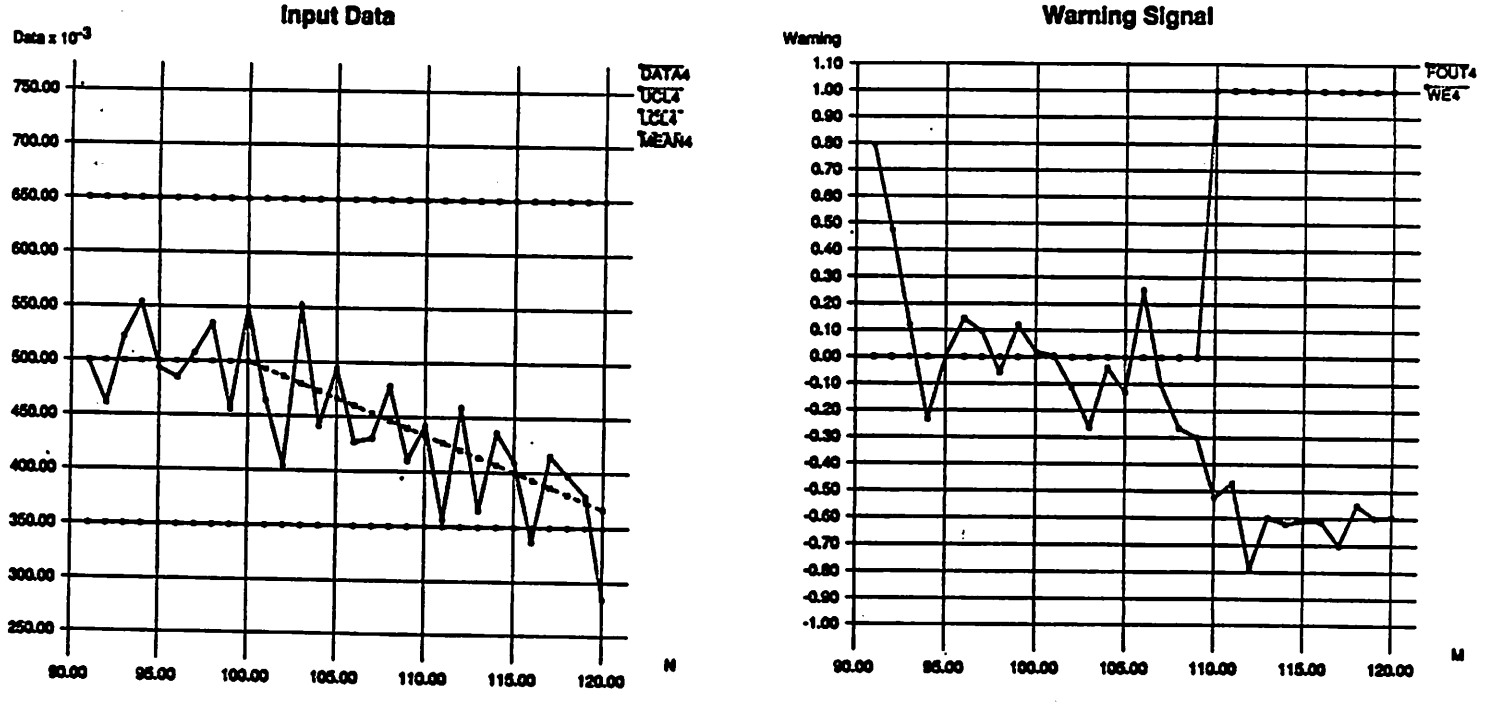


Figure 14

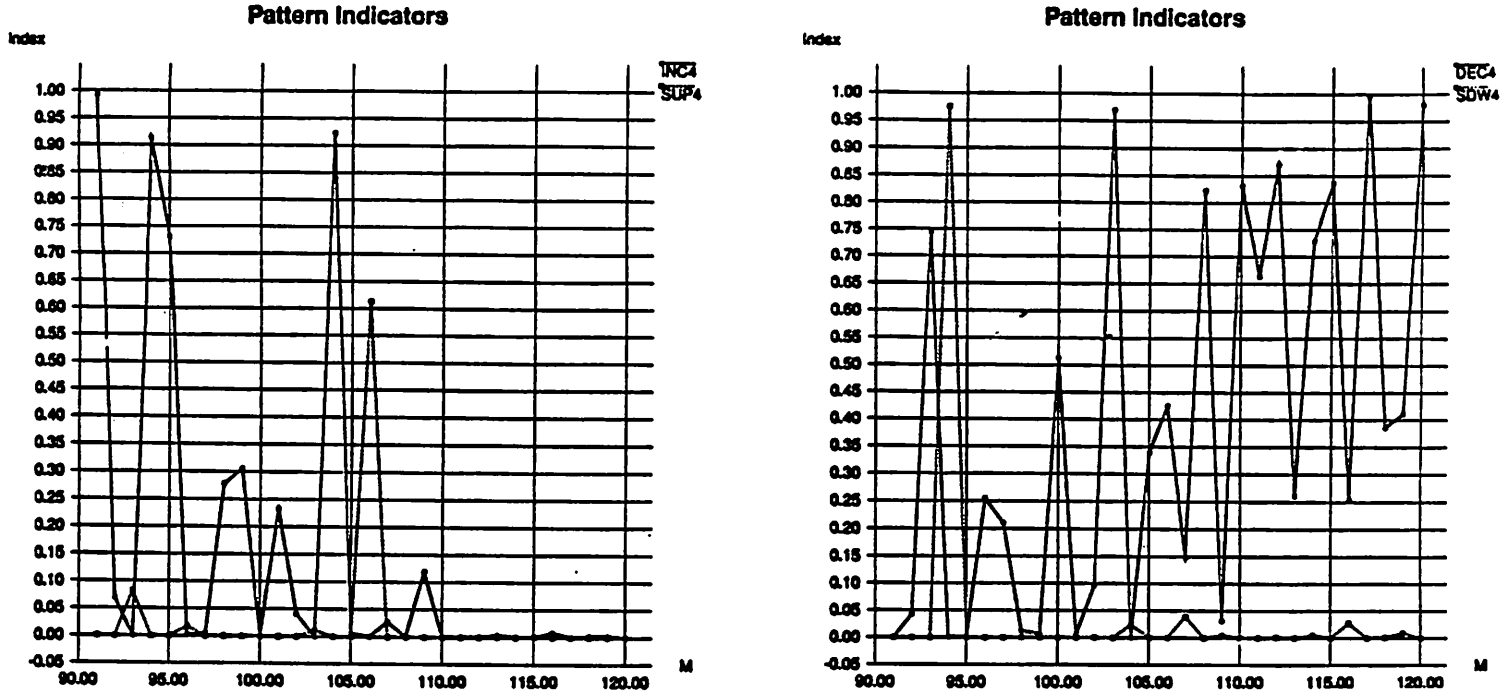


Figure 15

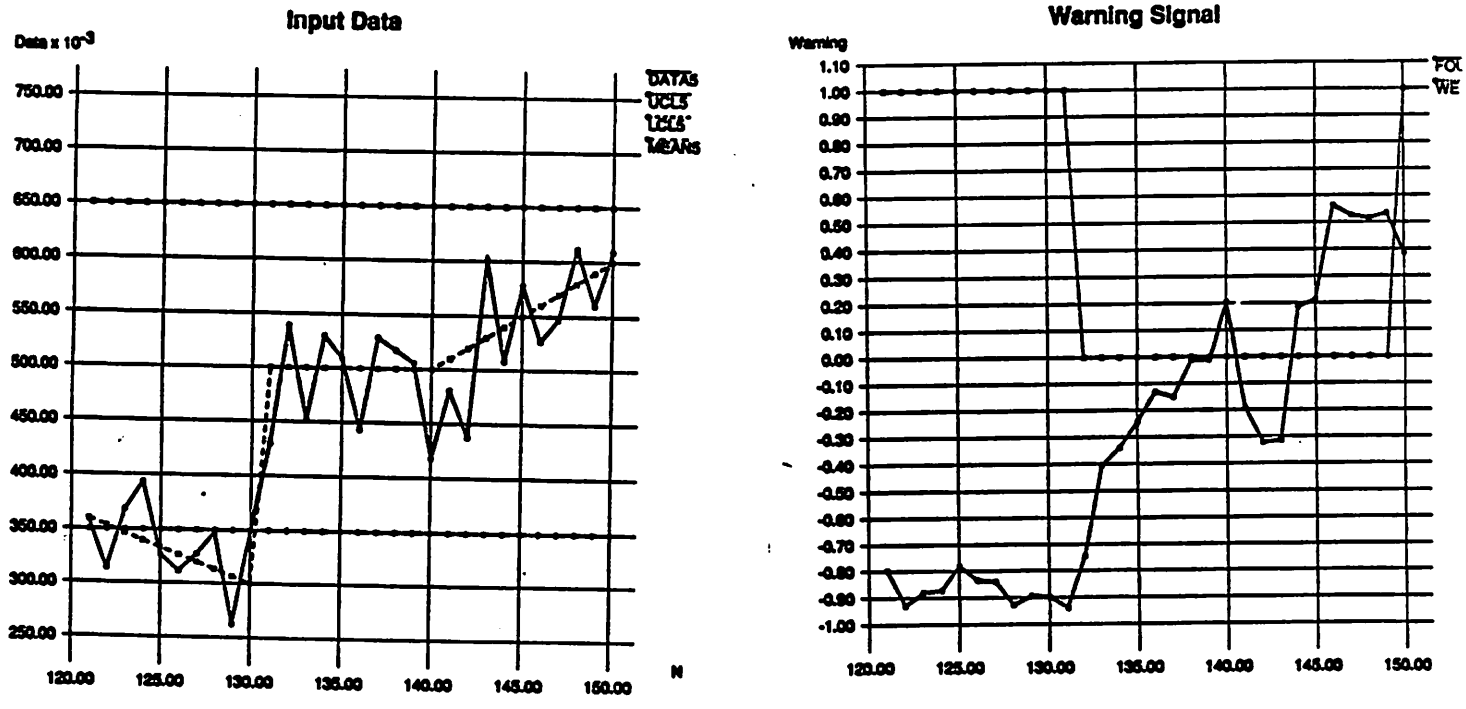
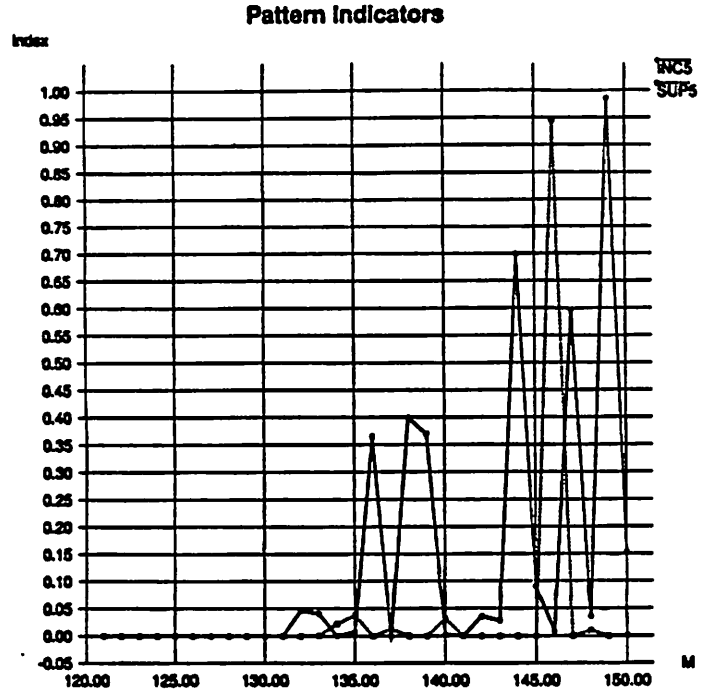


Figure 16



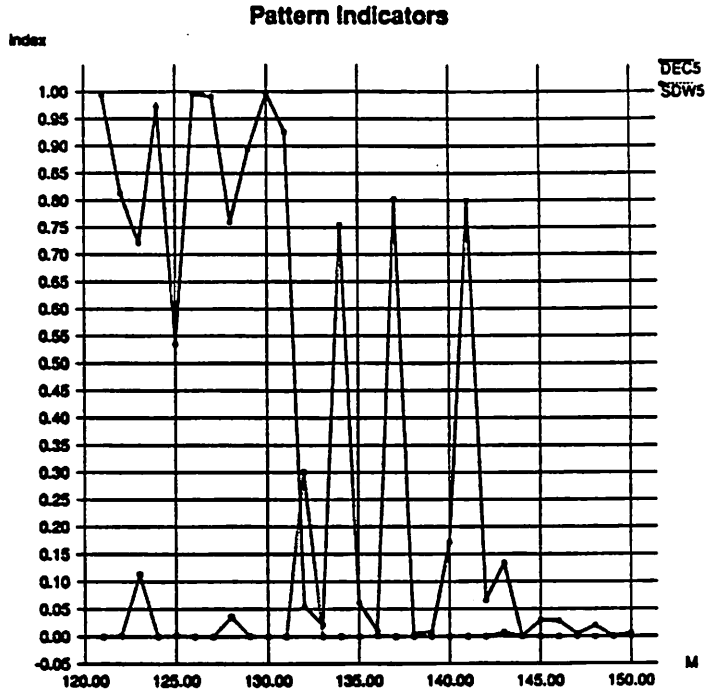
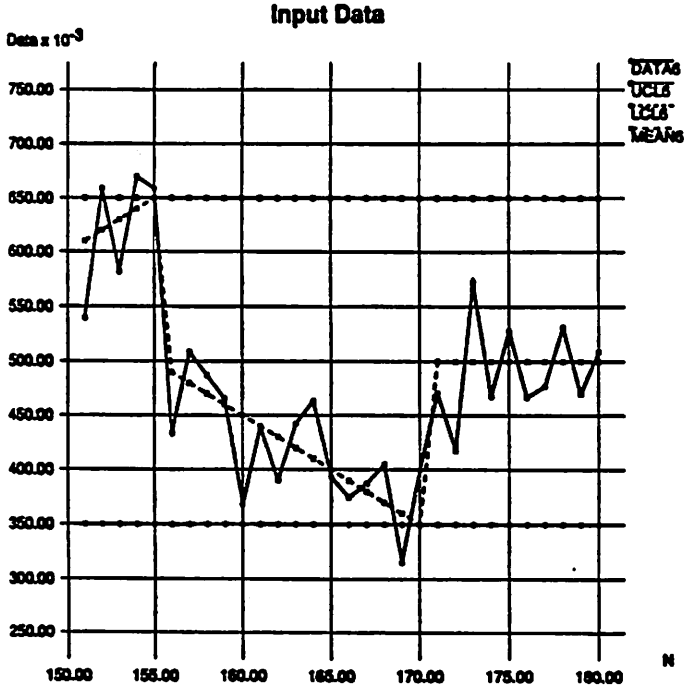


Figure 17



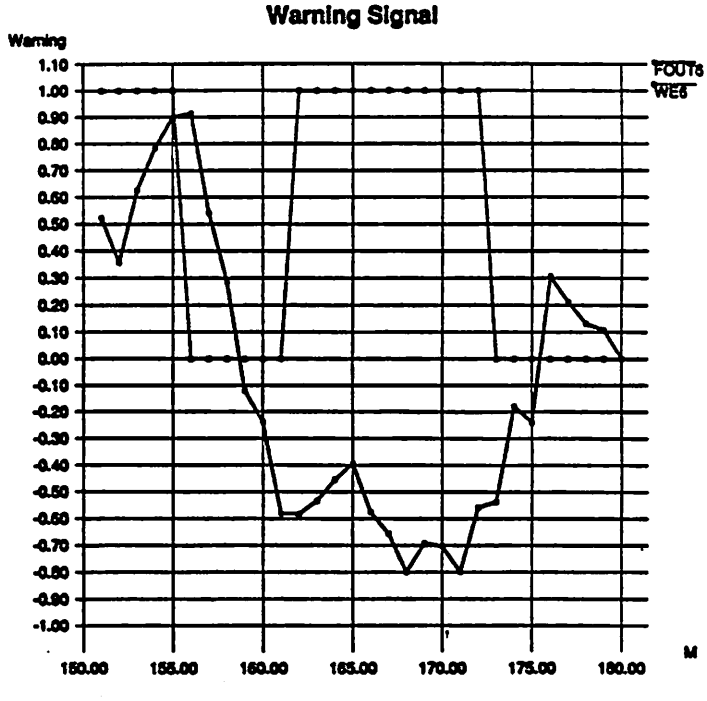
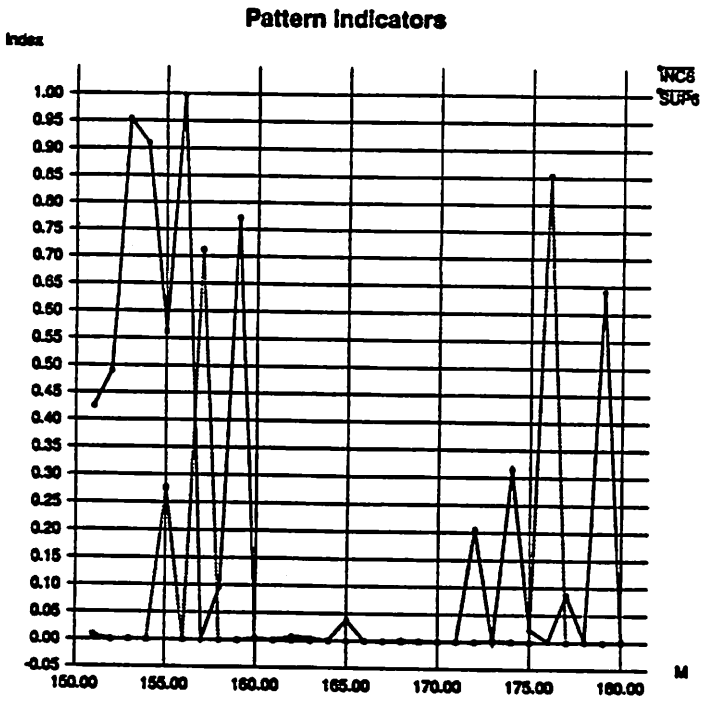


Figure 18


Pattern Indicators
Index
INC6
SUP6
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
150.00
155.00
160.00
165.00
170.00
175.00
180.00
M

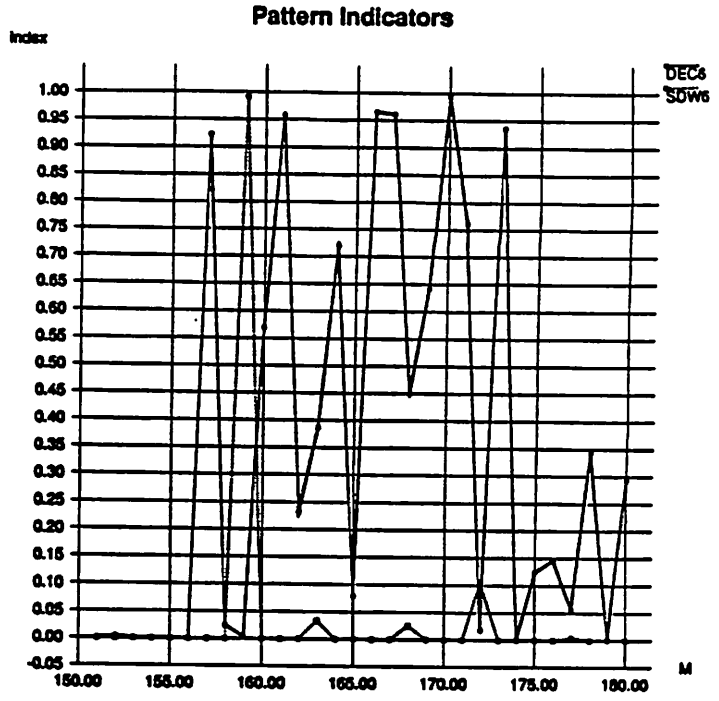

Pattern Indicators
Index
DEC6
SOW6
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
-0.05
150.00
155.00
160.00
165.00
170.00
175.00
180.00
M

Figure 19

Investigation of CUSUM Control Chart Run-Length Distributions

Christopher J. Hegarty

Abstract

The method of Brook and Evans [1] is implemented and used to evaluate the run-length distribution of cusum control charts. The method is shown to give accurate results and appears capable of generating the run-length distribution of any desired cusum chart.

1. Introduction

While CUSUM charts can be designed by reference to tables, occasionally it is valuable to evaluate the actual probability distribution of their run lengths. This might be important when interpolation or extrapolation from tabulated values cannot be relied upon, or when non-standard limits or violation rules must be investigated. Brooks and Evans [1] have described a procedure that can be used for the evaluation of this probability distribution. The objective of this project is the implementation of and experimentation with this algorithm.

2. Methodology

The original definition of the cusum chart from Page [2] is the following: Plot

$$S_n = \sum_{i=1}^{n} (D_i - K_1) \tag{1}$$

against the number of samples, where D_i is the i^{th} sample value, K_1 is the reference value and n is the number of samples. If S_n exceeds the value H, known as the decision interval, the process is considered to be out of control. If the value S_n ever drops below zero it is reset to zero. This is a one-sided test, but a two-sided test can be conducted by running two one-sided tests in parallel, choosing (for example), the second test sum to be

$$S_n^* = \sum_{i=1}^{n} (K_2 - D_i) \tag{2}$$

A description of the state of the process is the following: if $S_n^* > H_2$ and $S_n < H_1$, the process is in control, otherwise the process is out of control.

The choice of values for K_1 and H_1 (and K_2 and H_2 for a two-sided test) determines the properties of the test. Consider, for example, a test to determine whether the mean of a process has shifted from μ_0 to μ_1. A sensible choice for K will lie between μ_0 and μ_1, because if the average value of the samples Di becomes larger than K_1, ultimately the value of S_n will exceed H_1 and the process will be considered to be out of control. The value of H_1 determines the probabilities of type I and type II errors: a small value of H_1 will result in many false alarms due to random fluctuations of the value of S_n exceeding H_1, and a large value of H_1 will result in large type II error and a long run-length to detect the out of control condition. The most common criterion for choosing H_1 and K_1 in practice is to try and achieve certain average run lengths (ARL, the average number of points plotted before the value H_1 is exceeded); we require $ARL(\mu_0)$ to be large and $ARL(\mu_1)$ to be small.

An alternative approach to determining whether a cusum chart is out of control is the V-mask proposed by Barnard [3]. The approach consists of placing the V-mask on the cusum control chart with the origin 0 on the last value S_n and plotting points

$$S_n = \sum_{i=1}^{n} (D_i - \mu_0) \tag{3}$$

If all the previous plotted values $S_1, \cdots S_{n-1}$ lie within the two V-mask lines at an angle of θ, then the process is in control. This is a two-sided test: if any point lies above the upper arm, a downward shift in the mean is indicated, whereas if any point lies below the lower arm an upward shift is indicated. It can be shown [4] that the V-mask with parameters d and θ is equivalent to two one-sided

tests with:

$$K_1 = \mu_0 + w \tan(\theta) \tag{4a}$$

$$H_1 = d \tan(\theta) \tag{4b}$$

$$K_2 = \mu_0 - w \tan(\theta) \tag{4c}$$

$$H_2 = -d \tan(\theta) \tag{4d}$$

where w is the scale factor defined as the ratio of horizontal distance between points to unit distance on the vertical scale. A one-sided test can be conducted with a V-mask by just using one arm of the V-mask. Note that the average run length using two one-sided test can easily be calculated from the ARL of each test (ARL_1 and ARL_2) using the formula:

$$\frac{1}{ARL} = \frac{1}{ARL_1} + \frac{1}{ARL_2} \tag{5}$$

The parameters d and θ of the V-mask can be calculated by using

$$d = \frac{2}{\delta^2} \ln(1 - \frac{\beta}{\alpha}) \tag{6}$$

$$\theta = \arctan(\frac{\delta}{2w}) \tag{7}$$

where δ is the shift in process mean we desire to detect, in units of sample standard deviation, and α and β are the desired probabilities of type I and type II errors.

Brooks and Evans' technique is described in detail in [1], but a brief overview is as follows: Given a continuous variable D_i and two positive real numbers K and H, divide the region between K and H into T discrete regions each of width $\phi = (H - K)-T$. Let each region be considered as corresponding to a state E_i, where the cusum chart is in state E_i if the cumulative sum S_n satisfies

$$K + \phi\, i \le S_n < K + \phi\,(i+1) \qquad \text{if } 0 \le i < T \tag{8a}$$

$$S_n > T \qquad \text{if } i = T \tag{8b}$$

There are T+1 states, $E_0, E_1, \cdots E_T$. The sum S_n will change value and cause the system to move between these states, and the state E_T corresponds to the out of control condition and is therefore a terminal state. This system forms a Markov chain, and the transition probabilities between states are determined by the distribution of D_i, and are given by:

$$pr(E_i \to E_0) = pr(D_i \le K + (i+1/2)\ \phi) \tag{9a}$$

$$pr(E_i \to E_j) = pr((j-i-1/2)\ \phi \le D_i \le (j-i+1/2)\ \phi) \tag{9b}$$

$$pr(E_i \to E_t) = pr(D_i \ge K + (T-i-1/2)\ \phi) \tag{9c}$$

These values can be used to form the Markov chain transition probability matrix. Letting $P_r = pr((r-1/2)\ \phi \le D_i \le (r+1/2)\ \phi)$, and $F_r = pr(D_i \le K + (r+1/2)\ \phi)$, the transition probability matrix is

$$P = \begin{bmatrix} F_0 & P_1 & \cdots & P_j & \cdots & P_{T-1} & 1-F_{T-1} \\ F_{-1} & P_0 & \cdots & P_{j-1} & \cdots & P_{T-2} & 1-F_{T-2} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ F_{-i} & P_{2-i} & \cdots & P_{j-i} & \cdots & P_{T-1-i} & 1-F_{T-1-i} \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ F_{1-H} & P_{2-H} & \cdots & P_{j-H+1} & \cdots & P_0 & 1-F_0 \\ 0 & 0 & \cdots & 0 & \cdots & 0 & 1 \end{bmatrix} \tag{10}$$

If we form a matrix by removing the final row and column of P, the matrix R so formed has some very useful properties. In particular, the solution $\mu^{(s)}$ to the equation

$$(I - R)\mu^{(s)} = sR\mu^{(s-1)} \tag{11}$$

is the s^{th} factorial moment of the run length distribution. The i^{th} element of $\mu^{(s)}$ is the s^{th} factorial moment for state E_i, $\mu_i^{(s)}$. The factorial moments are defined as

$$\mu_i^{(s)} = E\{X_i^{(s)}\} = E\{X_i\ (X_i-1)\ \cdots\ (X_i-s+1)\} \tag{12}$$

where X_i is the number of steps required to reach state E_T from state E_i. In particular, X_0 is the number of steps required to reach state E_T from the starting state. From (12), it is clear that $\mu_0^{(1)} = E\{X_i\}$, which is the average run length.

The factorial moments are not particularly useful in statistics (except for the moments of some discontinuous binomial distributions), but it is easy to calculate the central moments from the factorial moments [5]. This means that not only are we able to calculate the ARL, but also the higher-order moments of the run length and so determine the variance, skewness and kurtosis of the run length distribution, but in this paper only mean and variance shall be calculated.

Note that we can calculate the parameters of the distribution of the run length for each state E_i, so it is possible to discover what the behavior of the chart will be for any desired value of S_n. For the purposes of this paper we chose only to examine the ARL starting at $S = 0$. It is also possible to calculate the cumulative probability function of the run length. If we set

$$\mathbf{L}_1 = (\mathbf{I} - \mathbf{R})\mathbf{1} \tag{13}$$

then the first element of $\mathbf{L}_1$ is the probability that the run length is one. Similarly if we define

$$\mathbf{L}_n = \mathbf{R}\mathbf{L}_{n-1} = \mathbf{R}^{n-1}\mathbf{L}_1 \tag{14}$$

the first element of $\mathbf{L}_n$ is the probability that the run length is n. While (14) gives us a way to calculate the distribution of run length as accurately as we desire, for normal applications the average run length can be several hundred, and it may necessary to work with values of n up to several times the ARL. While such calculations are certainly not beyond the capabilities of a modern workstation, it is possible to obtain approximations to the upper percentage points of the distribution much more easily. An approximation for $\mathbf{L}_n$ is

$$\mathbf{L}_n = (1 - \lambda)\lambda^{n-1}\left(\frac{\sum \mathbf{y}}{\sum \mathbf{xy}}\right)\mathbf{x} \tag{15a}$$

$$= (1 - \lambda)\lambda^{n-1}\mathbf{x}' \tag{15b}$$

where λ is the maximum real eigenvalue of R, x and y are the right- and left-hand eigenvectors corresponding to λ respectively. It may be shown that because of the properties of R, $\lambda < 1$. Similarly an approximation to the probability that the run length will be greater than n may be found from

$$1 - \mathbf{F}_n \approx \lambda^{n-1}\left(\frac{\sum \mathbf{y}}{\sum \mathbf{xy}}\right)\mathbf{x} \tag{16}$$

It is also possible, given a probability α, to calculate the approximate value of the upper-α percentage point of the distribution of run length. Let

$$c_i = \frac{x_i \sum y}{\sum xy} \tag{17}$$

Note that an approximation to ARL is ARL $\approx c_0(1 - \lambda)$. The upper-α percentage point of the run length starting from state i is

$$r_i(\alpha) \approx 1 + (1/\log\lambda)\log\left(\frac{\alpha}{c_i}\right) \tag{18}$$

For large run length, a better approximation is

$$r_i(\alpha) \approx 1 + (1/\log\lambda)\log\left(\frac{\alpha}{c_0}\right) \tag{19}$$

It is clear that this technique promises to reveal a great deal about the distribution of run lengths. One

important point about the above analysis is that the accuracy of the result depends on the value of T chosen, since the approach discretizes a continous function. In this paper, the probability distribution is always taken as Gaussian, but any distribution can be chosen, and exact answers can be obtained for discrete distributions of D. Since calculation involves inversion of T×T matrices, it is obviously important to keep T relatively small, and so T = 20 was used for all calculations other than accuracy checks carried out at T = 25 and T = 50.

3. Implementation

The implementation of the above algorithms is fairly straightforward, requiring nothing more than matrix multiplication, inversion, evaluation of eigenvalues and eigenvectors, and the ability to accurately and quickly estimate the cumulative probability distribution of the Gaussian. Matrix inversion and eigenvalue algorithms were taken directly from [6], with only minor modifications in detail. Solution for eigenvectors was via inverse iteration. Calculation of the central moments from the factorial moments was carried out using the equations in [5], and a rational polynomial expansion was used for the Gaussian cumulative distribution function [7].

4. Results

The results of this project were the output of the algorithm and analysis of that output. It was discovered that the algorithm performed very well, with good agreement between results calculated with it and those available from tables and approximations. Test calculations were carried out for a wide variety of examples, including discrete distribution functions, and verified with tables and numerical approximation. The next section will give some examples of plots of the probability density function of run length.

5. Examples

Consider the following example: Our process has mean $\mu_0 = 4$, and sample mean $\sigma_{\bar{x}} = 0.25$. It will produce useable output for $\mu \leq \mu_0 + 2\sigma_{\bar{x}} = 4.5$. Set up a cusum chart to detect this shift and determine the run length distribution. Using (7), $\delta = 2$ and let $w = 1$, so $\theta = 45°$. Chose $\alpha = 0.01$, and $\beta = 0.01$, so that $d = 2.30$. It then follows that $K_1 = \mu_0 + w \tan(\theta) = \mu_0 + \delta/2 = 4.5$. $H_1 = d \tan(\theta) = 2.30$. From tables of cusum chart values of $w \tan(\theta)$ and d, $ARL(\mu_0) \approx 500$, $ARL(\mu_0 + 2\sigma_{\bar{x}}) \approx 3.06$.

Using the technique outlined in section III above, we can find the ARL and the standard deviation of ARL for both nominal $\mu = \mu_0$ and at the operating limit $\mu = \mu_0 + 2\sigma_{\bar{x}}$. These values appear in Table 1 for different values of T.

	$\mu = \mu_0$		$\mu = \mu_0 + 2\sigma_{\bar{x}}$	
T	ARL	σ(RL)	ARL	σ(RL)
5	463.1	461.3	3.059	1.55
10	473.7	472.0	3.047	1.53
20	476.1	474.4	3.045	1.53
25	476.4	474.7	3.044	1.53
50	476.8	475.0	3.044	1.53
100	476.9	475.1	3.043	1.53

Table 1: Mean and variance of run length for normal operation and at the operating limit for different values of T.

Note that there is little change in the values in the table for T > 20 and consequently this value was chosen for subsequent analysis. The values of ARL from the table agree with those interpolated from standard tables. We can also evaluate the distribution of run length using Eqs. (13) and (14). The results of the analysis appear in figure 1(a) and (b). Note that in figure 1(b) the eigenvalue approximation to the run length distribution has been calculated, and give fair agreement for run lengths greater

than 4. The eigenvalue approximation is not shown in figure 1(a) because it is so close to the matrix calculations as to be indistinguishable. From (18), we can also calculate approximations to the upper percentage points of the distribution. For the case of $\mu = \mu_0$, the matrix calculation gives $r_i(0.05) = 1421$, and the approximation in (18) yields $r_i(0.05) = 1416$. For the case of $\mu = \mu_0 + 2\sigma_{\bar{x}}$, the matrix calculation gives $r_i(0.05) = 7$, and (18) gives $r_i = 6.5$.

We can also plot the distribution of run length as a function of the value of μ, as in figure 2. This type of data could be very useful in practice for the comparison of different cusum chart designs with the same values of $ARL(\mu_0)$ and $ARL(\mu_0 + 2\sigma_{\bar{x}})$.

Figure 3 contains a plot of the ratio of the average run length to the standard deviation of the run length as a function of average run length. Note that for large run lengths, ARL and the standard deviation of the run length are nearly equal. Furthermore, the very good agreement between the predictions of (15) and the matrix calculations for Fig. 1(a) means that for the case of large ARL, a reasonable approximation to the distribution of run length can be obtained from (15) and we are justified in saying that the ARL is well-approximated by a geometric distribution with parameter λ, but with a multiplying constant x'. For small ARL, the geometric approximation is not very accurate.

As a final example, consider a two-sided test based on the earlier one: i.e., design a cusum chart to test $\mu = \mu_0 \pm 2\sigma_{\bar{x}}$. So $K_2 = 3.5$ and $H_2 = -2.30$. Fig. 4 contains a plot of ARL as a function of mean deviation. Note that the average run length for nominal conditions is half the value of the one-sided test (since the two tests have the same ARL at $\mu = \mu_0$), but that the curve asymptotically approaches that of the one-sided case, since the ARL of the other test when the mean is close the one of the control limits will be very large. For example, $ARL(\mu_0 - 2\sigma_{\bar{x}}) = 16.7 \times 10^6$ for the test with $K_1 = \mu_0 + 2\sigma_{\bar{x}}$. Note that evaluation of the moments of the run-length distribution for the two-sided test is more difficult than for the one-sided case; it is necessary to use the definitions of the moments. For example, the variance of the two-sided run length distribution may be calculated using

$$\text{Variance} = \sum_{i=1}^{\infty} (x - ARL)^2 [f_1(i)(1 - F_2(i)) + f_2(i)(1 - F_1(i))] \qquad (20)$$

where f_1 and f_2 are the probability density functions of the two run length distributions, and F_1 and F_2 are the cumulative probabilities. Using this formula to calculate variance and plotting the ratio of ARL to standard deviation for the two-sided case, we arrive at Fig. 5. Except for near $5 = 50$, the variance of the two-sided cusum chart run length distribution may be approximated by choosing the smallest of the two variances given by the central moment calculations, so ordinarily the calculation of (20) is unnecessary.

6. Conclusions

This technique is very powerful and permits rapid evaluation of the distribution of run length for a cusum chart. The matrix calculations in (13) and (14) are capable of accurately determining the run length distribution for any values of K and H desired, although for values leading to large ARL the much faster eigenvalue approximation of (15) is quite accurate. This technique could readily be used in a control chart design package for rapid evaluation of cusum charts. Although the technique was originally intended for use with one-sided cusum charts, extension to two-sided charts in straightforward. Calculation of the moments of the two-sided distribution is more time consuming than the one-sided case, but is computationally feasible.

References

[1] D. Brooks and D. A. Evans, "An approach to the probability distribution of cusum run length", Biometrika., vol. 59, no. 3, pp. 539Q549, 1972.

[2] E. S. Page, "Continous inspection schemes", Biometrika, vol. 41, pp. 100Q115, 1954.

[3] G. A. Barnard, "Sampling inspection and statistical decisions", J. Royal Stat. Soc. (B), vol 21, no. 2, pp 239Q257, 1959.

[4] K. W. Kemp, "The use of cumulative sums for sampling inspection schemes", Appl. Stat., vol 11, pp 16-31, 1962.

[5] S. M. Kendall and A. Stuart, "Distribution theory", in The Advanced Theory of Statistics, Volume 1, pp 65Q68, 4th edition, 1977.

[6] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, "Numerical recipes in C", Cambridge University Press, Cambridge, 1988.

[7] Formula 26.2.17, p 932, Handbook of Mathematical Functions, M. Abramowitz andI. Stegun, Ed., National Bureau of Standards Applied Mathematics Series, 10th printing, 1972.

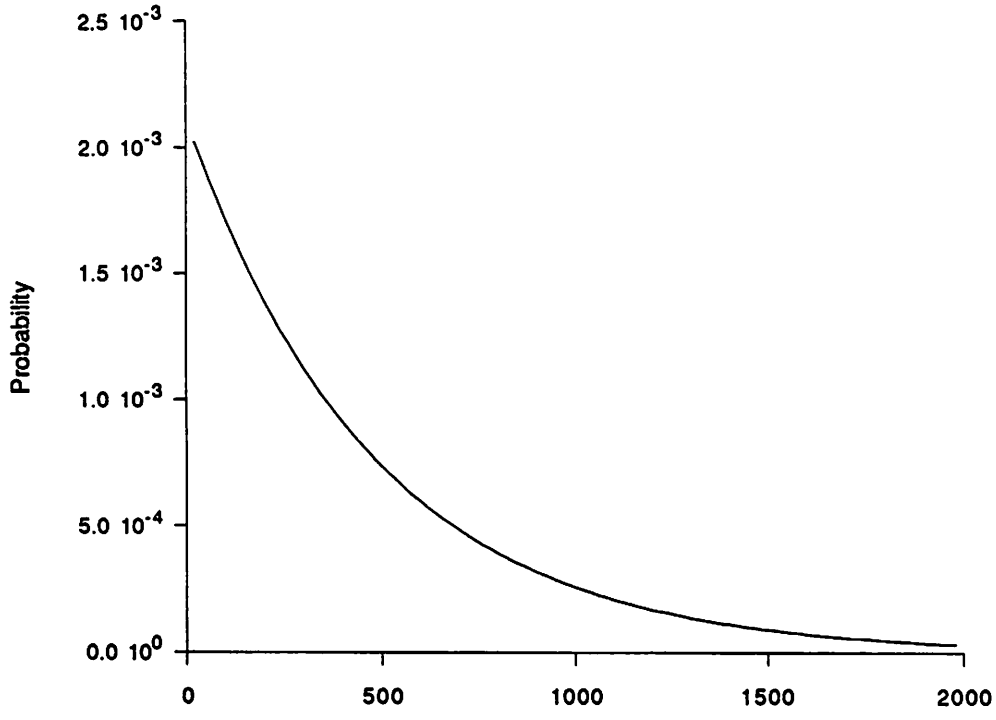


Figure 1a. Run length probability density function at $\mu = \mu_0$.

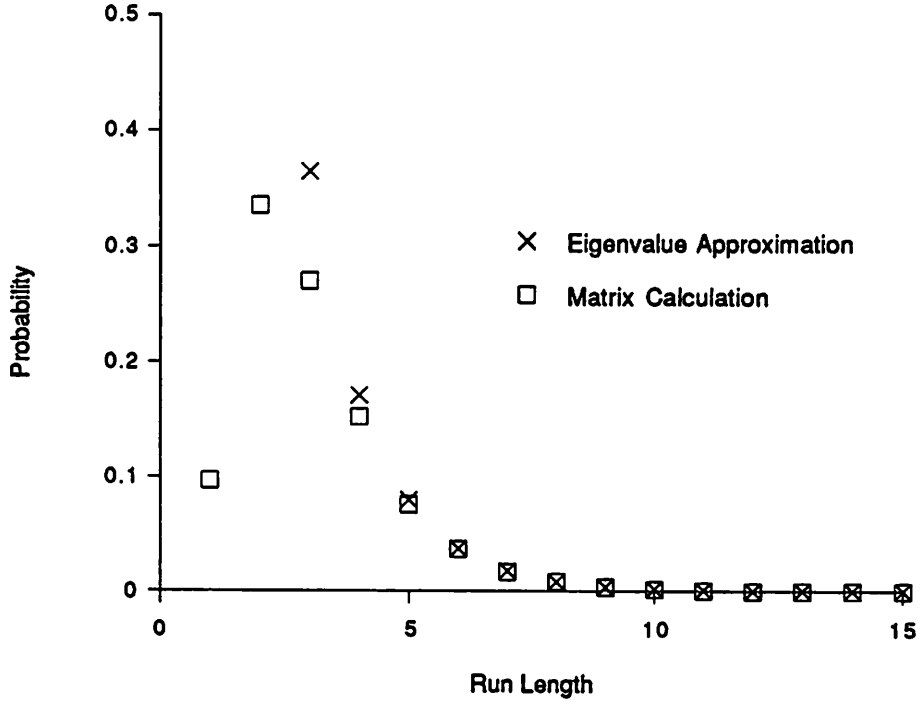


Figure 1b. Run length probability distribution at the operating limit. The value of λ used for the eigenvalue approximation was 0.468733.

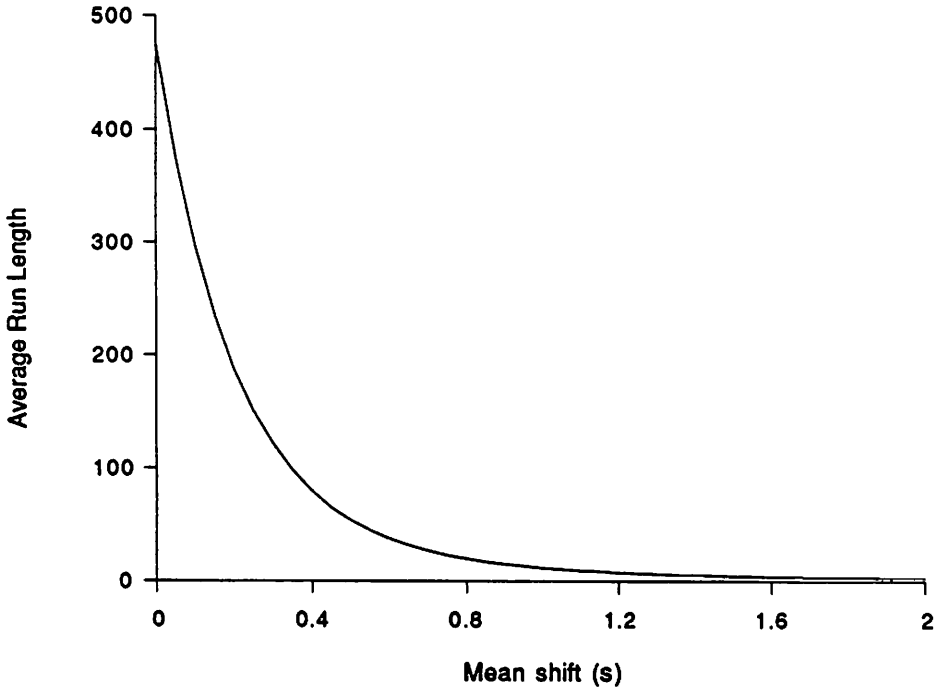


Figure 2. Average run length as a function of the mean shift for the one-sided case.

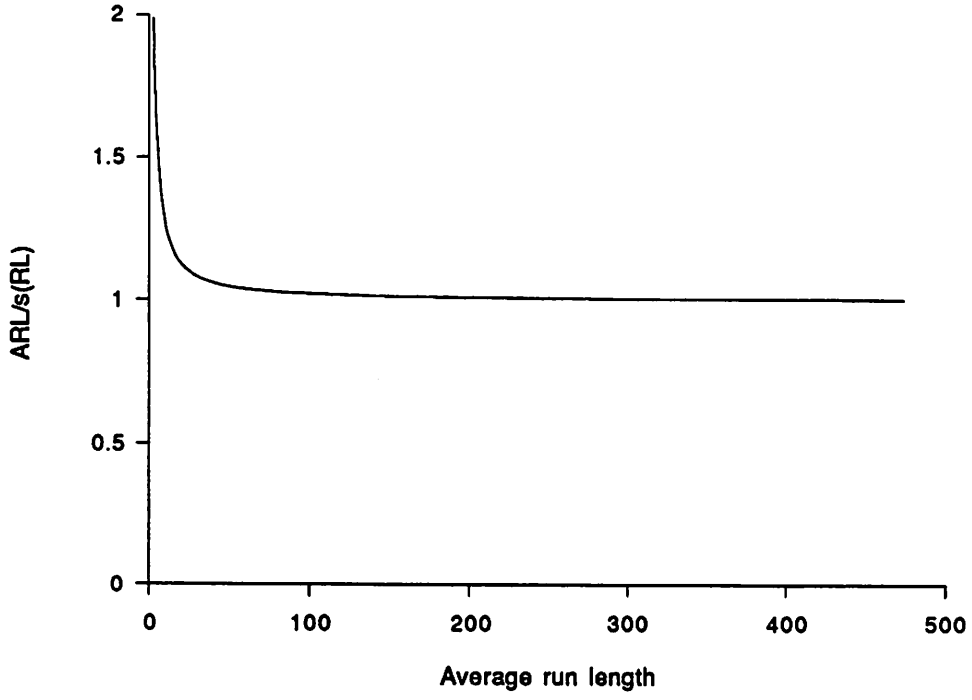


Figure 3. Ratio of average run length to the standard deviation of the run length distribution as a function of run length, for the one-sided case.

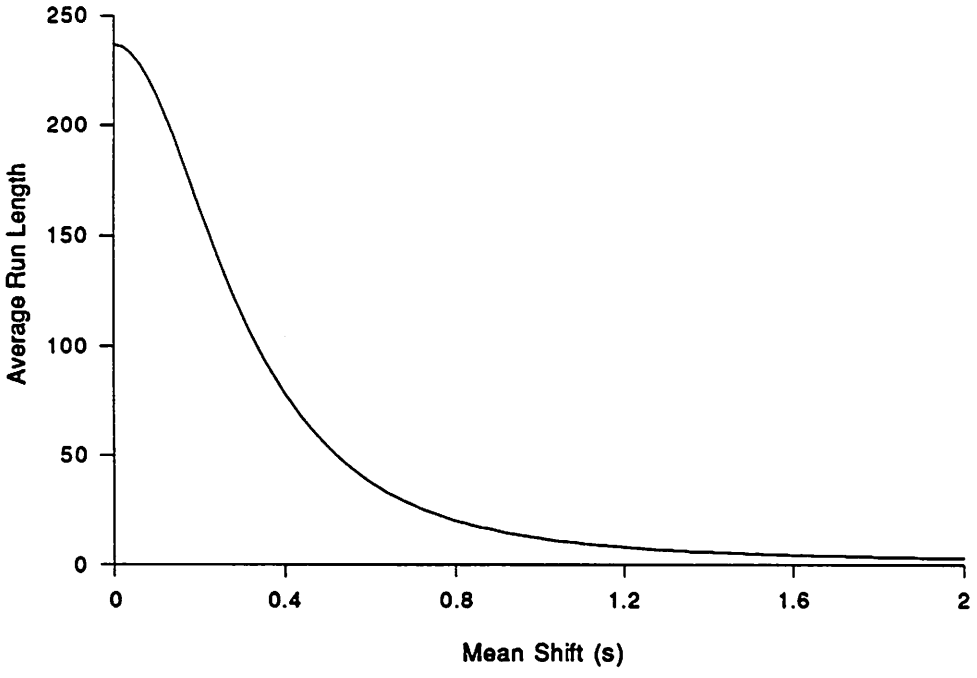


Figure 4. Average run length as a function of mean shift for the two-sided case. The curve is symmetric about x = 0.

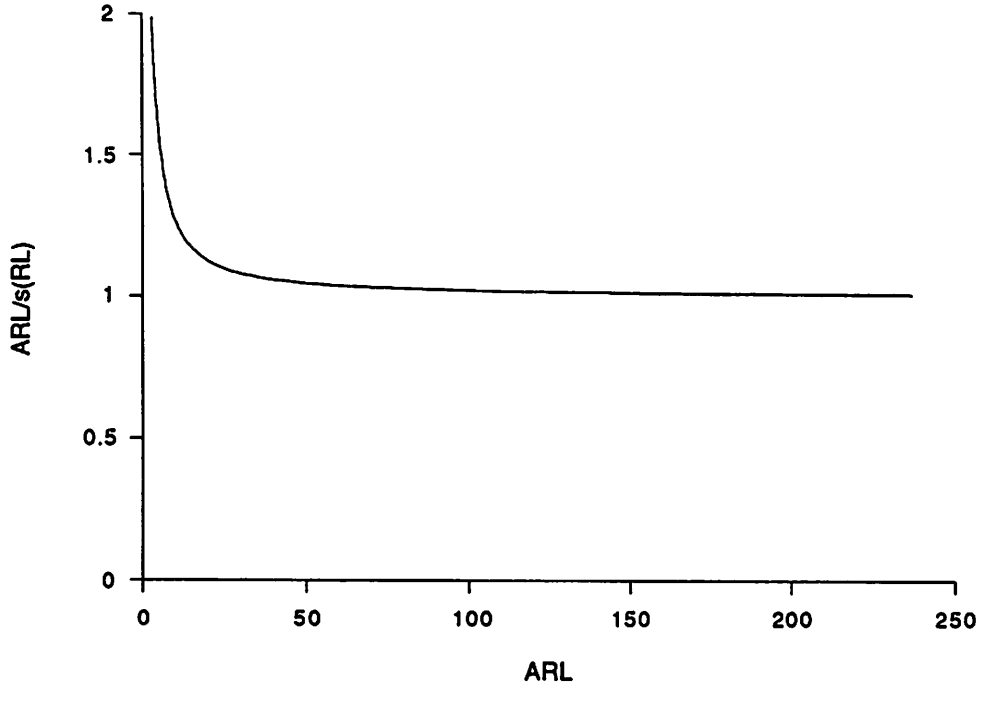


Figure 5. Average run length as a function of the mean shift, for the two-sided case

A Variable Sampling Interval Control Chart Using Runs Rules

Tom Garfinkel

Abstract

Variable sampling interval charts offer significant advantages for processes with costly in-line measurements. However, in order to compete with traditional, fixed sampling interval charts, one must consider control strategies that incorporate a number of runs rules. Here, a program is presented for the simulation of VSI charts with runs rules and it is shown that their sensitivity compares favorably to the equivalent FSI charts.

1. Introduction

Process control charts for variables are widely used throughout industry to maintain product quality standards.This paper discusses the use of a variable sampling interval (VSI) $\overline{X}$ control chart [1] to minimize the average run length (ARL) of a process that has shifted outside acceptable specifications limits. The standard $\overline{X}$ control chart samples at constant intervals, i.e, has a fixed sample interval (FSI). The VSI chart, however, samples at an interval which is dependent on the most recent $\overline{X}$ measurement. Fig. 1 displays the control chart configuration of the VSI chart utilized. An $\overline{X}$ sample which lands within the d_2 interval, on either side of the mean, would produce a sampling interval (SI_2) different from the interval (SI_1) which would occur if the $\overline{X}$ value fell outside d_2 but within the signal limits. The corresponding probability of each sampling interval are abbreviated P_2 and P_1. When the process shifts away from the desired mean, P_1 increases and SI_1 occurs more frequently. Thus the out of control VSI ARL can be reduced, relative to the FSI, by making SI_1 less than the sampling time interval associated with the FSI chart.

There are several additional considerations necessary to design an optimal VSI chart for a given manufacturing application. Knowledge about process yield, loss in revenue due to machine down time, actual sampling costs, and other economic implications will all effect the process control implemented. Finally, to minimize the ARL of a VSI chart, one can apply runs rules. Instead of identifying an out of control process by a single $\overline{X}$ reading beyond the upper or lower control limits (UCL,LCL), a combination(s) of measurements exceeding set limits, .e.g., three consecutive $\overline{X}$ values larger than UCL can be used. An optimal runs rules VSI chart can thus be developed for a specific process control situation.

2. Description

The situations studied here are FSI and VSI charts with shifts in the process mean. The most common $\overline{X}$ chart, ARL=1/β, is for a FSI chart where the probability of exceeding UCL is β. This paper uses a closed form approximation to the normal distribution to calculate these probabilities [2]:

$$P(x) = 0.5 + 0.5(1 - e^{2x\pi})^{1/2} \tag{1}$$

This apporoximation applies for all x and is quite accurate (see Fig. 2). The FSI ARL can therefore easily be characterized as a function of the mean shift. The ARL of a VSI chart, where one $\overline{X}$ beyond UCL is used as a signal, can be calculated from the following equation [1]:

$$ARL = \sum tP(t)\beta \tag{2}$$

where t = time, and P(t) is the probability of sampling at time t. With the FSI chart the sampling times are constant and known in advance. VSI charts, due to SI_1 and SI_2, can sample at any time which is an integral combination of the SI_1 and SI_2 intervals. Physically P(t) is therefore just a permutation of the number of ways the time t can be reached, multiplied by the probability of each occurrence. It is formulated below:

$$P(t) = \frac{1}{(1-\beta)} \sum \binom{(t-(SI_2-1)r)/SI_1}{r} P_1^{(t-SIr)/SI_1} P_2^r \tag{3}$$

where r refers to the number of SI_2 steps that could have occurred prior to the sample in question. The algorithm used to implement these equations is shown in the program section (#1).

A plot of the ARL versus shift in mean (measured in standard deviations σ from an in control $\overline{X}$) is shown in Fig. 3. The details of the control charts are displayed in the upper right of the plot. "Consec" describes the number of readings beyond the UCL necessary to indicate the process is out of control. All of the symbols except for the one followed by FSI correspond to VSI data. The two numbers describing each VSI symbol are SI_2 and SI_1 respectively. The data is plotted on a semi-logarithmic scale in order to include all points, not to downplay the significance of small shifts. Normally distributed data, using P(x), was subjected to the process control limits and agreed reasonably well with theoretical calculations. The program to generate and test the normal data is also included (#2). Although the VSI shows a significant ARL advantage over the FSI, the VSI chart can still be improved upon by applying runs rules.

The algorithms to perform FSI runs rules are just extensions of the simple Bernoulli summation used to derive the relationship ARL = $1/\beta$. A generalized program (#3) performs different runs rules calculations based on these summations. To illustrate this algorithm the runs rule where n of n-1 measurements must exceed UCL to signal a problem is briefly discussed. At each sample, starting with sample n-1, the number of ways of getting at least n-1 readings beyond UCL, multiplied by the probability of each, will give the probability of a signal occurring at a sample. Each term is then weighted by the cumulative probability that none of the previous samples produced the signal. The VSI chart can again be approximated, and the results are discussed in the following section. One additional program (#4) was written to study a situation not previously mentioned.

All of the cases discussed assume that the process shift occurs at a known time. A more accurate model depicts the process at the appropriate mean until, randomly, a shift takes place. A modification to the summations outlined was performed to account for this effect. A more rigorous derivation is performed in [1].

3. Results

In Figs. 4-8 the FSI and VSI plots of ARL vs $\overline{X}$ shift are shown for different runs rules. Each of these plots has the ARL(in control) matched to 370, the FSI result when UCL=3. Thus the number of false alarms, on average, is no larger for the in control VSI charts. The plots were reproduced for the case where the process is initially in control before shifting No significant change was observed for any of the VSI charts. This disagrees with the result in [1] for very large SI_2, and is attributed to the less complex algorithm implemented here. Both the values of SI_2 and SI_1 vary within a given plot, but Fig. 9 can be used to help distinguish their effects. The effect of increasing SI_2 is shown to saturate quickly; Fig. 4-8 are therefore dominated by the SI_1 value. Since, ultimately, generating the minimal ARL is of interest, Fig. 10 re-plots the lowest ARL curves. Note that a smaller SI_1 could have been chosen to reduce the ARL even more.

In Fig. 10 it is immediately apparent that different runs rules prevail, or possess the lowest ARL, in different shift regions. Recall that VSI's are matched when the process is in control, and therefore, with identical SI_2 and SI_1 in all curves, have variable UCL values. The 8/8 curve is clearly superior for small shifts, and offers the largest possible improvement. As the shift reaches 3σ, however, the ARL, like all the other VSI curves, approaches the product of its SI_1 value and the number of consecutive samples required (0.8). Changing the desired ARL(in control) will move the "crossover" points at which the different runs rules prevail (Fig. 11). Thus, for a specific process where the economic implications as a function of process shift are known, it is possible to select a runs rules scheme which is most suitable. A linear combination could, of course, be utilized to balance the strengths of different runs rules. One trivial example might incorporate both the 8/8 and 2/2 consecutive rules while maintaining a satisfactory alarm rate. Thus, if either runs rule condition were reached, the process would be considered out of control. This compromise would decrease the advantage obtained by using the 8/8 alone for small shifts, but in a case where larger shifts are unacceptable financially it would improve the chart.

4. Conclusions

The VSI $\overline{X}$ control chart can significantly reduce the average run length for a process that shifts beyond acceptable limits. Minimizing SI_1 will produce the most dramatic decrease in ARL, without raising the probability of false signals. The application of runs rules may optimize a VSI chart for a well characterized operation by selecting the most suitable curve of ARL vs $\overline{X}$ shift. A combination of these rules may best satisfy the process control demands for production.

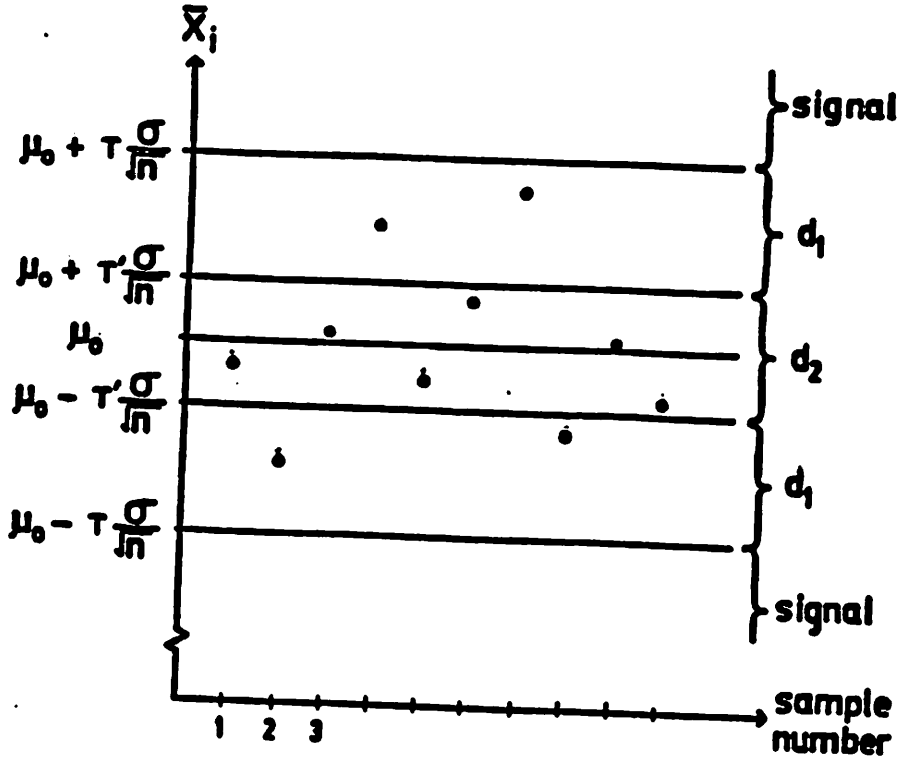


Figure 1. $\bar{X}$ Chart With Variable Sampling Intervals.

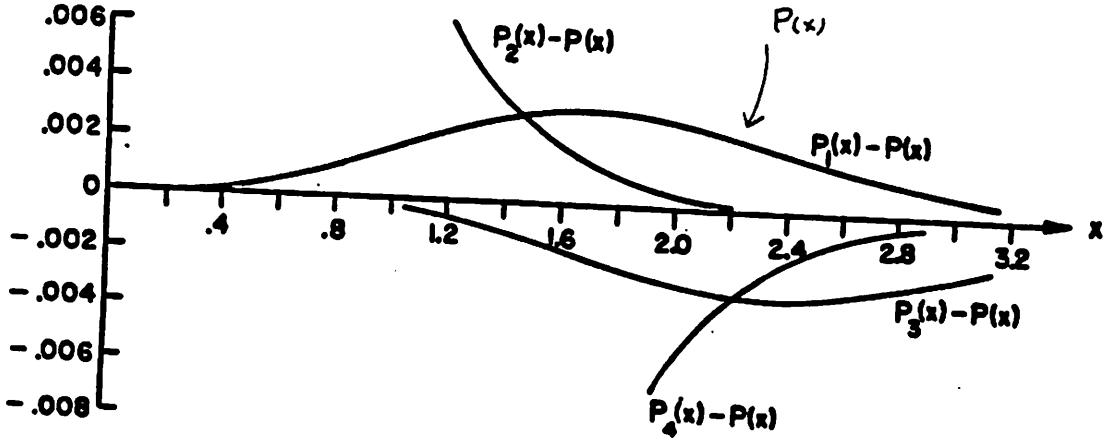


Fig. 2. *Error curves for bounds on normal distribution.*

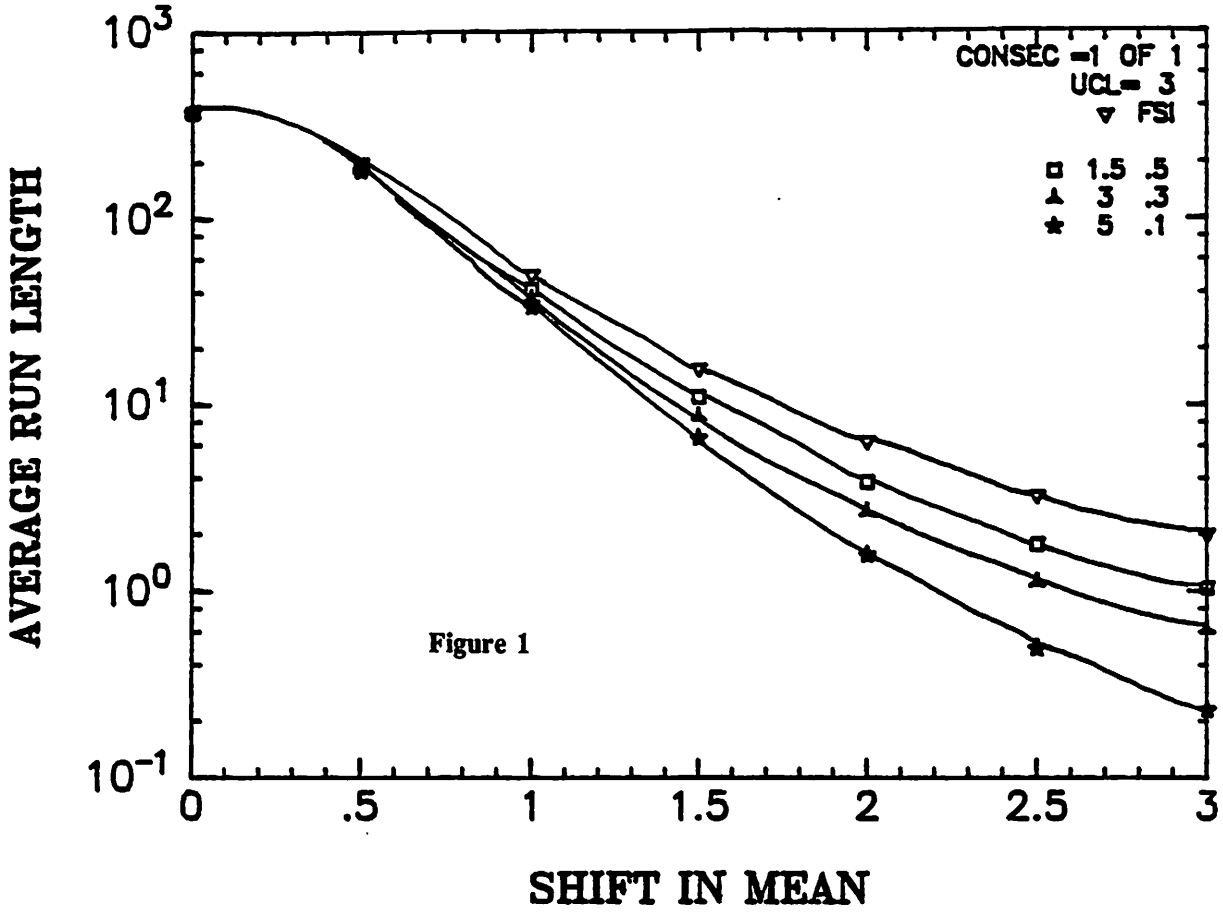


Figure 1

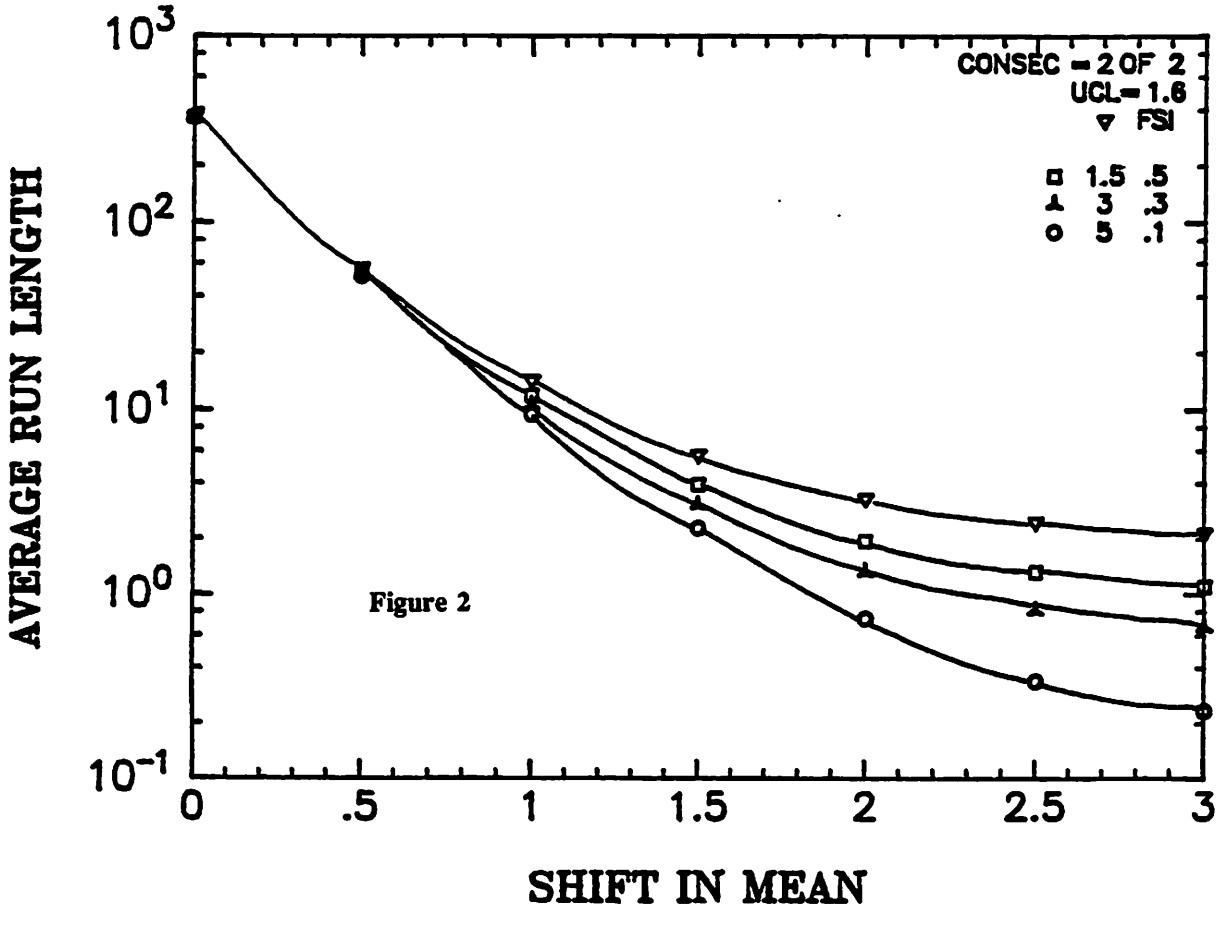


Figure 2

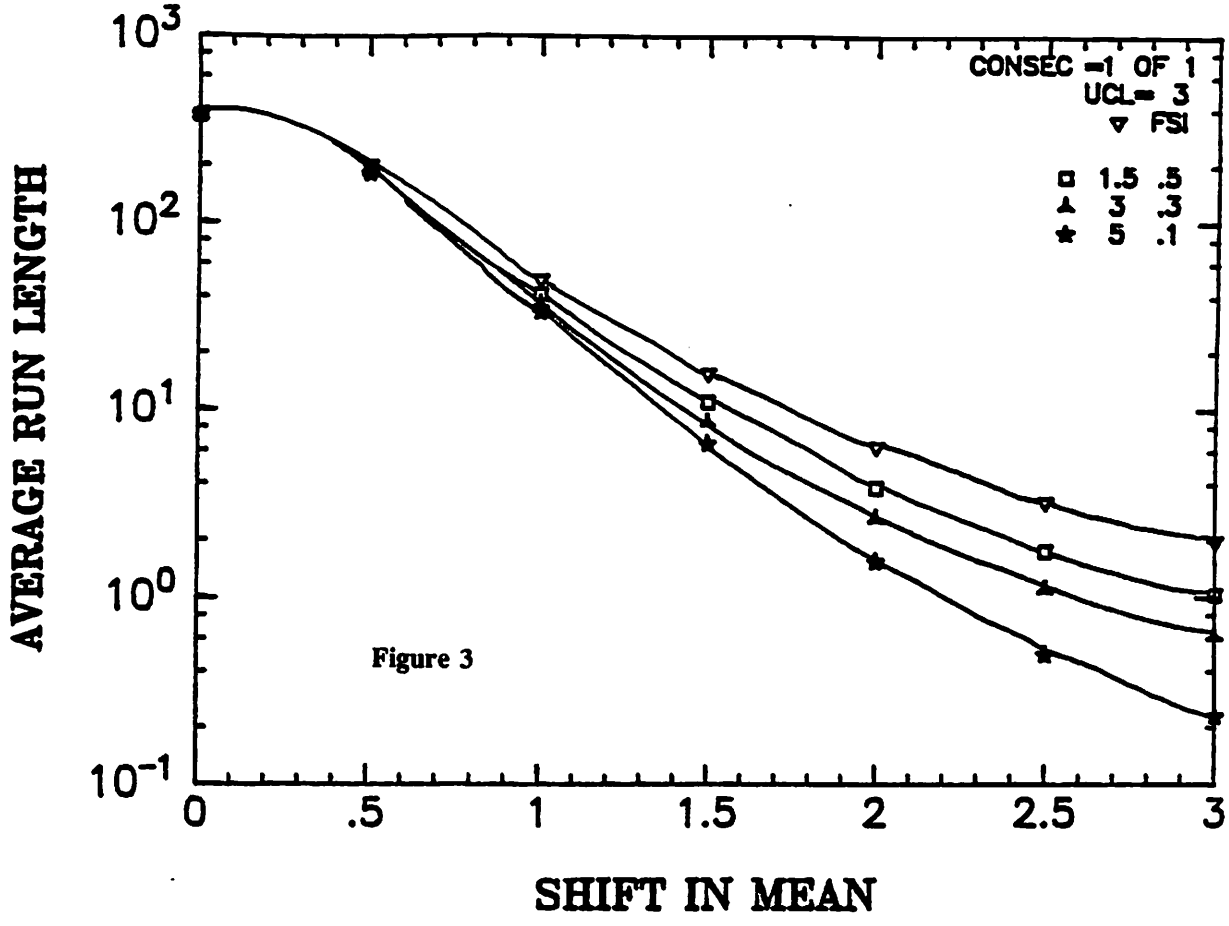


Figure 3

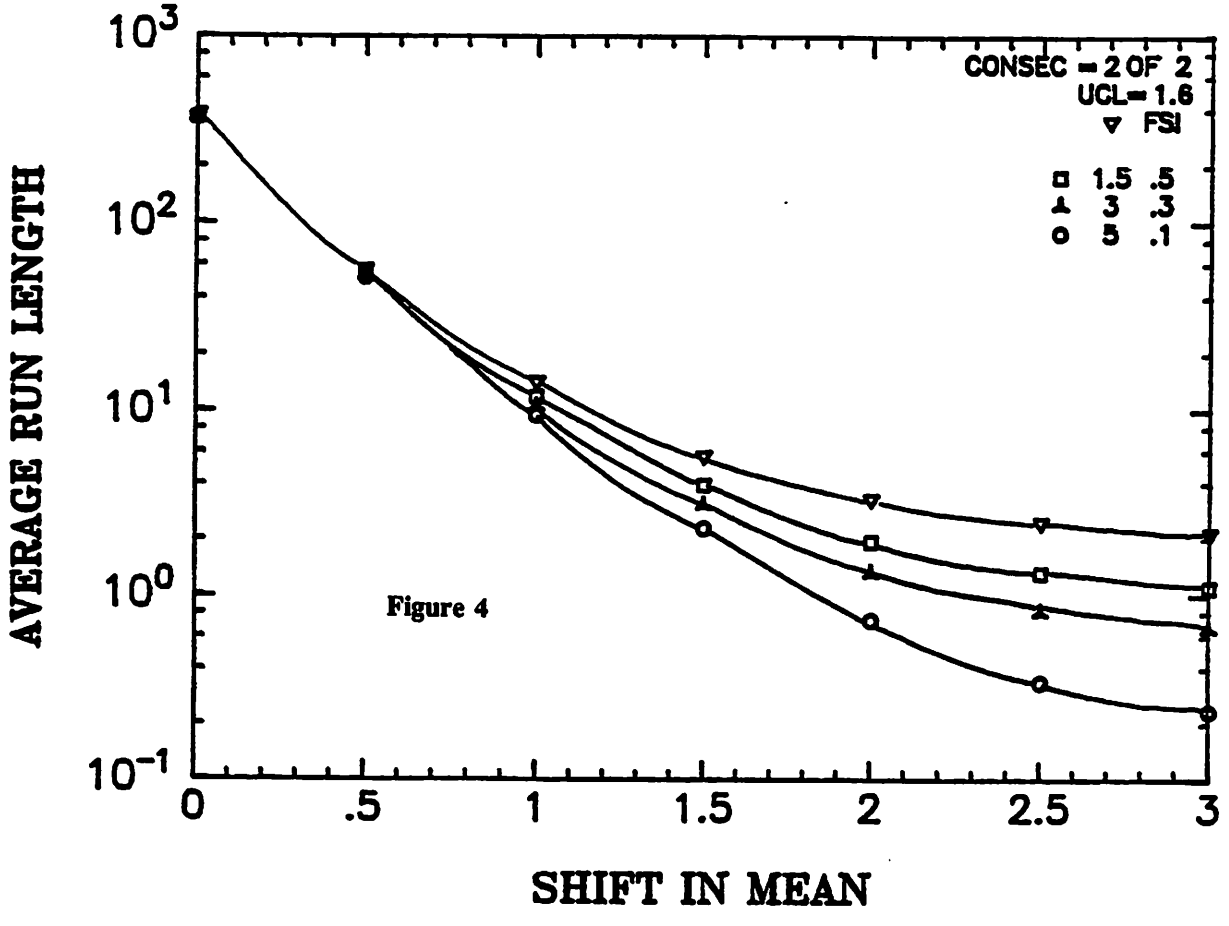


Figure 4

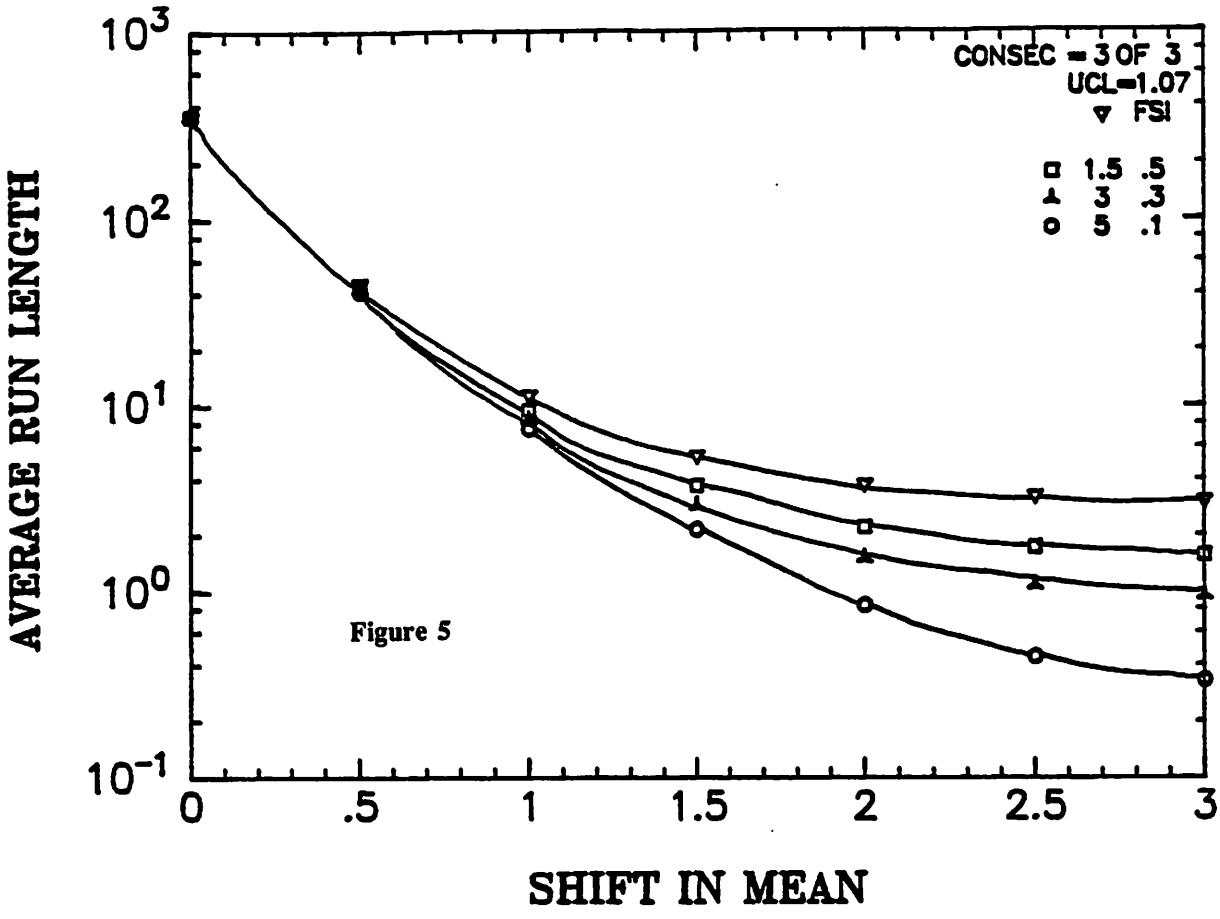


Figure 5

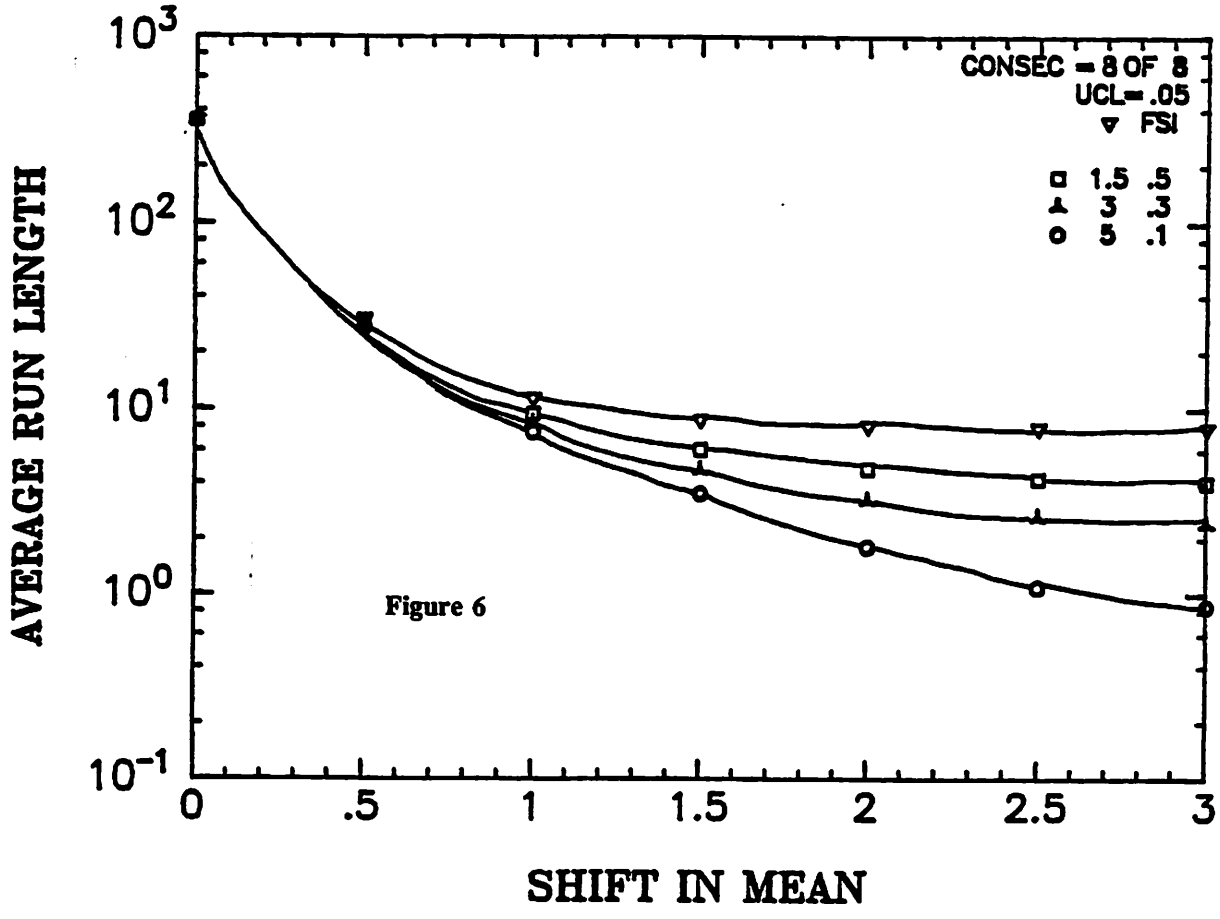


Figure 6

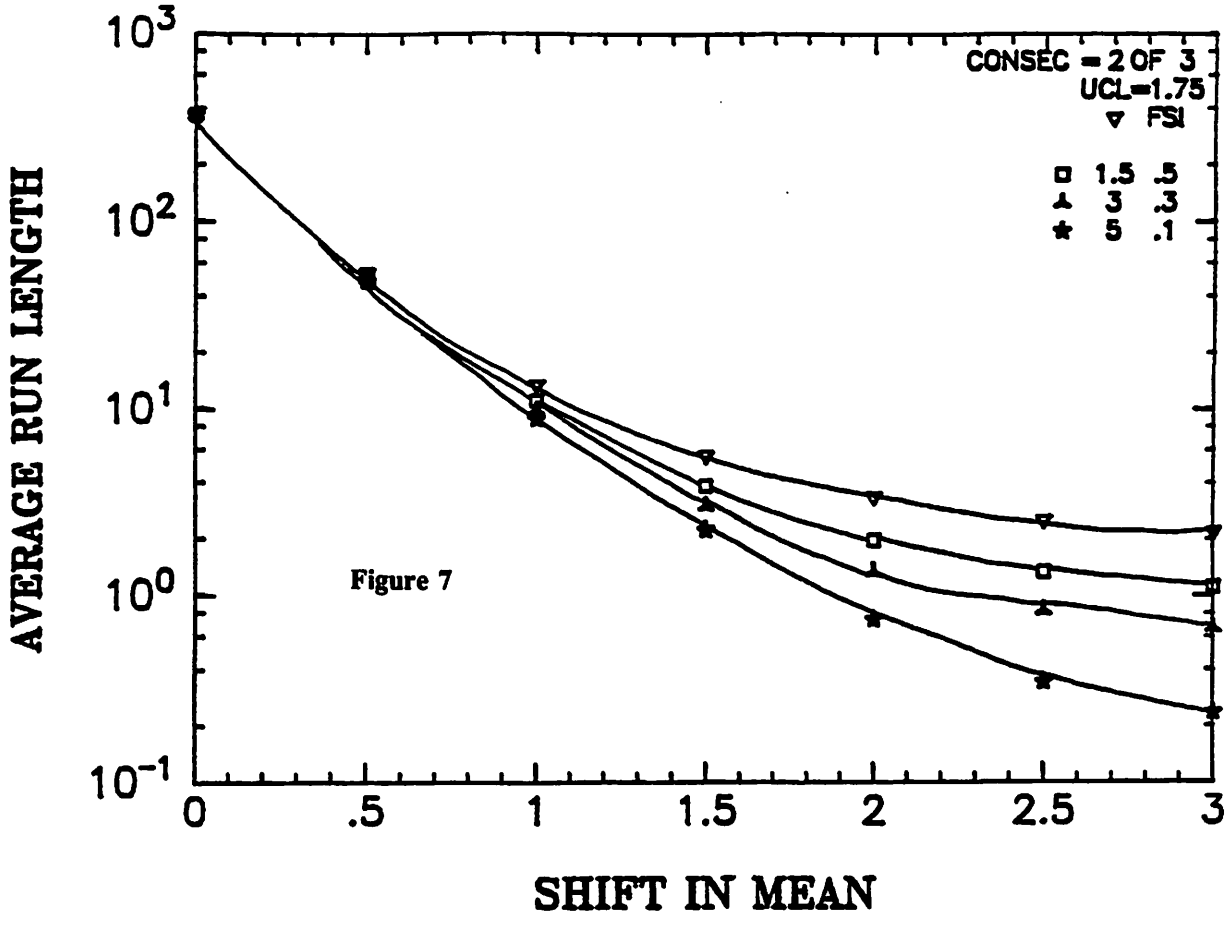


Figure 7

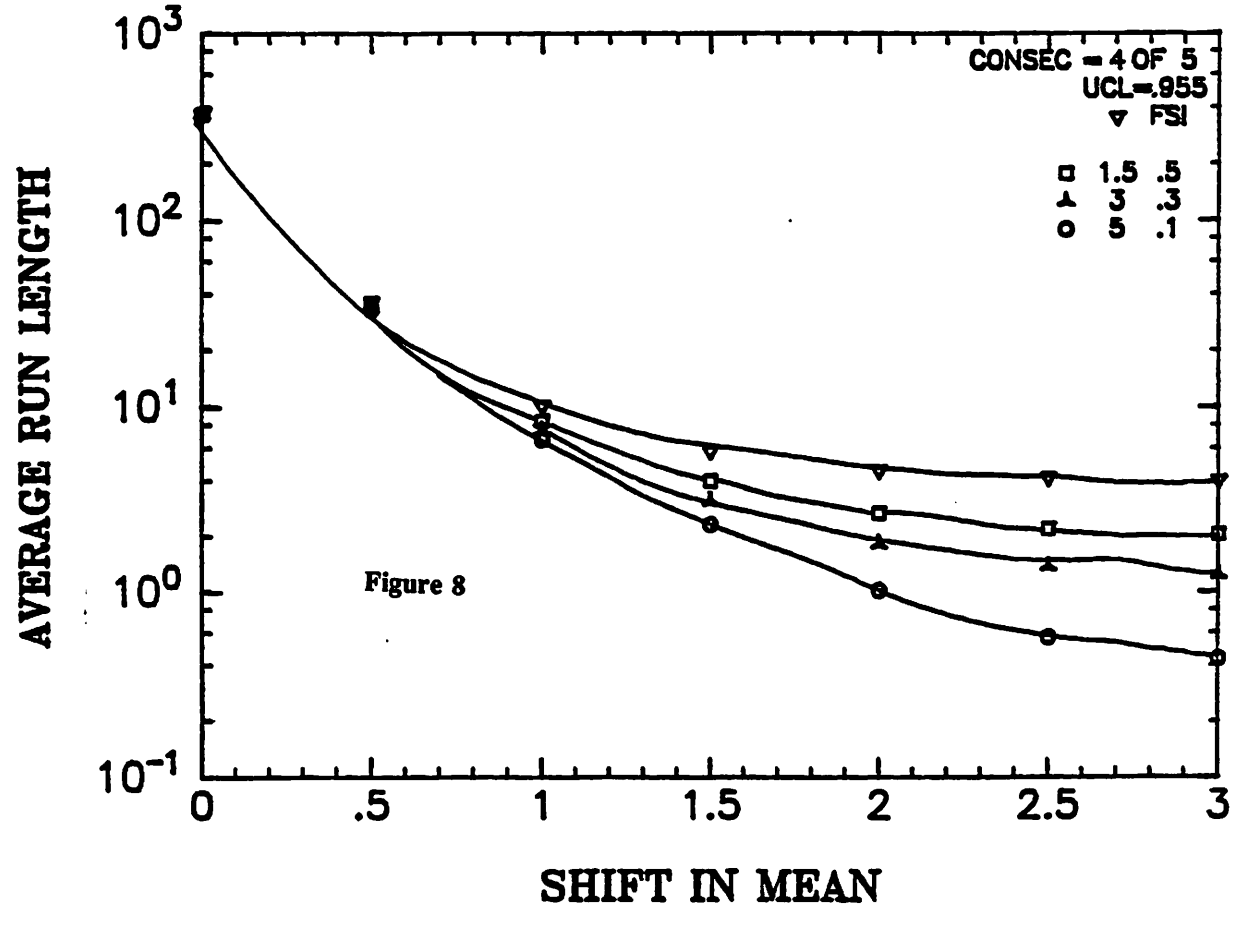


Figure 8

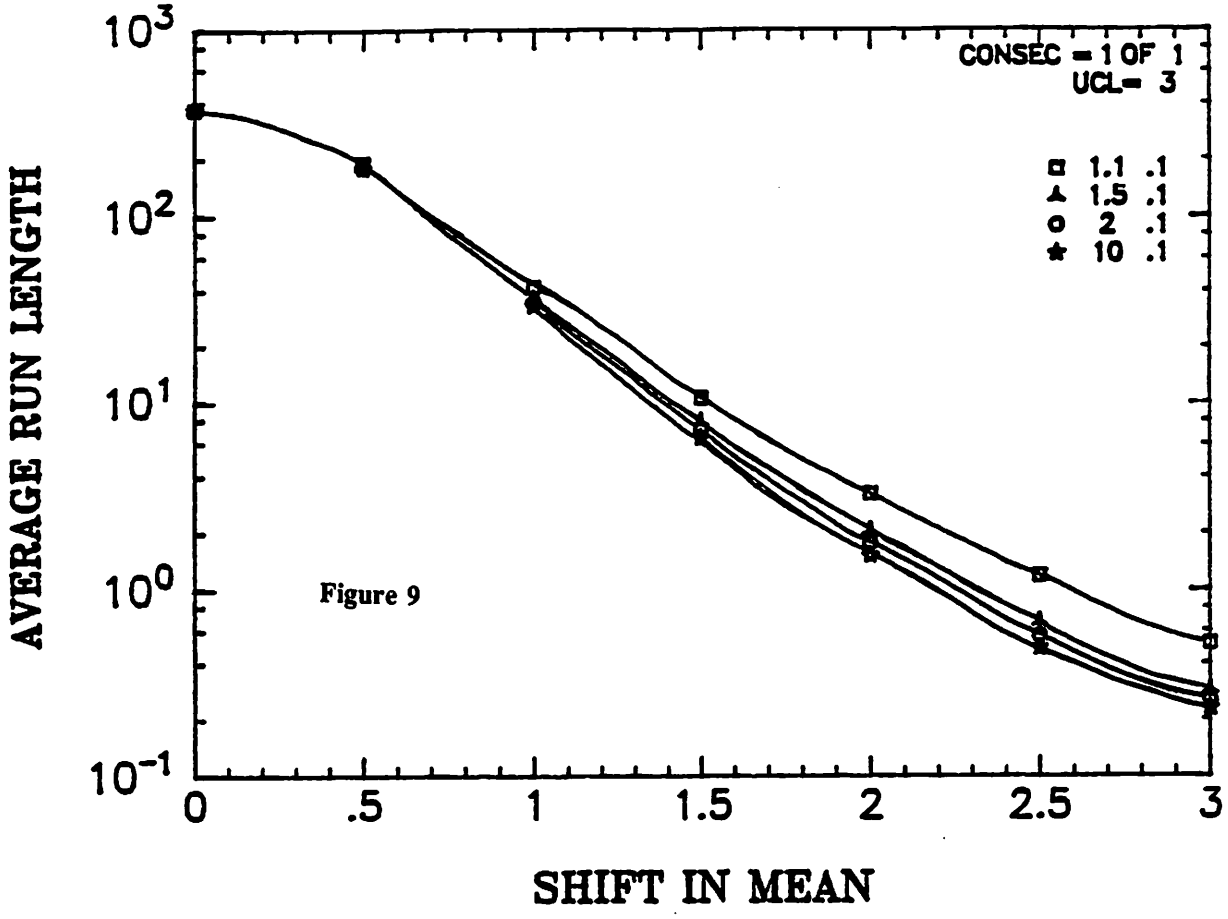


Figure 9

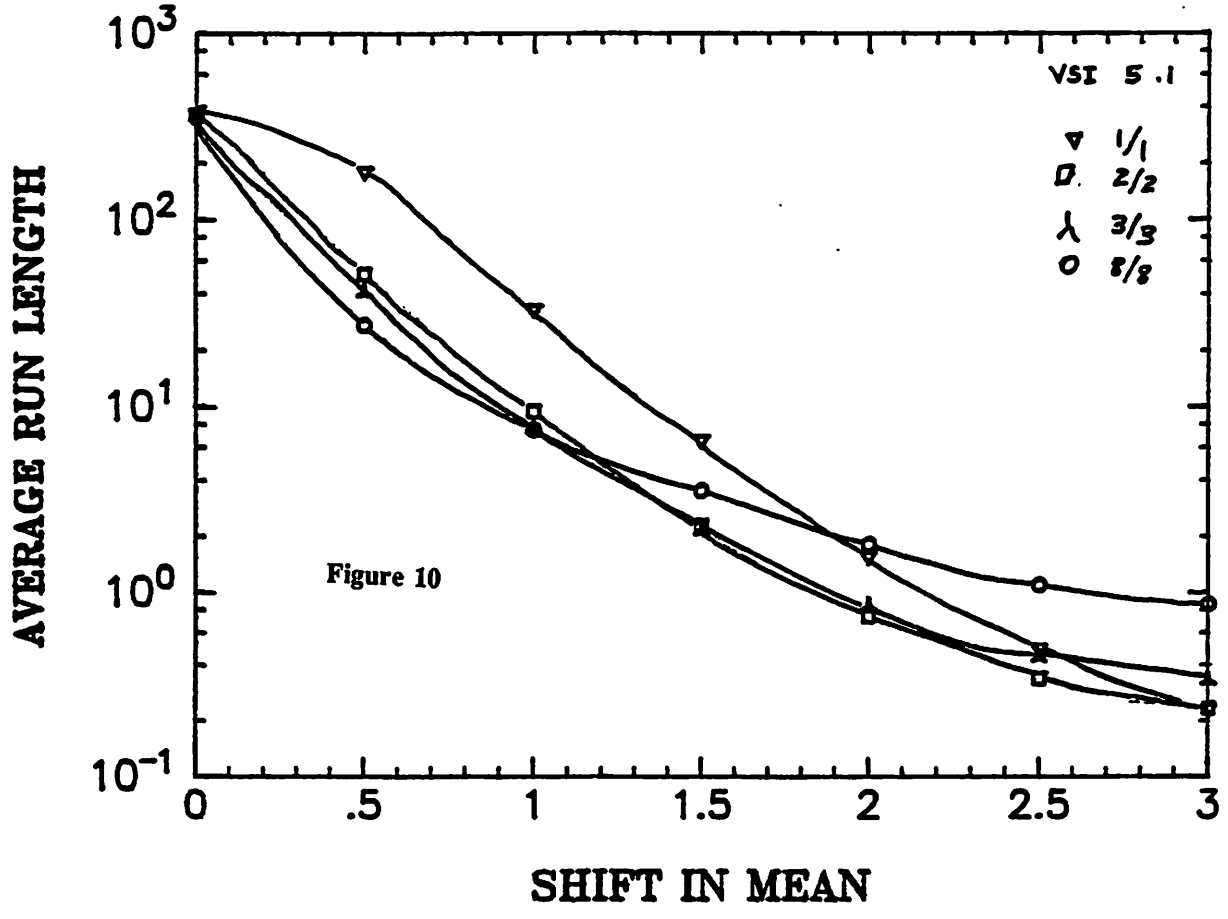


Figure 10

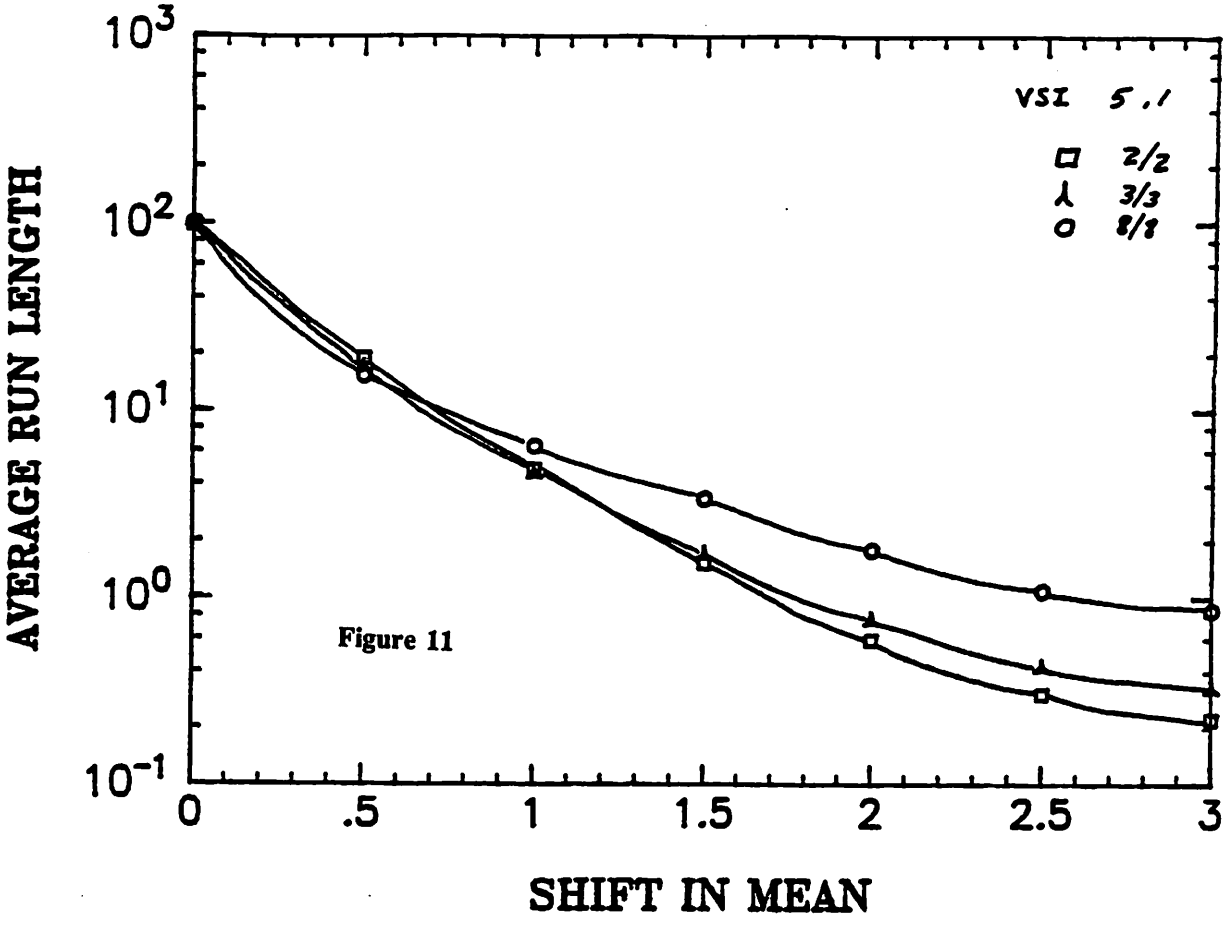

AVERAGE RUN LENGTH
10^3
10^2
10^1
10^0
10^{-1}
0
.5
1
1.5
2
2.5
3
SHIFT IN MEAN
VSI 5.1
□ 2/2
λ 3/3
○ 8/8

Figure 11

$\bar{X}$ Chart with Variable Sampling Interval for the Control of a Photolithography Process

Jaime Ramírez

Abstract

A variable sampling interval chart has the advantage of increasing its sampling rate in anticipation of an out of control condition, while it regularly maintains a slower, more economical sampling rate when the process appears to be operating properly. Given the costs associated with routine, in-line wafer testing in VLSI manufacturing, it is shown here that a VSI scheme applied on a photolithogrphic operation can offer significant advantages over a traditional, fixed sampling interval control strategy.

1. Introduction

Control charts are used to monitor processes, and many different types and variations exist [1-6]. The Shewhart $\bar{X}$ control charts are very common in the U.S., and several modifications to the original chart have been suggested to improve its performance; some of these are: warning limits [1,2], supplementary runs rules [3], and variable sampling intervals [4-6]. By incorporating the variable sampling interval (VSI) which was proposed by M. R. Reynolds [4], the efficiency of the chart is greatly increased.

The basic principle behind the VSI chart is that if the process seems to be in control, then sampling should be done at a lower frequency, but if the process seems to be running out of control, then the sampling rate should be increased in order to detect with greater accuracy when the process runs out of control. By detecting with greater accuracy when the process runs out of control, the amount of material that needs to be tested in order to ensure its quality (ie.- that between the previous sample and the one that produced the signal) is smaller, and since less testing is needed, the operating costs are reduced. The cost advantages of using the VSI $\bar{X}$ control chart for monitoring a process have not been pointed out, and this is what is investigated in this paper. In particular, an example of a photolithography process is taken, and some numerical values have been calculated for the costs involved at a first order. The cost of implementing a VSI scheme is compared to that of implementing a FSI one.

2. Methodology

It is necessary to have a thorough understanding of how the VSI scheme applied to the $\bar{X}$ chart works. After reviewing the available articles which are related to this subject, an attempt to replicate some of the important published results will be made. The software package BLSS will be used to compute all of the probabilities and statistics necessary to obtain some of the results which have been published by Reynolds. These results will be used to promote the improvements which result by varying the sampling time. Once an understanding of the concept behind the VSI $\bar{X}$ chart is obtained, the cost factor will be integrated. A general cost equation will be formulated and compared to that of the FSI $\bar{X}$ chart. Several factors need to be considered in this cost equation, and an analysis to determine the most important ones will be followed. Finally, some values representative of the costs involved in a lithography process will be incorporated into these equations, and the cost for several cases for both FSI and VSI $\bar{X}$ charts will be compared.

3. Implementation

The basic idea behind the VSI chart is that one should sample with a high frequency when the process is close to being out of control, and if the process seems to be in control then the sampling frequency should be low. It has been determined by several authors that the optimal number of sampling intervals is two, which helps maintain the complexity of the chart at a low level. It has also been proved that the best choice is to have one interval be as large as possible and one as small as possible. When comparing the FSI and VSI charts, it is necessary to normalize them by picking one unit of time as the base, and having it be the expected value of the sampling interval when $\mu=\mu_o$ for both charts. In reality, since both charts have the same control limits, the average number of samples to signal (ANSS) is always the same, so the expected value of the sampling interval is the one that determines the

average time to signal (ATS), which is the parameter used to compare both types of charts.

Although the average number of samples to signal is the same, this does not mean that the sampling costs will not play a role in the cost equation. This is because even though both charts test the same number of samples, the VSI method tests them in a shorter period of time because it detects the out of control signal quicker. Therefore, for a large fixed interval of time, the VSI method takes more samples. The advantage however is that since the out of control signal is detected quicker, less low quality products will be shipped or will need to be tested; therefore a reduction in sample testing is obtained together with an increase in quality, both of which are good.

4. Results

BLSS was used to calculate the probabilities corresponding to the particular intervals that were chosen setting the condition that E(Ri)=1 when $\mu=\mu_o$ in order to normalize all the information so that a true comparison may be made. From these probabilities, the q-value (gamma') was obtained. This is the value where an imaginary line is drawn on the $\bar{X}$ chart to indicate the separation between the regions where the different sampling rates will occur. In a way, this is like having a warning line which tell us to sample in a quicker fashion because it is more probable that the process is running out of control due to a shift in the process mean. After this "warning line" is found, we divide the region into sections corresponding to those of the sampling intervals, and the probability of being in these sections for given shifts in the mean is calculated. Then we have a table with the ATS values for the different shifts in the process mean and for different choices of sampling intervals. A few more calculations are necessary in order to calculate the adjusted ATS, which correspond to the case when the change in the mean occurs between two samples.

The BLSS input files and output files are found in Appendix A. The first output listing replicates the results found in [4], and the second listing investigates the effects of having a fixed long sampling interval and various short sampling intervals (asymmetric cases), and the effects of very small intervals with very large ones while maintaining symmetry. The results agree with what was postulated by Reynolds in [4], and the best choice of sampling intervals is to have them as far apart as possible (ie.- a very short interval and a very long one).

5. Examples

In applying the VSI $\bar{X}$ chart to a photolithography process, some information regarding the cost of the equipment, testing time, type of testing to be done, wait time, etc. need to be investigated. When working in a photolithography process, one must be aware that there are many parameters which need to be monitored, and each one will be affected only by certain equipment, chemicals, processing, etc.

Consider the case of monitoring the thickness of the photoresist prior to exposure. The resist thickness plays an extremely crucial part in the resolution of a system because of the linewidth variations which result. There are many ways to monitor the thickness of the resist after it has been spined on: the resist thickness can be measured using and ellipsometer, or it can be measured after exposure by measuring the reflectivity (ie.- indirectly measuring the amount of bleach present in the resist), etc. The reflectivity measurement can be done in situ, and an autoexposure may be possible. I will assume an ellipsometer is used, and assuming it takes approximately 3 minutes to measure a wafer, the testing cost is roughly $1 (assuming an equipment cost of $100K run continuously during 3 shifts, and the cost of labor is $10/hr).

If the resist thickness is too large, then it may be possible to bake the resist to remove some of the material (although some of its photo-chemical properties will be slightly altered), or the resist can be removed and re-deposited. I will assume that the resist is removed and re-deposited. The cost of re-working a wafer can be calculated to be about $4 if the cost for processing a 4" wafer is about $400-$500 and 120 processing steps are needed (15-20 lithography steps), and I assume the cost is distributed proportionately. (The costs are all estimates, and are intended to be used just for the purpose of illustration.)

There are usually 20-24 wafers per batch. If 1000 wafers per week are processed in batches of 20 wafers, then at least 4 wafer steppers would be needed. If 4 wafer steppers are used, then 250

wafers pass through each, or the equivalent of about 13 batches in a week, or 2 batches per day, running the 20 processing steps on each. It is clear that in reality all of the equipment is synchronized so as to have the wafers "flow" through them, in this case, the equivalent of 40 batches run per day on each stepper. I will assume that the standard sampling interval is one sample every 5 batches. If we set the VSI chart to have the following two sampling intervals: d1=0.2 (ie.- sample each batch), and d2=1.6 (ie.- sample one batch for every 8), then we will obtain ATS values which are slightly better than those found for (0.2,1.5) in the second sample of intervals in Appendix A.

If the process is in control, but we get a false alarm: in the FSI method, we need to inspect the last 5 batches and rework the necessary wafers. If we assume that half the wafers need to be reworked, then the cost for inspecting and reworking these wafers is \$300 (\$100 from inspection and \$200 form rework); in the VSI method, we would detect the error with a small sampling interval, and therefore only one batch would need to be inspected and half of those wafers would need to be reworked, at an expense of only \$60. Since the process is in control, the ATS is equal for both, and therefore in the same period of time they both only detect one error.

If a shift of 1.0 occurs, then for the FSI method, it would take on the average 43.89 standard time units to signal (ie.- after 43.89*5 batches are processed). For the VSI method, it would only take on the average about 30 (<31.53) std. time units to signal. In that period of time, the same number of samples would have been taken on the average for both methods, therefore in a period of 43.89 standard time units, the VSI method would have sampled at most 1.463 times as many batches. In a period of 43.89 std. time units, 43.89 samples were taken on the average, at a cost of \$43.89. Multiplying \$43.89 by 1.463 roughly give us 64.21 samples, or approximately \$64.21. Therefore the total expense for the FSI method on one time period when a shift of 1.0 occurs is roughly \$344, whereas for the VSI method it is only \$125 (60+65) (assuming the problem which caused the process to go out of control was fixed, otherwise the cost would have been 1.463(60+44) = 152. The advantages of the VSI method are clear.

Since the cost of re-processing a wafer is larger than testing it, and since the batches contain many wafers, the VSI method proves to be very good from an economic, as well as statistical, standpoint.

6. Conclusions

The VSI $\bar{X}$ control chart is much more effective than its FSI counterpart in both detecting the shifts in a shorter period of time, and reducing the cost which is incurred when the process is out of control. It is very important to accompany this chart with an R chart which monitors the variance of the process; an $\bar{X}$ chart alone does not serve its purpose.

The cost incurred when the process is out of control is the following:
For the FSI chart:

$$\text{COST} = (\text{ATS}) \times (\text{sampling cost}) + (\text{batch size}) \times \frac{\text{rework cost}}{2}$$

For the VSI chart:

$$\text{COST} = (\text{ATSfixed}^2) \times \frac{\text{sampling cost}}{\text{ATSvariable}} + (\text{batch size}) \times \frac{\text{rework cost}}{2}$$

As can be observed, the cost is composed of two costs: a fixed cost (second term), and a variable cost. This equation can be maximized for any given costs.

References

[1] E. S. Page, "Control Charts With Warning Lines," *Biometrika*, Vol. 42, pp. 243-257, 1955.

[2] J. I. Weindling, S. B. Littauer, and J. T. de Oliveira, "Mean Action Time of the $\bar{X}$ Control Chart with Warning Limits," *Journal of Quality Technology*, Vol. 2, No. 2, pp. 79-85, April 1970.

[3] C. W. Champ and W. H. Woodall, "Exact Rules for Shewhart Control Charts With Supplementary Runs Rules," *Technometrics*, Vol. 29, No. 4, pp. 393-399, November 1987.

[4] M. R. Reynolds, Jr., R. W. Amin, J, C. Arnold, and J. A. Nachlas, "$\bar{X}$ Charts With Variable Sampling Intervals," *Technometrics*, Vol. 30, No. 2, pp. 181-192, May 1988.

[5] M. R. Reynolds, Jr. and J. C. Arnold, "Optimal One-Sided Shewhart Control Charts With Variable Sampling Intervals," *Sequential Analysis*, 8(1), pp. 51-77, 1989.

[6] M. R. Reynolds, Jr., "Optimal Variable Sampling Interval Control Charts," *Sequential Analysis*, 8(4), 1989 (In press).

Appendix A

BLSS output which contains both input commands and output. Some comments have been added to simplify the understanding of the data, and to make it consistent with [4].

```
. . d1=0.0,.5,.3,.1,.1,.1,.1,.1
. . d1=d1'
. . d2=1.0,1.5,1.7,1.9,1.1,1.3,1.5,4.0
. . d2=d2'
. . q0=.0027
. . p01=(d2-1)*(1-q0)/(d2-d1)
. . p02=(1-d1)*(1-q0)/(d2-d1)
. . p=(1-p02)/2+p02
. . qgau p > gamma
. . show d1,d2,p01,p02,gamma
```

	d1	d2	Po1	Po2	gamma'
A	0.000	1.000	0.000	0.9973	3.000
B	0.5000	1.500	0.4987	0.4987	0.6724
C	0.3000	1.700	0.4986	0.4986	0.6724
D	0.1000	1.900	0.4987	0.4987	0.6724
E	0.1000	1.100	0.09973	0.8976	1.633
F	0.1000	1.300	0.2493	0.7480	1.145
G	0.1000	1.500	0.3562	0.6411	0.9175
H	0.1000	4.000	0.7672	0.2301	0.2926

```
. . x1=0.0,0.2,0.5,0.7,1.0,1.2,1.5,2.0,2.5,3.0,4.0
. . gup=gamma-x1
. . glow=-gamma-x1
. . up=3-x1
. . low=-3-x1
. . pgau up > pup1
. . pgau gup > pup2
. . pgau glow > pdown2
. . pgau low > pdown1
. . p2=pup2-pdown2
. . p1=pup1-pdown1-p2
. . percent=(d1*p1+d2*p2)/(p1+p2)
. . ats=percent/(1-p1-p2)
. . ats=ats'
. . show (shape) ats
```

ATS

	FSI	symmetric			asymmetric			
shift	d=1	(.5,1.5)	(.3,1.7)	(.1,1.9)	(.1,1.1)	(.1,1.3)	(.1,1.5)	(.1,4.0)
0	370.39	370.39	370.39	370.39	370.39	370.39	370.39	370.39
.2	308.43	305.89	304.87	303.85	306.46	305.08	304.44	303.24
.5	155.22	147.59	144.53	141.48	149.10	145.02	143.19	139.71
.7	92.32	83.87	80.48	77.10	85.28	80.86	78.90	75.25
1	43.89	36.52	33.57	30.62	37.29	33.60	32.03	29.19
1.2	27.82	21.65	19.19	16.72	21.98	19.01	17.80	15.65
1.5	14.97	10.52	8.735	6.954	10.35	8.375	7.611	6.320
2	6.303	3.814	2.818	1.822	3.303	2.388	2.075	1.593
2.5	3.241	1.775	1.189	0.6029	1.229	0.8171	0.6946	0.5257
3	2.000	1.039	0.6551	0.2709	0.5434	0.3527	0.3042	0.2451
4	1.189	0.5976	0.3611	0.1247	0.1861	0.1382	0.1296	0.1217
	A	B	C	D	E	F	G	H

```
. . first=(d1*d1*p01+d2*d2*p02)/(2*(d1*p01+d2*p02))
. . first=first'
. . show (shape) first
```

E(Y)

0.5000	0.6250	0.7450	0.9050	0.5450	0.6350	0.7250	1.850

```
. . second=(d1*p1+d2*p2)/(1-p1-p2)
. . second=second'
. . show (shape) second
```

(E(N)-1)(E(Ri))

369.39	369.39	369.39	369.39	369.39	369.39	369.39	369.39
307.43	304.90	303.88	302.87	305.46	304.09	303.46	302.25
154.22	146.64	143.60	140.57	148.14	144.09	142.26	138.81
91.32	82.96	79.61	76.27	84.36	79.98	78.05	74.44
42.89	35.69	32.80	29.92	36.44	32.83	31.30	28.52
26.82	20.88	18.50	16.12	21.19	18.33	17.16	15.09
13.97	9.813	8.151	6.489	9.661	7.816	7.103	5.898
5.303	3.209	2.371	1.533	2.779	2.009	1.746	1.340
2.241	1.228	0.8223	0.4169	0.8497	0.5650	0.4803	0.3635
1.000	0.5197	0.3276	0.1354	0.2717	0.1764	0.1521	0.1226
0.1886	0.09481	0.05730	0.01979	0.02952	0.02193	0.02056	0.01930

```
. . total=first+second
. . show (shape) total
```

Adjusted ATS

shift	A	B	C	D	E	F	G	H
0.0	369.89	370.02	370.14	370.30	369.94	370.03	370.12	371.24
0.2	307.93	305.52	304.63	303.77	306.01	304.72	304.18	304.10
0.5	154.72	147.26	144.35	141.47	148.68	144.72	142.99	140.66
0.7	91.82	83.58	80.36	77.17	84.90	80.61	78.77	76.29
1.0	43.39	36.31	33.55	30.82	36.99	33.47	32.03	30.37
1.2	27.32	21.50	19.24	17.02	21.74	18.97	17.88	16.94
1.5	14.47	10.44	8.896	7.394	10.21	8.451	7.828	7.748
2.0	5.803	3.834	3.116	2.438	3.324	2.644	2.471	3.190
2.5	2.741	1.853	1.567	1.322	1.395	1.200	1.205	2.213
3.0	1.500	1.145	1.073	1.040	0.8167	0.8114	0.8771	1.973
4.0	0.6886	0.7198	0.8023	0.9248	0.5745	0.6569	0.7456	1.869

For this BLSS file, I have tried different intervals, and the results follow. It is clear that these additional runs are not necessary for obtaining an idea of what the best combination of intervals will be.

```
. . d1=0.0,.05,.01,.02,.05,.2,.3,.5
. . d1=d1'
. . d2=1.0,1.95,1.99,1.5,1.5,1.5,1.5,1.5
. . d2=d2'
. . q0=.0027
. . p01=(d2-1)*(1-q0)/(d2-d1)
. . p02=(1-d1)*(1-q0)/(d2-d1)
. . p=(1-p02)/2+p02
. . qgau p > gamma
. . show d1,d2,p01,p02,gamma
```

	d1	d2	Po1	Po2	gamma'
A'	0.000	1.000	0.000	0.9973	3.000
B'	0.05000	1.950	0.4987	0.4986	0.6724
C'	0.01000	1.990	0.4987	0.4987	0.6724
D'	0.02000	1.500	0.3369	0.6604	0.9549
E'	0.05000	1.500	0.3439	0.6534	0.9412
F'	0.2000	1.500	0.3836	0.6137	0.8664
G'	0.3000	1.500	0.4155	0.5818	0.8095
H'	0.5000	1.500	0.4987	0.4987	0.6724

```
. . x1=0.0,0.2,0.5,0.7,1.0,1.2,1.5,2.0,2.5,3.0,4.0
. . gup=gamma-x1
. . glow=-gamma-x1
. . up=3-x1
. . low=-3-x1
. . pgau up > pup1
. . pgau gup > pup2
. . pgau glow > pdown2
. . pgau low > pdown1
. . p2=pup2-pdown2
. . p1=pup1-pdown1-p2
. . percent=(d1*p1+d2*p2)/(p1+p2)
. . ats=percent/(1-p1-p2)
. . ats=ats'
. . show (shape) ats
```

ATS

	FSI	symmetric		asymmetric				
shift	d=1	(.05,1.95)	(.01,1.99)	(.02,1.5)	(.05,1.5)	(.2,1.5)	(.3,1.5)	(.5,1.5)
0	370.39	370.39	370.39	370.39	370.39	370.39	370.39	370.39
.2	308.43	303.60	303.40	304.20	304.29	304.77	305.12	305.89
.5	155.22	140.72	140.10	142.43	142.71	144.18	145.24	147.59
.7	92.32	76.26	75.58	78.04	78.36	80.03	81.23	83.87
1	43.89	29.88	29.29	31.24	31.53	33.06	34.15	36.52
1.2	27.82	16.10	15.61	17.11	17.37	18.69	19.63	21.65
1.5	14.97	6.509	6.153	7.084	7.280	8.294	9.003	10.52
2	6.303	1.573	1.374	1.750	1.871	2.492	2.920	3.814
2.5	3.241	0.4564	0.3391	0.4872	0.5646	0.9577	1.225	1.775
3	2.000	0.1748	0.09798	0.1607	0.2144	0.4852	0.6680	1.039
4	1.189	0.06564	0.01836	0.03666	0.07149	0.2461	0.3629	0.5976
		A'	B'	C'	D'	E'	F'	G'	H'

```
. . first=(d1*d1*p01+d2*d2*p02)/(2*(d1*p01+d2*p02))
. . first=first'
. . show (shape) first
```

E(Y)

0.5000	0.9513	0.9900	0.7450	0.7375	0.7000	0.6750	0.6250

```
. . second=(d1*p1+d2*p2)/(1-p1-p2)
. . second=second'
. . show (shape) second
```

(E(N)-1)(E(Ri))

369.39	369.39	369.39	369.39	369.39	369.39	369.39	369.39
307.43	302.62	302.41	303.21	303.30	303.78	304.13	304.90
154.22	139.81	139.20	141.51	141.79	143.25	144.31	146.64
91.32	75.43	74.76	77.20	77.51	79.17	80.35	82.96
42.89	29.20	28.62	30.53	30.81	32.31	33.37	35.69
26.82	15.52	15.05	16.50	16.74	18.02	18.92	20.88
13.97	6.074	5.742	6.610	6.793	7.739	8.402	9.813
5.303	1.324	1.156	1.472	1.574	2.096	2.456	3.209
2.241	0.3156	0.2345	0.3369	0.3904	0.6622	0.8473	1.228
1.000	0.08741	0.04899	0.08034	0.1072	-0.2426	0.3340	0.5197
0.1886	0.01041	0.002913	0.005817	0.01134	0.03905	0.05758	0.09481

```
. . total=first+second
. . show (shape) total
```

Adjusted ATS

shift	A'	B'	C'	D'	E'	F'	G'	H'
0.0	369.89	370.34	370.38	370.14	370.13	370.09	370.07	370.02
0.2	307.93	303.57	303.40	303.96	304.04	304.48	304.80	305.52
0.5	154.72	140.76	140.19	142.26	142.53	143.95	144.98	147.26
0.7	91.82	76.38	75.75	77.94	78.25	79.87	81.03	83.58
1.0	43.39	30.15	29.61	31.27	31.55	33.01	34.05	36.31
1.2	27.32	16.47	16.04	17.24	17.48	18.72	19.60	21.50
1.5	14.47	7.025	6.732	7.355	7.531	8.439	9.077	10.44
2.0	5.803	2.275	2.146	2.217	2.312	2.796	3.131	3.834
2.5	2.741	1.267	1.225	1.082	1.128	1.362	1.522	1.853
3.0	1.500	1.039	1.039	0.8253	0.8447	0.9426	1.009	1.145
4.0	0.6886	0.9617	0.9930	0.7508	0.7488	0.7390	0.7326	0.7198

Multivariate Statistical Process Control for a Plasma Etcher

Hai-Fang Guo

Abstract

A set of Remote Procedure Language (RPL) procedures has been developed for the RS/1 statistical package. These procedures take raw real-time data collected off a plasma etcher and will calculate the Hotelling's T^2 statistic. The resulting T^2 values have been analyzed and used on a multivariate SPC scheme for plasma etch control.

1. Introduction

While the size of IC devices is shrinking, the complexity of the IC process increases. In order to get high yield and high quality of the products, tight control of the process is critical. Plasma etching plays a fairly important role in the whole process. The objective of this project is to develop a reliable multiavriate SPC methodology for real-time control of the plasma eching process.

There are many control variables which will effect the performance of a plasma etcher. These are gas flows, RF power, pressures and other parameters. With the proliferation of modern equipment communication protocols like the Semiconductor Equipment Communication System II (SECSII), it is fairly easy to collect real time readings of these parameters at a reasonable sampling frequency. As a matter of fact, many processing equipment now have an array of sensors and enough on-board intelligence to accomplish the task. The challenge however remains on how to make the best use of this information in a formal, robust (SPC) scheme. In the past, univariate SPC methods have been applied. These methods treat each variable separately by plotting one control chart per variable. Unfortunately, the critical variables tend to be highly cross correlated, so these methods might give misleading information by means of false as well as missing alarms. In fact, it has been shown that as the number of variables increases, the distortion in the joint control procedure can be severe. One way to avoid this is by using a multivariate SPC scheme. This can be accomplished by using Hotelling's T^2 statistic to convert multivariate information to a single variable.

The objectives of this project is to apply the Hotelling's T^2 statistic to the real-time data that has been collected from the Lam Reseach 490 Plasma Etcher at UCB, and draw some conclusions about the applicability of the control scheme and the stability of the equipment.

2. Methodology

The Hotelling's T^2 is a method that can be used to solve the multivariate SPC problem. The T^2 parameter is defined below:

$$T^2 = n\,(\bar{X} - M)'\,S^{-1}\,(\bar{X} - M) \qquad (1)$$

where

$$\bar{X}' = [\bar{X}_1, \bar{X}_2, ..., \bar{X}_p]$$

$$M' = [M_1, M_2, ..., M_p]$$

$$S = \begin{bmatrix} S_1^2 & S_{12} & \cdots & S_{1p} \\ S_{12} & S_2^2 & \cdots & S_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ S_{1p} & S_{2p} & \cdots & S_p^2 \end{bmatrix}$$

$\bar{X}_i$ is the average of the ith parameter
M_i is the target value of the ith parameter
S is the covariance matrix of the outcome variables
n is the sample size
p is the number of variables

T^2 is the square of the maximum possible univariate t computed on any linear combination of the various outcome measures. It has been shown that when multipled by (n-P)/P(n-1), the new statistic obeys an F-distribution with p and n-p degrees of freedom. That is,

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

where α is the P{type I error}, an n, p as defined in Eq. (1). The T^2 control chart can be constructed by using the T^2 values with upper control limit at $T^2_{\alpha,p,n-1}$, which can be calculated by Eq. (2) According to this control scheme, an alarm will be initiated if the sample mean results in a T^2 value that exceeds the critical value at the chosen level of significance.

3. Implementation

The implementation of this experiment can be divided into four steps as follows:

Step 1: *Data Collection*

In this experiment we etched polysilicon off 32 wafers, and we have collected the raw sensor data generated during the etch processes. In total, we have monitored seventeen parameters at a rate of one sample per second. Among the seventeen, there are five parameters which can be adjusted for different recipes. These are the gas flows of O_2, He and CCl_4, the RF power of the process and the pressure inside the reaction chamber. The data has been collected during the entire process, including the initialization step, etching step, and over etching step. Although all the steps are important to the outcome of the process, here we only consider the main etching step. To manipulate the data, we cluster the samples from the etching step into groups of size 10 and then calculate the average of each group. These averages, calculated from samples collected during the main etch process, form the data that we will use in this study.

Step 2: *Choosing a "good" process run as standard.*

Since we do not know when the etcher is actually in statistical control, we have to choose a "good" process run and use it as a standard. In this sense, our definition of a "good" process is relative. We assume that a process is in control if all of its parameters are close to their measured averages. So, we calculate the parameter averages for each of the 32 wafers. The results have been plotted on a chart from which we select the "good" process. In this experiment, the run for wafer pe8 is chosen as the standard. Once the standard process has been identified, we can construct the variance-covariance matrix and calculate the grand average of the parameters. Note that our initial assumption about the standard process has to be checked and validated, much in the way one checks and validates the original control limits in a new Shewhart control chart. Specifically, we use the calculated variance-covariance matrix and target means, in order to calculate the T^2 for wafer pe8 as described in step 3 below. We then drop the samples with a T^2 higher than the control limit. We subsequently use the rest of the sample averages to re-calculate the variance-covariance matrix and the target means. A new limit is set and this procedure continues until all the T^2 values for wafer pe8 all below the limit.

Step 3: *Calculate* T^2 *values and plot* T^2 chart.

Under the assumption that the autocorrelation within each of the parameters is negligible, we apply Hotelling's T^2 method using the calculated variance-covariance matrix and the grand average for each of the 32 wafers. We then create the T^2 charts, from which the conclusions about whether the process is in statistical control can be drawn. The upper control limit of the T^2 control chart is determined according to Eq. (2) as follows:

$$T^2_{0.05,5,9} = \frac{5\times(10-1)}{10-5} F_{0.05,5,5} = 9\times 5.05 = 45.45$$

There is no lower control limit for this chart.

Step 4: *Analyze the out-of-control points in the* T^2 *chart and investigate the causes.*

This step involves the discovery of the physical cause for the abnormal behavior of the equipment. Beyond some simple, intuitive explanations, this investigation is beyond the scope of this project.

4. Results

Listed in the order of the steps outlined above, the results are:

For step1:

A set of RPL procedures has been written to manipulate the raw data and read the resulting average values into RS/1 tables.

For step 2:

The overall averages and standard deviations for each of the 32 wafers have been calculated and presented in Table 1. The respective charts are shown in Fig. 1. Based on these charts, the run for wafer pe8 has been chosen as the standard. The variance-covariance matrix and grand-average of wafer pe8 have been calculated and presented in Table 2.

For step 3:

A procedure has been written for the calculation of the T^2 values. The T^2 values are summarized in Table 3. As we can see in this table, the T^2 values are rather large at the beginning of the etching step. This indicates a persistent process stabilization problem.

For step 4:

The out of control point during the processing of wafer pe1 has been investigated by comparing the parameter averages with the standards. The deviation might have been caused by unusually high pressure variability. The large T^2 values for the wafers following pe18 might be related to the fact that all these wafers have been processed inividually. i.e. as separate lots of one wafer each. This indicates that our plasma etcher suffers from a chronic recipe stabilization problem at the beginning of each lot. Indeed, it is typically after the first wafer of the lot (in a cassette to cassette mode of operation) that the process stabilizes. Further investigation is needed to evaluate the impact of this instability to product quality.

5. Conclusions

Using the T^2 statistic to analyze data from the Lam etcher gives us valuable information about the process. We found a number of instances in our data where the individual charts either missed a true alarm or generated false alarms. Further, the T^2 value varies with the changing of the group sample size. This suggests that our assumption that there is no auto-correlation within each parameter might false. A remedy to this will be to filter the data before applying T^2 statistic. The Box-Jenkins autoregressive models might be useful in this respect.

The conclusions about the stability of our plasma etcher ticular control scheme are as follows: First, the tendency that the T^2 values are higher that the upper control limit at the beginning of the process suggests that there might be a stablization problem for the Lam etcher at the beginning of the etching step. Second, the T^2 values for wafers that have been processed individually (single wafer process) tend to be higher than the wafers been processed together in a cassette to cassette mode. This suggests that the first wafer run in each lot is not very stable.

6. Acknowledgement

We are grateful to Texas Instruments and the Semiconductor Research Corporation for support of this research.

References

[1] Douglas C. Montgomery, Introduction to Statistical Quality Control, John Wiley & Sons, New York, 1985.

[2] Richard Harris, A Primer of Multivariate Statistics, Academic Press, New York, 1975.

[3] Steven Dolins, "Monitoring and Diagnosis of Plasma Etch Processes", IEEE transactions of semiconductor manufacturing, Vol. 1, No. 1, Feb. 1989.

[4] J. Bert Keats, Statistical Process Control in Automated Manufacturing Marcel Dekker, Inc., New York and Basel, 1989.

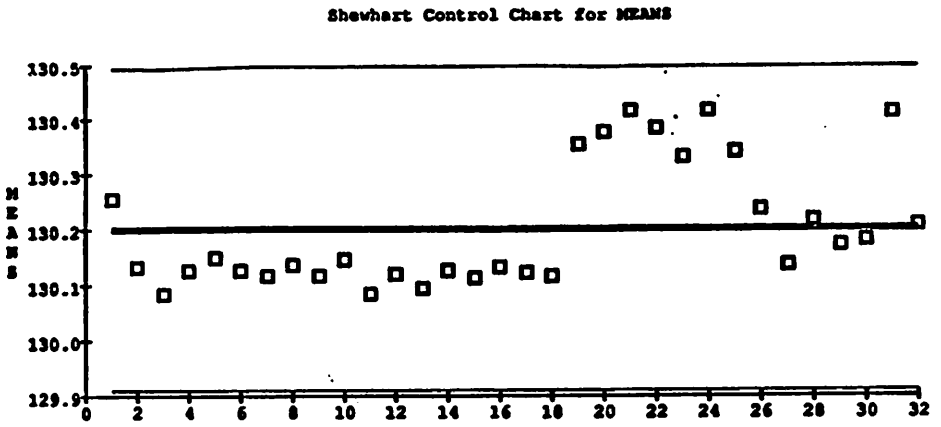


Fig.1 (a) overall means of CCl4

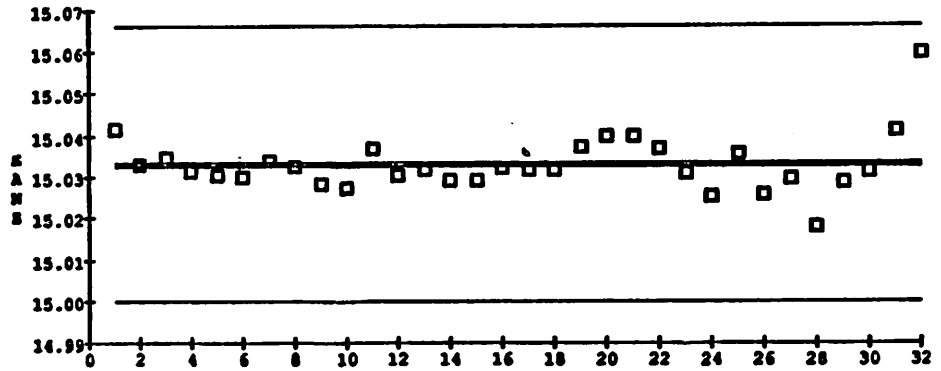


Fig.1 (b) overall means of O2

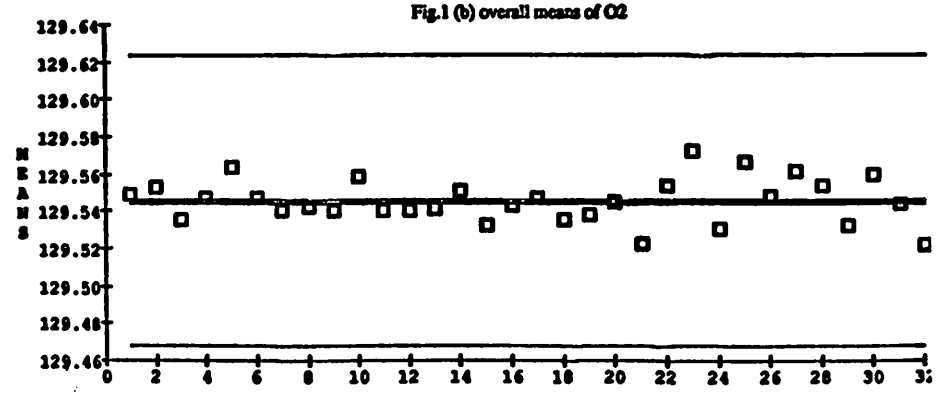


Fig.1 (c) overall means of He

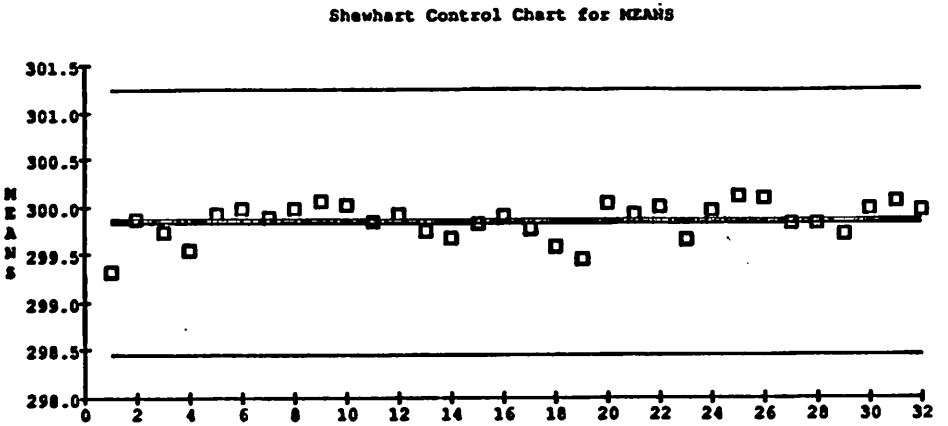


Fig.1 (d) overall means of RF power

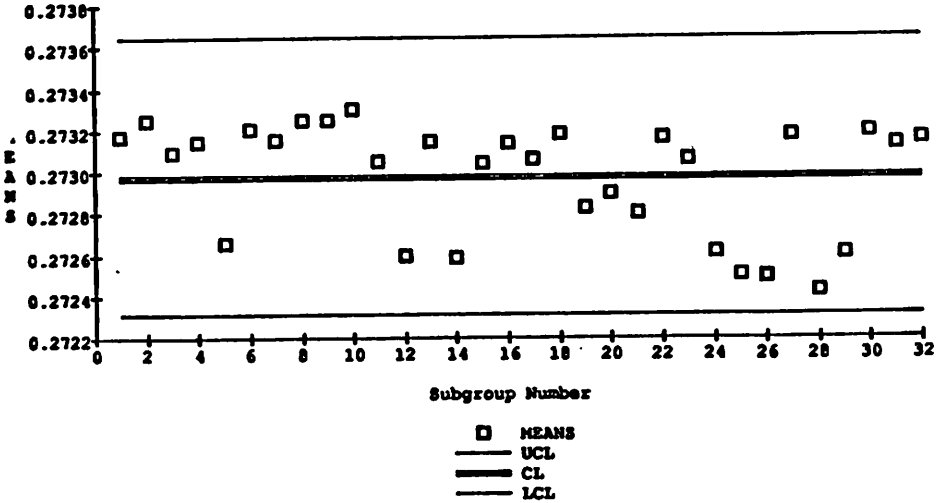
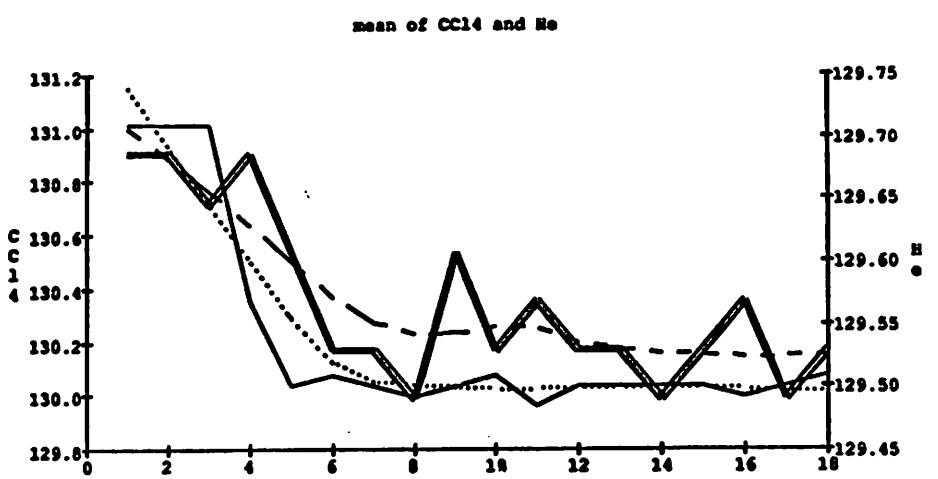
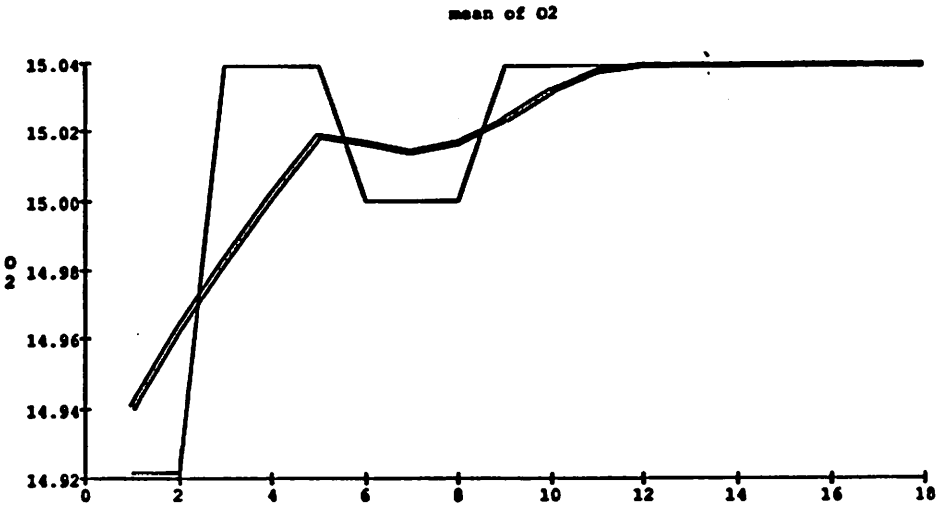


Fig.1 (e) overall means of Press





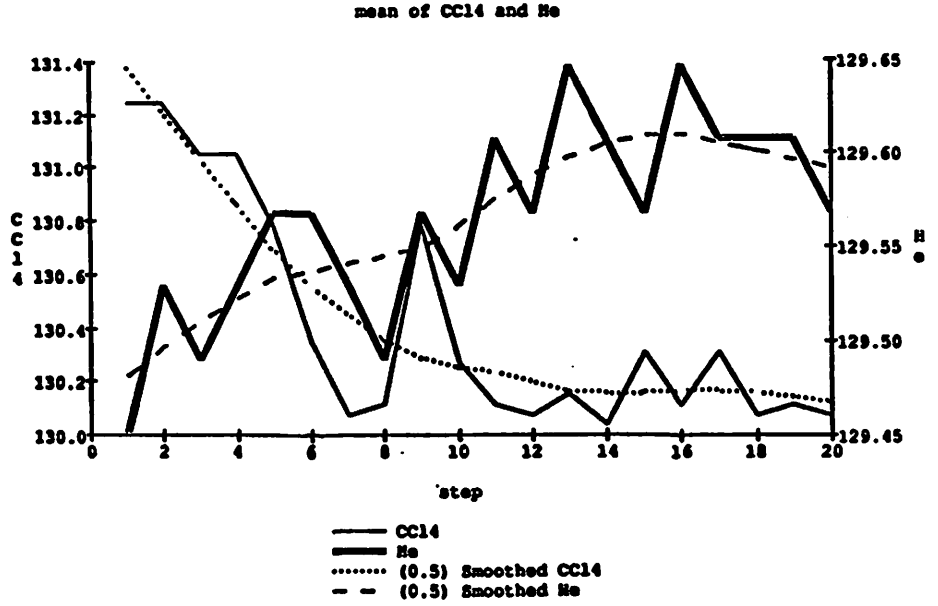


Fig.2 (a) comparison of CCl4 and He of pe1 to pe8

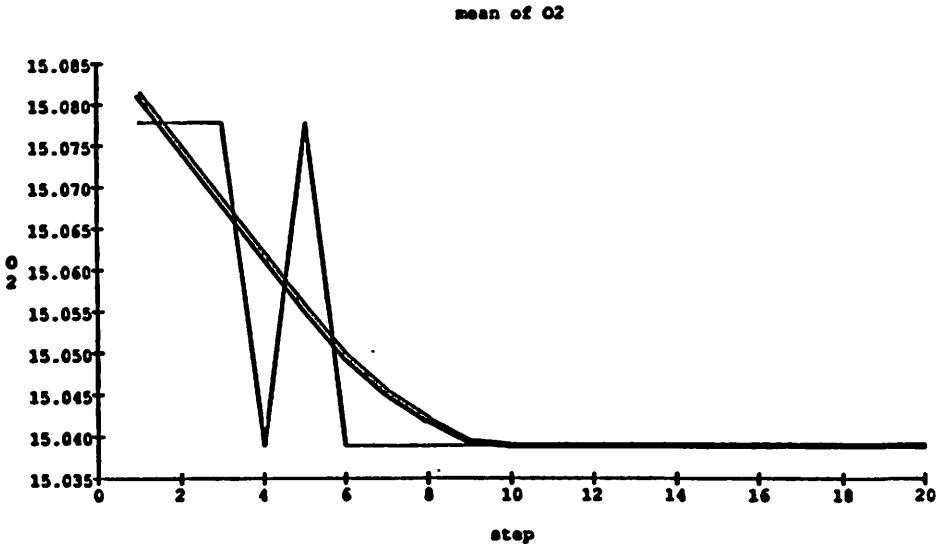
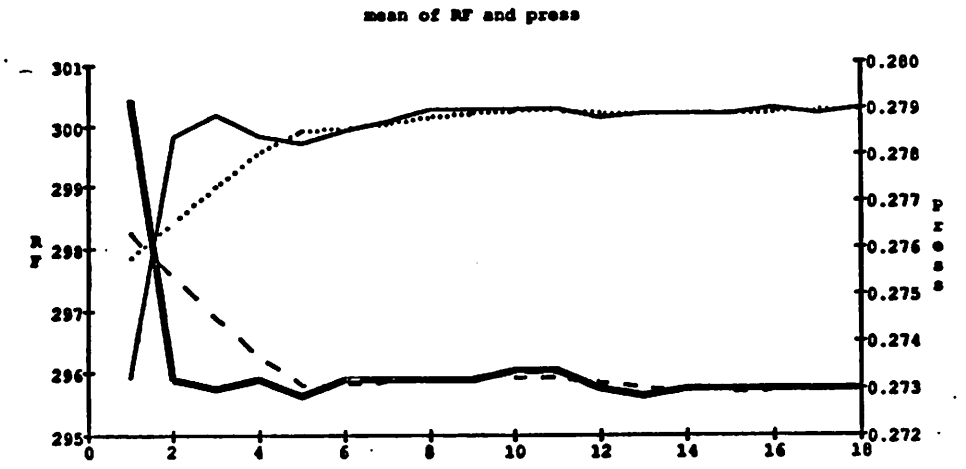
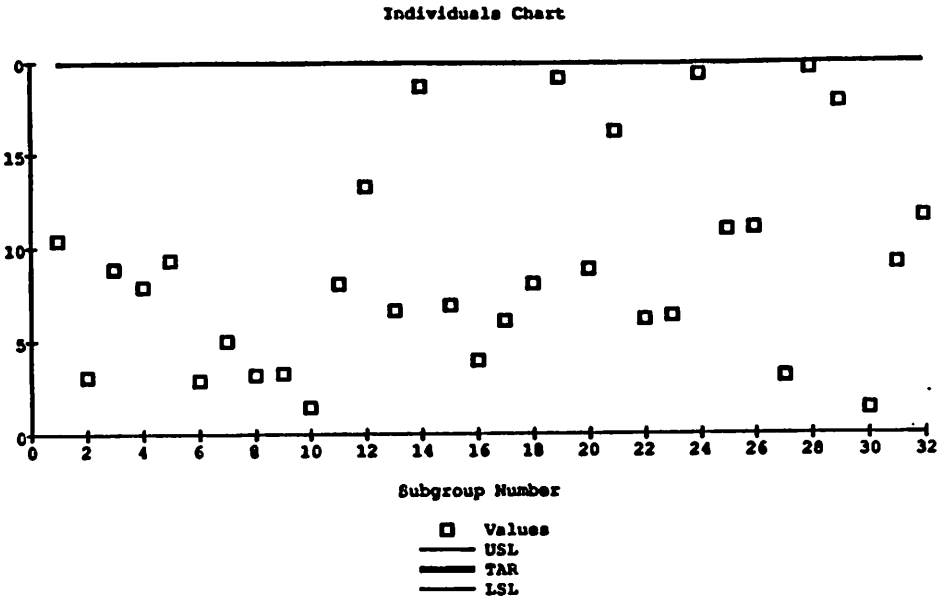


Fig.2 (b) comparison of O2 of pe1 to pe8





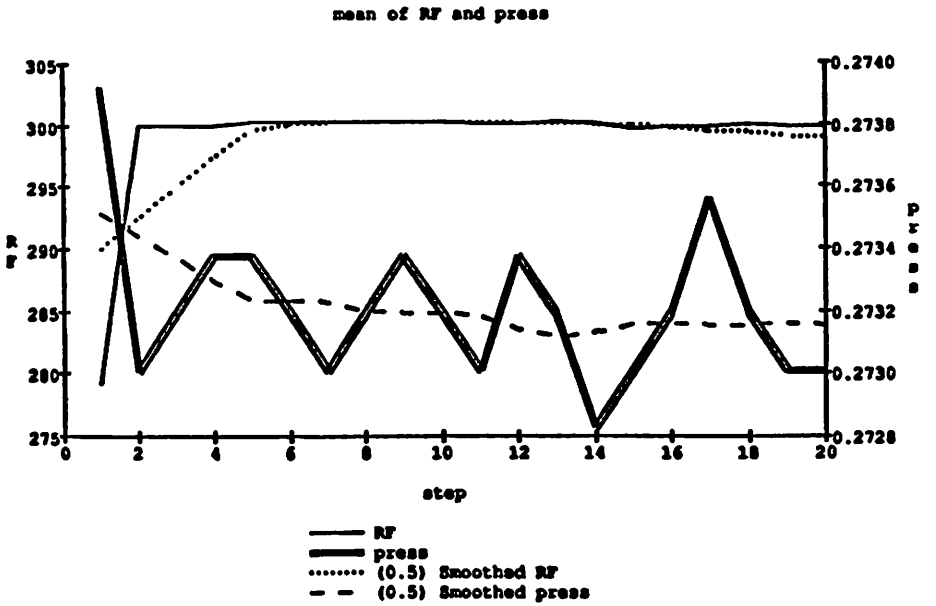


Fig.2 (c) comparison of RF power and Pressure of pe1 to pe8

Fig.3 The overall T^2 values of the 32 process

0	1 CCl4	2 O2	3 He	4 RF	5 Pres
1	pe1	130.253841	15.041701	129.549027	299.320738	0.273171
2	pe2	130.133622	15.033154	129.552456	299.865569	0.273246
3	pe3	130.084950	15.034624	129.535110	299.722376	0.273097
4	pe4	130.125521	15.031322	129.547140	299.542103	0.273147
5	pe5	130.147153	15.030482	129.563619	299.926204	0.272653
6	pe6	130.126649	15.030141	129.547266	299.977257	0.273205
7	pe7	130.114897	15.033676	129.540394	299.880742	0.273151
8	pe8	130.135480	15.032431	129.542038	299.972208	0.273253
9	pe9	130.114735	15.028074	129.540582	300.051868	0.273245
10	pe10	130.144560	15.027306	129.558630	300.009168	0.273305
11	pe11	130.083041	15.036676	129.539904	299.837455	0.273056
12	pe12	130.118687	15.030424	129.540147	299.928506	0.272598
13	pe13	130.094961	15.031674	129.540897	299.747749	0.273145
14	pe14	130.125285	15.029104	129.551291	299.678557	0.272583
15	pe17	130.113981	15.029249	129.532861	299.816389	0.273043
16	pe18	130.132881	15.032360	129.543696	299.908430	0.273138
17	pe19	130.123769	15.031591	129.547544	299.771492	0.273057
18	pe20	130.117361	15.031712	129.535521	299.573084	0.273176
19	tp5	130.351472	15.037434	129.537919	299.440051	0.272830
20	tp6	130.373930	15.039872	129.545615	300.023440	0.272893
21	tp7	130.412761	15.040026	129.522887	299.927259	0.272803
22	tp8	130.382161	15.036747	129.554216	299.997430	0.273165
23	tp9	130.329075	15.030683	129.572394	299.653211	0.273057
24	tp10	130.412573	15.025244	129.530271	299.956587	0.272614
25	tp11	130.339116	15.035503	129.566856	300.105286	0.272498
26	tp12	130.235821	15.025690	129.548043	300.087743	0.272489
27	tp13	130.135991	15.029470	129.561550	299.816641	0.273171
28	tp14	130.216906	15.018022	129.554168	299.820165	0.272425
29	tp15	130.171821	15.028784	129.532708	299.706277	0.272608
30	tp17	130.181543	15.031440	129.559523	299.978846	0.273193
31	tp18	130.408595	15.041077	129.544597	300.043829	0.273127
32	tp19	130.207096	15.059565	129.521571	299.955547	0.273152

Table 1 overall means of the 32 process

0	1	2	3	4	5
1	130.215345	15.019339	129.568652	299.868535	0.273438

Table 2 (a) target means

PE8_C 5R x 5C

0	1	2	3	4	5
1	0.037793	0.000848	0.000565	-0.028818	0.000021
2	0.000848	0.006876	0.001483	0.011008	-0.000018
3	0.000565	0.001483	0.011538	0.044855	-0.000049
4	-0.028818	0.011008	0.044855	1.899715	-0.001832
5	0.000021	-0.000018	-0.000049	-0.001832	0.000002

Table 2 (b) target variance-covariance matrix

PE8_COR 5R x 5C

0	1	2	3	4	5
1	1.000000	0.408248	-2.467162e-17	-0.198574	0.097720
2	0.408248	1.000000	0.456435	0.298936	-0.407564
3	-2.467162e-17	0.456435	1.000000	0.749292	-0.755918
4	-0.198574	0.298936	0.749292	1.000000	-0.936041
5	0.097720	-0.407564	-0.755918	-0.936041	1.000000

Table 2 (c) target correlation matrix

0	1 pe1_a	2 pe2_a	3 pe3_a	4 pe4_a	5 pe5_a	6 pe6_a	7 pe7_a
1	5758.19135	240.08694	66.78265	1300.80241	208.30671	139.05253	117.30744
2	181.33546	66.98479	3.53988	121.34801	173.53903	151.57213	176.62329
3	132.43264	168.94190	4.86700	68.05341	109.47781	105.38021	51.51504
4	118.01211	8.72403	10.67143	18.96807	13.05616	8.94767	6.56957
5	64.27761	33.34526	19.27623	17.56244	26.90750	8.05831	15.46376
6	5.06283	13.71105	17.91328	8.58989	36.87851	8.27301	7.62067
7	8.13416	27.93749	17.15171	10.43061	24.55663	21.53777	9.45323
8	12.22774	5.66717	8.05831	15.46376	20.98066	7.72885	14.51348
9	55.08347	5.27272	8.16658	8.15767	19.50634	13.44193	11.84322
10	5.69374	12.86221	14.43746	6.40771	11.05727	21.93987	5.89251
11	4.58784	13.90736	15.98514	15.39117	16.25604	5.94150	14.52776
12	6.42496	6.06943	26.10856	7.34942	16.65082	35.50389	8.96910
13	7.82293	8.27301	27.18992	13.06459	12.74986	5.82194	13.43680
14	9.67623	11.53211	20.35866	8.58985	19.33507	11.18923	24.71235
15	7.36313	14.09410	13.75804	14.71295	24.71235	9.91798	14.51348
16	10.29417	20.59735	6.34956	24.14013	23.74943	8.08301	12.40878
17	4.03980	21.21364	19.67944	21.49987		11.18923	13.39110
18	7.04068	10.51388	18.71334	16.91675			18.50885
19	7.60986	16.49059		16.73246			
20	7.92130	7.94148		13.49928			

0	8 pe8_a	9 pe9-a	10 pe10_a	11 pe11_a	12 pe12_a	13 pe13_a	14 pe14_a
1	305.91520	259.68037	376.78658	59.99021	990.40537	1932.28288	1602.94093
2	196.41730	271.12324	265.93567	50.99618	255.45837	121.58823	118.57456
3	103.06010	108.02405	57.46805	10.54873	105.28927	64.06816	174.11259
4	14.51134	13.57129	9.53374	8.63933	15.12392	15.64299	15.95451
5	16.04591	9.51679	7.00223	15.17766	21.41191	14.39385	18.63258
6	6.87195	14.21650	6.41902	20.74724	11.70358	11.26197	18.58006
7	7.52280	6.08822	7.39589	11.95223	11.89249	11.35371	14.56761
8	13.28355	23.87947	13.57516	9.13327	25.71047	23.49125	14.02027
9	9.21662	23.88743	10.43061	9.63765	7.76801	28.16705	28.08535
10	9.34190	18.77429	6.68411	16.73246	14.53995	19.97739	21.49615
11	15.48320	20.44900	6.09761	29.10228	23.37163	13.39110	34.28591
12	10.36638	10.16539	13.61706	9.21977	16.53870	14.51348	29.33566
13	11.53211	10.08669	10.08669	13.44193	7.31679	14.54208	16.25604
14	15.82009	9.34190	10.16539	13.67166	14.99187	9.10990	19.70300
15	10.15864	15.46376	18.13502	26.78402	21.31115	11.80434	22.73976
16	10.26144	26.65162		11.15822	11.53211	8.10562	27.18992
17	15.82009	8.15767		11.66886	20.98066	13.59326	18.36586
18	8.08813	17.18832			30.84424	11.35878	36.49989
19		14.39385				7.62067	
20							

Table 3 The T^2 values of the 32 processed wafers, calculated every ten readings

0	15 pe17_a	16 pe18_a	17 pe19_a	18 pe20_a	19 tp5_a	20 tp6_a	21 tp7_a
1	1430.07320	1430.07320	940.65396	1360.84672	5009.17863	149.91262	307.14370
2	126.95215	126.95215	108.15574	167.31664	437.90649	66.14574	114.99537
3	176.14734	176.14734	176.96578	72.16903	211.67149	41.64165	194.58629
4	8.26048	8.26048	5.31533	22.01809	34.20553	58.80194	51.28824
5	5.65506	5.65506	4.39081	13.06459	56.83881	3.58482	178.00999
6	11.53211	11.53211	6.09761	14.39385	39.12915	123.81182	27.66459
7	6.09761	6.09761	9.92633	10.47265	23.49375	19.90883	8.03925
8	6.93444	6.93444	9.13327	9.95989	72.55255	52.34253	54.37850
9	19.54548	19.54548	7.31679	7.80660	23.67273	15.00494	91.21816
10	15.02735	15.02735	6.14638	5.93806	5.72547	6.95772	44.57909
11	37.71168	37.71168	15.72254	9.89986	19.43256	5.06283	48.89140
12	8.48370	8.48370	46.92124	12.59024	16.86998	85.89010	54.61601
13	16.27350	16.27350	12.35951	14.39385	18.48445	41.27451	2.14781
14	13.90736	13.90736	19.85850	16.14822	13.45589	4.24411	5.78595
15	10.08669	10.08669	13.93283	18.79200	2.05533	34.89661	47.44064
16	12.59024	12.59024	10.73295	21.94076	42.72021	26.24413	44.52044
17	15.69578	15.69578		26.57987	10.26144	6.09273	
18	23.97810	23.97810		19.07905	115.36012	8.36453	
19	11.53865			14.00942	32.70687	9.81783	
20	7.09910				8.38223		

0	22 tp8_a	23 tp9_a	24 tp10_a	25 tp11_a	26 tp12_a	27 tp13_a	28 tp14_a
1	473.07415	91.30946	204.04251	84.07442	233.90076	332.65348	1438.86556
2	56.63346	20.51357	169.75833	52.27576	69.16685	112.80646	143.35942
3	40.68370	45.56510	46.12414	26.32508	125.94738	59.51179	241.47361
4	86.89661	24.26614	47.60528	10.14714	78.76734	10.72869	13.85640
5	2.57938	4.36552	17.40896	36.71416	20.13331	15.37091	27.50644
6	36.75801	10.24314	62.52525	48.21961	16.04706	4.39168	42.49022
7	10.97008	18.15435	19.24455	26.51114	19.81670	13.08465	27.71356
8	18.66085	13.51019	17.53508	21.53586	16.77203	16.29776	44.06569
9	46.89375	32.23089	24.04999	11.47073	38.95592	25.81641	23.60045
10	39.21270	96.55226	36.22489	40.11208	56.89595	11.66886	8.96655
11	38.36873	19.68059	67.98370	40.91217	20.98066	27.20562	15.53504
12	20.91118	9.49403	45.05285	18.87932	37.81602	6.77182	19.01477
13	6.61466	28.14905	83.92630	35.05405	34.02392	10.99553	31.90309
14	13.40719	6.97399	37.35864	23.28155	26.43592	28.58535	4.60002
15	3.06651	19.87899	18.65332	23.85334	55.46203	35.21916	42.26916
16	8.31025	9.70491	16.15889	63.55608	34.97882	13.57516	20.91874
17	29.87208		80.71983	19.25212		15.83301	21.99581
18	9.06694					15.39117	
19							
20							

Table 3 The T^2 values of the 32 processed wafers, calculated every ten readings

0	29 tp15_a	30 tp17_a	31 tp18_a	32 tp19_a
1	744.28093	271.77910	255.18034	328.19128
2	128.09459	146.77548	19.80738	221.22236
3	50.86479	174.19987	31.46249	110.43668
4	66.24057	7.66384	94.04347	24.33590
5	20.70335	6.79298	36.61293	12.50735
6	23.29762	11.13989	27.68211	24.64880
7	22.97733	10.87000	24.58823	6.71590
8	22.54737	9.70156	25.53682	16.00726
9	12.92175	11.18923	30.81521	8.30557
10	30.46499	15.63164	110.12755	9.82055
11	16.29496	11.53211	2.53309	6.93444
12	32.19813	6.62447	20.02022	27.14483
13	28.55429	12.31913	15.37568	16.00726
14	9.02550	16.49059	17.05793	11.17729
15	18.84331	15.72254	36.11031	12.35951
16	40.65790	16.27350	19.29121	
17	21.58312	8.08301	16.59651	
18	42.95374	5.69374		
19	25.73997			
20	24.41181			

Table 3 The T^2 values of the 32 processed wafers, calculated every ten readings

A Strategy for Adaptive Regression Modeling of LPCVD Reactors

Sherry F. Lee

Abstract

Since VLSI fabrication equipment often change in time, there is an interest in creating equipment models that can update themselves accordingly. In this report we present a "smart" regression model that can decide whether it should be refitted for better predictive performance, or for new parameter values if the equipment has changed. These decisions are based on formal statistical tests. Safeguards to prevent overcorrection and extensive equipment wear are also incorporated to this adaptive strategy.

1. Introduction

With the development of fields such as computer-aided manufacturing (CAM) and computer-integrated manufacturing (CIM), statistical equipment modeling has become important. An effective model should provide the capability to periodically check the fit between the model and the equipment process, and update the model automatically when necessary.

A systematic method of building and calibrating equipment specific process models has been developed and sucessfully applied [1] to the modeling of a low pressure chemical vapor deposition (LPCVD) furnace for undoped polysilicon. The goal of this project is to develop an algorithm to systematically update the previously developed linear model for the LPCVD process.

2. Methodology

The algorithm to update the equipment model uses three distinct statistical tools. The first is the regression control chart [2], which pin-points out of control data. A second useful method to check for out of control conditions is the cumulative sum of the difference between the actual and model predicted rates. By looking at the control chart and the cumulative sum, and by using scientific judgement, the user can determine whether to update the model or to check the equipment. Should the decision be to update the model, regression analysis will be performed to determine the new coefficients of the revised model.

2.1. Regression Control Chart

The regression control chart is similar to a conventional Shewart control chart in that they both consist of a center line with upper and lower control limits. When the points fall outside of the control limits, the process is considered to be out of control. However, while the center line and control limits of the Shewart control charts are parallel to the horizontal axis, indicating control over a single fixed average, the control limits of the regression control chart follow the regression line, thereby controlling a varying average. The control limits of the regression control chart are based on the standard deviation of the residuals (difference between the actual value and the predicted value).

The regression control chart is used in the first step of the analysis. Out of control data points can be observed easily, informing the user either that there is a problem in the equipment or that the model needs to be updated. Thus, the regression control chart enables the user to quickly pinpoint those data that do not follow the model within a specified limit.

2.2. Cumulative sum

The cumulative sum method also alerts the user that the data is moving out of the desired range. It is based on the sum of the difference between the actual and predicted values (residuals). Therefore the CUSUM, unlike the regression control chart which indicates a point-by-point deviation, indicates that as a

whole the deviations from the model are significant. In some cases, when most but not all of the data points are in control on the regression control chart, the cumulative sum shows no significant difference. This is because the cumulative sum is based on a summation of the residuals, and not on each residual alone. So although the single point may be out of control, the model as a whole still holds. The cumulative sum of the residuals can be tested for significance, by using a student-t test, which indicates whether or not a revision of the model and the regression control chart limits is necessary. Residual plots are also useful in detecting runs or trends in time.

2.3. Regression analysis

Since both the regression control chart and the cumulative sum only indicate that the predicted value is significantly different from the actual value, a regression analysis is used to determine which coefficients need to be revised.

3. Implementation

In this project, we would like to update the model that describes the rate of deposition of the polysilicon in an LPCVD furnace. The deposition rate R(z) depends on the rate R_0, the deposition rate at the first wafer position (z_0) in the furnace.

$$R(z) = [\frac{1 - const * R_0}{1 + const * R_0}]R_0$$

Since R_0 can be described by a linear model, it is easier to do the analysis on R_0 and apply the results to the full deposition rate R(z). From K.K. Lin's results [1], the linear model of the deposition rate R_0 is known:

$$\ln(R_0) = A + B\ln(P) + C(1/T) + D(1/Q)$$

where P is the pressure (mtorr), T is the temperature (K), Q is the silane flow (sccm), and A, B, C, and D are the coefficients to be determined. To make notation easier, set $Y = \ln(R_0)$, $X_B = \ln(P)$, $X_C = 1/T$, $X_D = 1/Q$. Thus, the final equation is:

$$Y = A + Bx_B + Cx_C + Dx_D$$

The first step is to determine the coefficients of the linear model. By using the 23 data points that K.K. Lin observed, the coefficients can be obtained. I found slightly different coefficients than were reported in that paper, probably because I used BLSS instead of RS-1. For consistency with my results, I used the coefficients that I generated in BLSS.

Next, the variances of both the residuals and the sum of the residuals must be calculated. The variance of the residuals is used to generate the control limits for the regression control chart, while the variance of the sum of the residuals is used to test for significance of the sum of residuals. Calculations for each of these values are in Appendix A. The final result for the variance of the sum of the residuals is:

$$\text{var}\sum_{i=1}^{n}(Y_i' - y_i') = \frac{n^2 s_y^2}{N} + ns_y^2 + \text{varB}[\sum_{i=1}^{n}(x_{Bi}' - \bar{x}_B)]^2 + \text{varC}[\sum_{i=1}^{n}(x_{Ci}' - \bar{x}_C)]^2 + \text{varD}[\sum_{i=1}^{n}(x_{D_i}' - \bar{x}_D)]^2$$

where

x_i' = new data
Y_i' = predicted value based on new data
y_i' = actual value that corresponds to the new data
x_i = data used to generate the model
$\bar{x}_i$ = average of the x_i samples
N = the number of x_i samples
n = the number of x_i' samples
s_y = the standard error of the estimate of the regression based on original data

If n=1, we obtain the standard error of each residual, which is used to determine the control limits in the regression control chart.

$$\text{var}(Y_i' - y_i') = \frac{N+1}{N}s_y^2 + \text{varB}(x_{Bi}' - \bar{x}_B)^2 + \text{varC}(x_{Ci}' - \bar{x}_C)^2 + \text{varD}(x_{D_i}' - \bar{x}_D)^2$$

Using $s_i^2 = \text{var}(Y_i - y_i)$, the regression control chart can be constructed. With 20 degrees of freedom, the 95% student t statistic is 2.086, which results in control limits at $(Y_i +/- (2.086\ s_i))$.

To test the significance level of the sum of the residuals,

$$t_n = \frac{\sum_{i=1}^{n} (Y_i' - y_i)}{\text{var} \sum_{i=1}^{n} (Y_i' - y_i)}$$

N+n–4 degrees of freedom

The comparison of this calculated t with the student-t that corresponds to 95% confidence levels indicates whether the model is in agreement with the new observation. If the calculated value is greater than the given value, then the data point is considered to be out of control. The user must then decide whether to check the furnace or change the model. If the decision is to readjust the model, regression analysis must be performed to determine which coefficient needs to be revised. Several approaches can be implemented.

a) Recalculate the linear regression using solely the new data.

b) Recalculate the linear regression using the weighted method, giving more weight to the later data points.

c) Use regression to re-estimate one coefficient while holding the other three fixed. If the new coefficient differs from the old one by more than the standard deviation of the new value, the old value is replaced by the new value. This is continued until no further changes in coefficients will result in a linear model that better fits the data.

The first approach will only be valid if there are enough new data points to justify a new regression analysis. At first, there will not be enough new data points to implement this method. Also, there may not be enough "spread" in the runs to obtain an accurate linear regression. For example, users tend to use the same silane flow run after run (100 sccm). If all the data points contain the same flow rate, the linear regression will eliminate flow as an important parameter, which would result in an incorrect model. However, for large shifts, this approach may be the best.

The second approach can be used effectively for small changes in the deposition rate. For example, as a coat of polysilicon builds up on the furnace walls, the deposition rate may decrease in a steady fashion. In cases such as these, the weighted linear regression would be effective. However, this method is not recommended for data involving large shifts.

I chose to use the third approach for several reasons. First, by adjusting one parameter at a time, the model changes minimally. This may be important if the user is doing several runs that are supposed to be based on the same model. Varying the model significantly in the middle of the runs may alter the conditions too drastically. Small, gradual changes of the model are best in this case, assuming that the equipment did not change significantly.

Second, with the initial base of 23 points, the variance of the model decreases every time the model coefficients are adjusted (assuming that the shift is not too big). Essentially, the model simply fine-tunes itself as more runs are performed.

After the regression has been adjusted, the regression control chart limits should be recalculated. Then the entire process begins again for the next data point.

4. Results

Fig. 1 shows the calculated coefficients for the deposition model, using the reported 23 runs. Twelve sample runs were then generated, with three separate sets of possible 'actual' values. The first set consists of in control points, the second consists of in-control points, and the third consists of some in-control points and some out of control points. The 'actual' values for the in control deposition rates were generated by using a Gaussian random number generator.

The resulting regression control charts are found in Fig. 2. As expected, the points in the first set of in-control points are well within the control limits, while the points in the second set of out of control points are all outside the control limits (Fig. 2(a)). The third set of points results in nine points that are in

control, and three (run #3, 11, and 12) that are out of control (Fig. 2(b)).

The t-statistic computation for the cumulative sum in BLSS is found in Appendix B. The resulting values of t_n of the twelve points for each of the three sets of 'actual' values are in Figs. 3, 4, and 5. As expected, the first set of in control points shows a t-statistic that is well within the required limits. Also as predicted, the second set of out of control points shows that the cumulative sum catches trends that are obviously out of control. In this case, where there is such a large shift and all twelve points are outside of the regression chart control limits, a new model based on the last twelve points should be developed. The third set of data is more interesting. Although the regression control chart showed that points corresponding to runs #3, 11, and 12 are out of control, the cumulative sum shows no significant difference in the model for runs 1 through 10. This indicates that the error in sample #3 was not large enough to change the model significantly. However, runs #11 and 12 do result in a significant change.

The results of the regression analysis for data set three (Figure 6) show that after run #11, the coefficients C and D should be changed. (A change in coefficients occurs when the difference between the old and new values of the coefficients is greater than the standard error of the new value.) The regression is done on the 23 points that were used to generate the model plus the last 11 new points. When both C and D are changed to their new values, the cumulative t-statistic shows that the 'actual' data points are once again in control (Figure 7). In addition, the variance of each of the coefficients corresponding to the new model has decreased substantially. For example, in the original model, the standard error of the coefficient C was 520.8. In the revised model, the standard error is only 13.838. Thus, the model improves over time. Using the new model, run #12 is also now in control.

After the regression chart control limits are changed according to the new linear model (Figure 2(b)), we observe that although the runs #3, 11, and 12 are still out of control, the amount that they differ from the control limits has decreased. This shows that the second, revised model better predicts the deposition rate.

5. Conclusion

The analysis shows that the regression control chart in conjuction with the cumulative sum student-t are effective tools in reevaluating the model when a significant change is observed. Areas for future work include investigating cases in which the model slowly evolves until the final model is so significantly different from the original model that the user should be alerted. Another case in which an alarm should be sounded is when the data points are so far out of control that the furnace should be examined for problems. In the present analysis this situation is only detected by the regression control charts seen by the operator. More specifically, if all the points (as in data set #2) are out of control, the operator will know that something has gone wrong with the furnace.

Another area of interest is to implement the strategy presented while several recipes are being used and updated simultaneously.

References

[1] Lin, K.K. and C. Spanos, "Statistical Equipment Modeling for VLSI Manufacturing: An application for LPCVD," Dept. of Electrical Engineering and Computer Sciences, University of CA at Berkeley, 1989.

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

[3] Draper, N.R., and H. Smith, "Applied Regression Analysis" , John Wiley, New York, 1966.

[4] Chatterjee, Samprit, and Bertram Price, "Regression Analysis by Example" , John Wiley, New York, 1977.

APPENDIX A

Calculation for the variance of the sum of the residuals:

$$\text{var} \sum_{i=1}^{n} (Y_i' - y_i') \tag{1}$$

We begin with the deposition rate equation

$$\ln R_0 = A + B\ln P + C(1/T) + D(1/Q).$$

A change of variables $Y = \ln P$, $x_B = \ln P$, $x_C = (1/T)$, $x_D = (1/Q)$ results in more simple notation. The deposition rate equation becomes:

$$Y = A + Bx_B + Cx_C + Dx_D \tag{2}$$

and

$$\bar{Y} = A + B\bar{x}_B + C\bar{x}_C + D\bar{x}_D \tag{3}$$

Let

x_i' = new data
Y_i' = predicted value based on new data
y_i' = actual value that corresponds to the new data
x_i = data used to generate the model
$\bar{x}_i$ = average of the x_i values
N = the number of x_i values
n = the number of x_i' values
s_y = the standard error of the estimate of the regression based on original data

Substituting (2) into (1),

$$\sum_{i=1}^{n} (Y_1' - y_1') = (A + Bx_{B1}' + Cx_{C1}' + Dx_{D1}' - y_1') + \cdots + (A + Bx_{Bn}' + Cx_{Cn}' + Dx_{Dn}' - y_n') \tag{4}$$

Substituting (3) into (4) for constant A,

$$\sum_{i=1}^{n} (Y_i' - y_i') = [\bar{Y} - y_1' + B(x_{B1}' - \bar{x}_B) + C(x_{C1}' - \bar{x}_C) + D(x_{D1}' - \bar{x}_D)] + \cdots$$

$$+ [\bar{Y} - y_1' + B(x_{Bn}' - \bar{x}_B) + C(x_{Cn}' - \bar{x}_C) + D(x_{Dn}' - \bar{x}_D)]$$

Grouping terms,

$$\sum_{i=1}^{n} (Y_i' - y_i') = n\bar{y} - \sum_{i=1}^{n} y_i' + B\sum_{i=1}^{n} (x_{Bi}' - \bar{x}_B) + C\sum_{i=1}^{n} (x_{Ci}' - \bar{x}_C) + D\sum_{i=1}^{n} (x_{D_i}' - \bar{x}_D) \tag{5}$$

Let the standard error of the estimate of the regression based on original data be denoted by s_y. Substituting this value into (5), we obtain

$$\text{var} \sum_{i=1}^{n} (Y_i' - y_i') = \frac{n^2 s_y^2}{N} + n s_y^2 + \text{var}[B\sum_{i=1}^{n} (x_{Bi}' - \bar{x}_B)] + \text{var}[C\sum_{i=1}^{n} (x_{Ci}' - \bar{x}_C)] + \text{var}[D\sum_{i=1}^{n} (x_{D_i}' - \bar{x}_D)].$$

Thus,

$$\text{var}\sum_{i=1}^{n}(Y_i' - y_i') = \frac{n^2 s_y^2}{N} + ns_y^2 + \text{varB}[\sum_{i=1}^{n}(x_{Bi}' - \bar{x}_B)]^2 + \text{varC}[\sum_{i=1}^{n}(x_{Ci}' - \bar{x}_C)]^2 + \text{varD}[\sum_{i=1}^{n}(x_{D_i}' - \bar{x}_D)]^2$$

and the t-statistic based on the cumulative sum of the residuals is

$$t_n = \frac{\sum_{i=1}^{n}(Y_i' - y_i')}{\text{var}\sum_{i=1}^{n}(Y_i' - y_i')}$$

with N+n–4 degrees of freedom.

If n=1, we obtain the standard error of each residual, which is used to determine the control limits in the regression control chart.

$$\text{var}(Y_i' - y_i') = \frac{N+1}{N}s_y^2 + \text{varB}(x_{Bi}' - \bar{x}_B)^2 + \text{varC}(x_{Ci}' - \bar{x}_C)^2 + \text{varD}(x_{D_i}' - \bar{x}_D)^2$$

APPENDIX B

```
------------------------------------------------------------
BLSS computation of the cumulative sum student-t statistic
on 12 sample data points.

t =   (Y1-Y1act)/S

The 'actual' values were generated with a random number
generator (Gaussian distribution)
------------------------------------------------------------
```

```
. A=20.695 ;
. B=0.29346 ;
. C=-1.524e4 ;
. D = -48.584 ;

. Y1 = A + B*(log(P1)) + C*(1/T1) + D*(1/Q1) ;
```

```
. Y1res = Y1act - Y1
. temptab = k,P1,T1,Q1,Y1,Y1act,Y1res
```

run#	P	T	Q	Y	Yact	Yres
1	340.00	900.00	115.00	5.050	5.041	-0.008758
2	511.00	905.00	125.00	5.297	5.339	0.04233
3	538.00	890.00	175.00	5.139	5.184	0.04472
4	295.00	927.00	100.00	5.438	5.326	-0.1121
5	294.00	927.00	100.00	5.437	5.453	0.01606
6	466.00	883.00	207.00	5.004	5.033	0.02898
7	339.00	895.00	175.00	5.099	5.124	0.02486
8	537.00	899.00	100.00	5.102	5.013	-0.08868
9	517.00	902.00	125.00	5.244	5.281	0.03691
10	298.00	900.00	100.00	4.948	4.920	-0.02769
11	316.00	904.00	100.00	5.040	5.040	1.67e-04
12	427.00	881.00	200.00	4.931	4.977	0.04602

```
. dat1 = log(temptab[2]),1/(temptab[3]),1/(temptab[4]) ;
. show dat1 (shape)
```

ln(P)	1/T	1/Q
5.829	0.001111	0.008696
6.236	0.001105	0.008000
6.288	0.001124	0.005714
5.687	0.001079	0.01000
5.684	0.001079	0.01000
6.144	0.001133	0.004831
5.826	0.001117	0.005714
6.286	0.001112	0.01000
6.248	0.001109	0.008000
5.697	0.001111	0.01000
5.756	0.001106	0.01000
6.057	0.001135	0.005000

```
. dat1=dat1' ;
```

```
. means

   6.134  0.001112  0.007813
```

```
. dat1a = dat1 - means ;

. ltri (dims=12) > uppertri
. uppertri = uppertri'
. dat1b = dat1a#*uppertri ;
. dat1b=dat1b' ;
. show dat1b (shape)
```

(x - x)	(x - x)	(x - x)
-0.3049	-7.29e-07	8.82e-04
-0.2023	-7.60e-06	0.001069
-0.04832	4.16e-06	-0.001030
-0.4952	-2.89e-05	0.001157
-0.9454	-6.20e-05	0.003344
-0.9351	-4.14e-05	3.62e-04
-1.243	-3.59e-05	-0.001737
-1.091	-3.54e-05	4.49e-04
-0.9765	-3.86e-05	6.36e-04
-1.413	-3.93e-05	0.002823
-1.791	-4.49e-05	0.005010
-1.868	-2.17e-05	0.002197

```
. k=1:12:1 ;
. k=k' ;
. varY=0.05808^2 ;

. nsq=k^2 ;
. const1 = nsq*varY/23 ;
. const1 = const1 + k*varY ;
. dat1c=dat1b^2 ;
. dat1c=dat1c' ;
. vars

 0.003008  2.71e+05    27.76

. dat1d = vars * dat1c ;
. dat1d = dat1d' ;
. show dat1d (shape)

 2.80e-04  1.44e-07  2.16e-05
 1.23e-04  1.57e-05  3.17e-05
 7.02e-06  4.69e-06  2.94e-05
 7.37e-04  2.27e-04  3.72e-05
 0.002688  0.001043  3.10e-04
 0.002630  4.64e-04  3.63e-06
 0.004646  3.49e-04  8.38e-05
 0.003578  3.39e-04  5.61e-06
 0.002868  4.03e-04  1.12e-05
 0.006007  4.19e-04  2.21e-04
 0.009651  5.48e-04  6.97e-04
  0.01050  1.28e-04  1.34e-04
```

```
. dat1e = const1 + dat1d[1] + dat1d[2] + dat1d[3] ;
. dat1e (shape)

 0.003821
 0.007504
  0.01148
  0.01684
  0.02458
  0.02862
```

```
  0.03588
  0.04030
  0.04552
  0.05505
  0.06575
  0.07236

. datlf = sqrt(datle)


. res = temptab[7]' #* uppertri
. res = res'
. tn = res/datlf
. restab = res,datlf,tn,studt
. restab [shape]
```

(res)	S	tn	studt	df
-0.008758	0.06182	-0.1417	2.074	22
0.03357	0.08662	0.3875	2.069	23
0.07829	0.1071	0.7307	2.064	24
-0.03377	0.1298	-0.2602	2.060	25
-0.01770	0.1568	-0.1129	2.056	26
0.01127	0.1692	0.06663	2.052	27
0.03613	0.1894	0.1907	2.048	28
-0.05255	0.2007	-0.2618	2.045	29
-0.01564	0.2134	-0.07332	2.042	30
-0.04334	0.2346	-0.1847	2.040	31
-0.04317	0.2564	-0.1684	2.037	32
0.002850	0.2690	0.01059	2.035	33

ORIGINAL RUNS

run#	P (mtorr)	T (K)	Q (sccm)	Y ln(A/min)	Yact ln(A/min)	Yact-Y ln(A/min)
1	339	882	125	4.734	4.696	-0.03845
2	318	926	100	5.438	5.553	0.1147
3	549	881	125	4.855	4.796	-0.05871
4	548	897	250	5.354	5.321	-0.03259
5	427	882	175	4.911	4.919	0.007974
6	548	888	250	5.182	5.157	-0.02497
7	538	882	100	4.772	4.727	-0.04545
8	366	926	125	5.575	5.617	0.04203
9	517	927	175	5.802	5.751	-0.05148
10	296	882	100	4.599	4.560	-0.03918
11	547	927	100	5.613	5.568	-0.04525
12	547	927	125	5.709	5.693	-0.01619
13	548	879	100	4.719	4.810	0.09098
14	295	883	100	4.618	4.576	-0.04170
15	537	927	225	5.874	5.872	-0.002398
16	548	927	100	5.614	5.620	0.006221
17	552	883	250	5.087	5.234	0.1468
18	294	881	100	4.578	4.564	-0.01366
19	546	881	100	4.757	4.794	0.03682
20	466	900	200	5.315	5.336	0.02094
21	546	907	100	5.251	5.247	-0.004409
22	465	897	207	5.266	5.247	-0.01911
23	545	904	100	5.195	5.209	0.01370

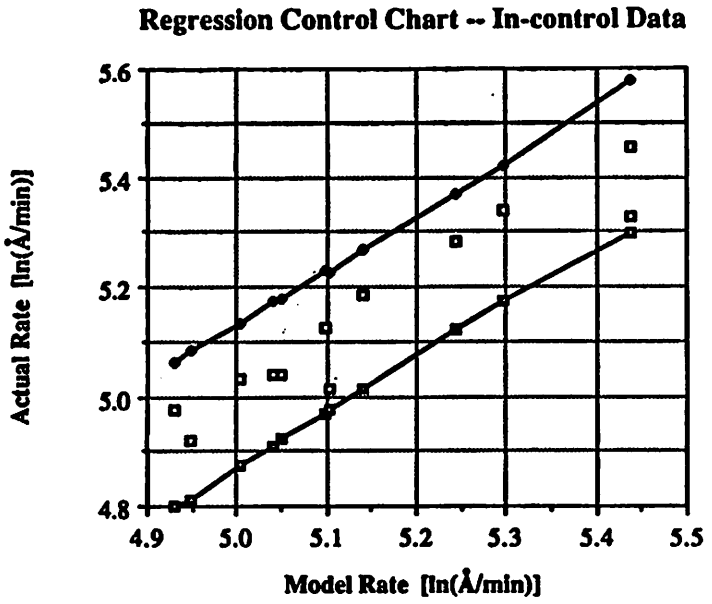
REGRESSION ON THE LINEAR MODEL
TO OBTAIN THE FOUR COEFFICIENTS

Dependent variable: treg[5]
Independent variables: treg[1 2 3]
Observations 23 Parameters 4

Parameter	Estimate	SE	t-Ratio	P-Value
A	20.695	0.73134	28.2969	0.0000
B	0.29346	0.054841	5.3510	0.0000
C	-1.524e+04	520.80	-29.2672	0.0000
D	-48.584	5.2689	-9.2208	0.0000

Residual SD 0.058084 Residual Variance 0:0033737
Multiple R 0.99168 Multiple R-squared 0.98342

Figure 1



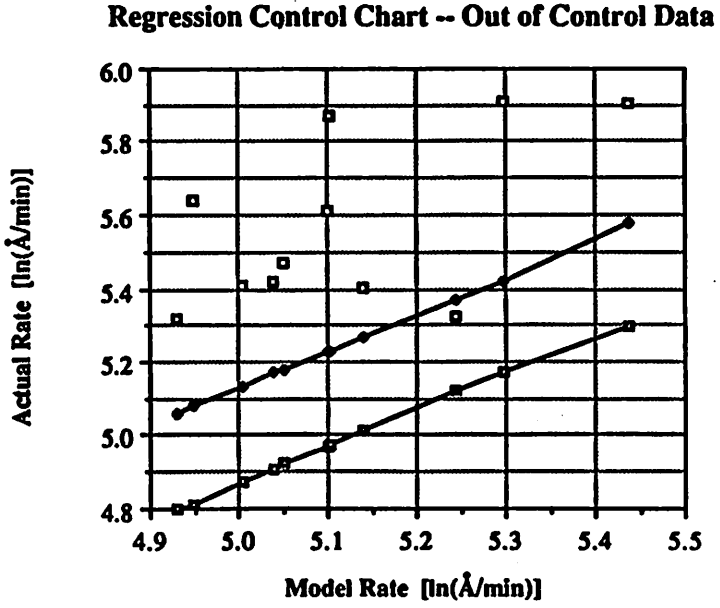
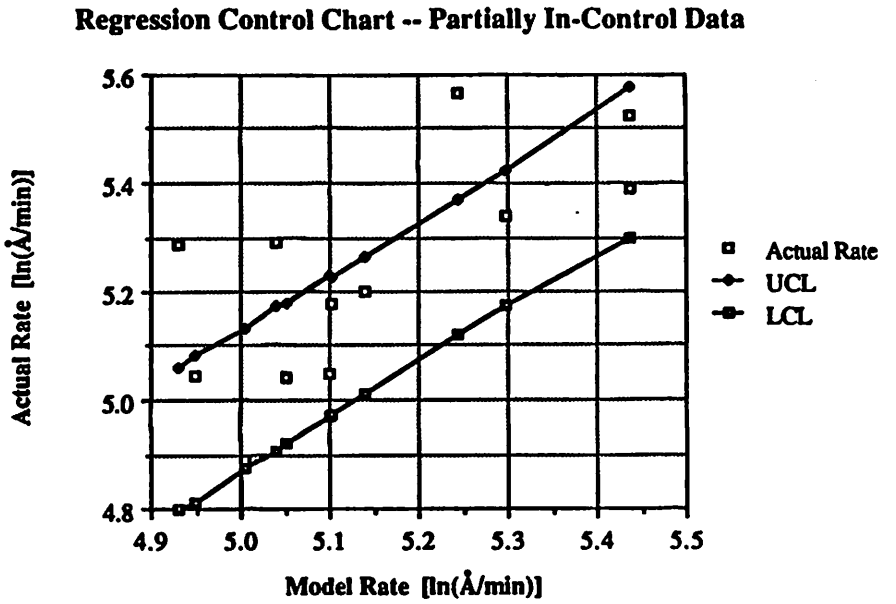


Figure 2(a)



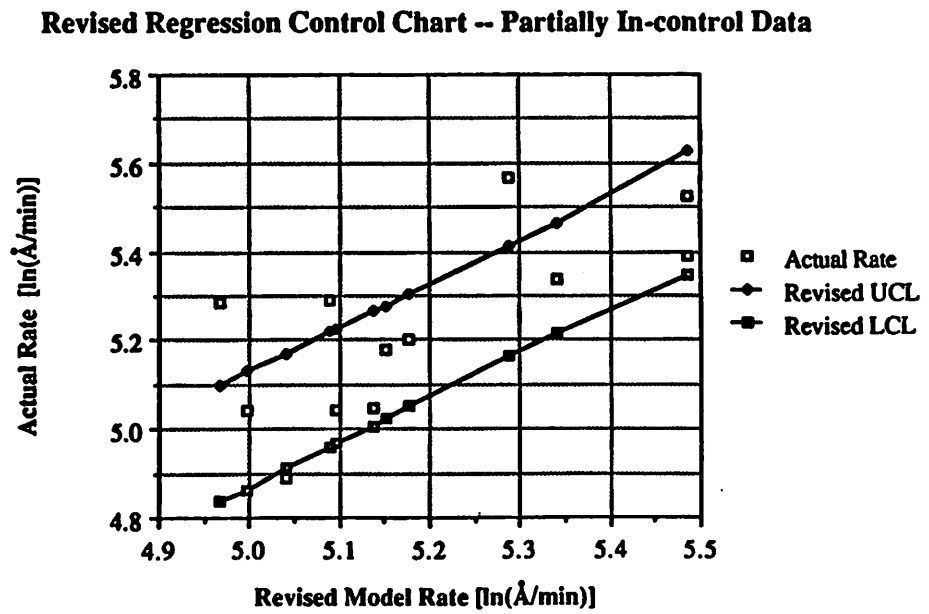


Figure 2(b)

BLSS computation of the cumulative sum student-t statistic on 12 sample data points.

The 'actual' values were generated with a random number generator (Gaussian distribution)

A=20.695
B=0.29346
C=-1.524e4
D = -48.584

Y1 = A + B*(log(P1)) + C*(1/T1) + D*(1/Q1)

Y1res = Y1act - Y1

run#	P	T	Q	Y	Yact	Yres
1	340.00	900.00	115.00	5.050	5.041	-0.008758
2	511.00	905.00	125.00	5.297	5.339	0.04233
3	538.00	890.00	175.00	5.139	5.184	0.04472
4	295.00	927.00	100.00	5.438	5.326	-0.1121
5	294.00	927.00	100.00	5.437	5.453	0.01606
6	466.00	883.00	207.00	5.004	5.033	0.02898
7	339.00	895.00	175.00	5.099	5.124	0.02486
8	537.00	899.00	100.00	5.102	5.013	-0.08868
9	517.00	902.00	125.00	5.244	5.281	0.03691
10	298.00	900.00	100.00	4.948	4.920	-0.02769
11	316.00	904.00	100.00	5.040	5.040	1.67e-04
12	427.00	881.00	200.00	4.931	4.977	0.04602

(res)	S	tn	studt	df
-0.008758	0.06182	-0.1417	2.086	20
0.03357	0.08662	0.3875	2.080	21
0.07829	0.1071	0.7307	2.074	22
-0.03377	0.1298	-0.2602	2.069	23
-0.01770	0.1568	-0.1129	2.064	24
0.01127	0.1692	0.06663	2.060	25
0.03613	0.1894	0.1907	2.056	26
-0.05255	0.2007	-0.2618	2.052	27
-0.01564	0.2134	-0.07332	2.048	28
-0.04334	0.2346	-0.1847	2.045	29
-0.04317	0.2564	-0.1684	2.042	30
0.002850	0.2690	0.01059	2.040	31

Figure 3

A large shift in deposition rate --

A=20.695
B=0.29346
C=-1.524e4
D = -48.584
Y = A + B*(log(P)) + C*(1/T) + D*(1/Q)

run#	P	T	Q	Ymod	Yact	res
1.000	340.00	900.00	115.00	5.050	5.473	0.4232
2.000	511.00	905.00	125.00	5.297	5.908	0.6113
3.000	538.00	890.00	175.00	5.139	5.400	0.2610
4.000	295.00	927.00	100.00	5.438	5.902	0.4641
5.000	294.00	927.00	100.00	5.437	5.906	0.4691
6.000	466.00	883.00	207.00	5.004	5.407	0.4032
7.000	339.00	895.00	175.00	5.099	5.611	0.5119
8.000	537.00	899.00	100.00	5.102	5.873	0.7713
9.000	517.00	902.00	125.00	5.244	5.325	0.08091
10.00	298.00	900.00	100.00	4.948	5.639	0.6913
11.00	316.00	904.00	100.00	5.040	5.421	0.3812
12.00	427.00	881.00	200.00	4.931	5.321	0.3900

(res)	S	tn	studt	df
0.4232	0.06182	6.847	2.086	20
1.035	0.08662	11.94	2.080	21
1.296	0.1071	12.09	2.074	22
1.760	0.1298	13.56	2.069	23
2.229	0.1568	14.22	2.064	24
2.632	0.1692	15.56	2.060	25
3.144	0.1894	16.60	2.056	26
3.915	0.2007	19.50	2.052	27
3.996	0.2134	18.73	2.048	28
4.687	0.2346	19.98	2.045	29
5.068	0.2564	19.77	2.042	30
5.458	0.2690	20.29	2.040	31

Figure 4

Sample data points

On the 11th sample, the student-t statistic is exceeded.

A=20.695
B=0.29346
C=-1.524e4
D = -48.584

Y1 = A + B*(log(P1)) + C*(1/T1) + D*(1/Q1) ;
Y1res = Y1act - Y1

Yact	Ymod	res
5.041	5.050	-0.008758
5.339	5.297	0.04233
5.201	5.139	0.06198
5.526	5.438	0.08807
5.389	5.437	-0.04794
4.890	5.004	-0.1140
5.048	5.099	-0.05084
5.178	5.102	0.07632
5.567	5.244	0.3229
5.045	4.948	0.09731
5.289	5.040	0.2492
5.286	4.931	0.3550

(res)	S	tn	studt	df
-0.008758	0.06182	-0.1417	2.086	20
0.03357	0.08662	0.3875	2.080	21
0.09555	0.1071	0.8918	2.074	22
0.1836	0.1298	1.415	2.069	23
0.1357	0.1568	0.8655	2.064	24
0.02166	0.1692	0.1280	2.060	25
-0.02918	0.1894	-0.1541	2.056	26
0.04714	0.2007	0.2348	2.052	27
0.3700	0.2134	1.734	2.048	28
0.4674	0.2346	1.992	2.045	29
0.7165	0.2564	2.794	2.042	30
1.072	0.2690	3.983	2.040	31

Figure 5

DETERMINE THE NEW COEFFICIENTS

REGRESSION WITH B,C,and D HELD FIXED, VARY A

Dependent variable: Yprime1
Independent variable: a
Observations 34 Parameters 1

Parameter	Estimate	SE	t-Ratio	P-Value
A	20.714	0.015365	1.3481e+03	0.0000

Residual SD 0.089595 Residual Variance 0.0080273
Multiple R 0.99999 Multiple R-squared 0.99998

REGRESSION WITH A,C,and D HELD FIXED, VARY B

Dependent variable: Yprime1
Independent variable: b
Observations 34 Parameters 1

Parameter	Estimate	SE	t-Ratio	P-Value
B	0.29656	0.0025259	117.4100	0.0000

Residual SD 0.089642 Residual Variance 0.0080357
Multiple R 0.99881 Multiple R-squared 0.99761

REGRESSION WITH A,B, and D HELD FIXED, VARY C

Dependent variable: Yprime1
Independent variable: dat1prime[2]
Observations 34 Parameters 1

Parameter	Estimate	SE	t-Ratio	P-Value
C	-1.522e+04	13.838	-1.1000e+03	0.0000

Residual SD 0.089626 Residual Variance 0.0080328
Multiple R 0.99999 Multiple R-squared 0.99997

REGRESSION WITH A,B, and C HELD FIXED, VARY D

Dependent variable: Yprime1
Independent variable: d
Observations 34 Parameters 1

Parameter	Estimate	SE	t-Ratio	P-Value
D	-45.811	1.8368	-24.9405	0.0000

Residual SD 0.088656 Residual Variance 0.0078598
Multiple R 0.97448 Multiple R-squared 0.94962

Figure 6

Calculated student-t for the twelve sample data points (in set #3), after changing the coefficients.

A=20.695
B=0.29346
C=-1.522e4
D = -45.811

Y1 = A + B*(log(P1)) + C*(1/T1) + D*(1/Q1) ;
Y1res = Y1act - Y1

Yact	Ymod	res
5.041	5.096	-0.05509
5.339	5.341	-0.001959
5.201	5.177	0.02367
5.526	5.487	0.03876
5.389	5.486	-0.09724
4.890	5.040	-0.1501
5.048	5.137	-0.08903
5.178	5.152	0.02634
5.567	5.288	0.2786
5.045	4.998	0.04735
5.289	5.090	0.1993
5.286	4.968	0.3185

(res)	S	tn	studt	df
-0.05509	0.06182	-0.8912	2.086	20
-0.05705	0.08662	-0.6586	2.080	21
-0.03339	0.1071	-0.3116	2.074	22
0.005378	0.1298	0.04144	2.069	23
-0.09186	0.1568	-0.5860	2.064	24
-0.2419	0.1692	-1.430	2.060	25
-0.3310	0.1894	-1.747	2.056	26
-0.3046	0.2007	-1.518	2.052	27
-0.02607	0.2134	-0.1222	2.048	28
0.02128	0.2346	0.09071	2.045	29
0.2206	0.2564	0.8603	2.042	30
0.5390	0.2690	2.004	2.040	31

Figure 7

The Effects of Wafer Orientation on Oxide Breakdown

Elyse Rosenbaum

Abstract

Circuits that integrate silicon with compound semiconductor devices have some distinct performance advantages over traditional IC families. These technologies might require the growth of high quality oxides on off-axis silicon substrates. To study the reliability of these oxides, an experimental study has been completed and the results are reported here.

1. Introduction

MOS circuits are typically fabricated on silicon wafers which have been cut along the (100) plane. Recently, there has be interest in evaluating the effect of using wafers which are not cut along one of the major crystallographic planes. If reliable circuits can be fabricated on "off-axis" substrates, integration of silicon and gallium arsenide devices will be possible as quality GaAs films may be grown on off-axis Si substrates.

This study reports the effect of substrate rotation on oxide breakdown of large-area capacitors (.01 cm^2). The capacitors were fabricated on substrates which were rotated at various angles off the (100) plane around the <011> axis. Two different processes were used to grow the capacitor oxides; an 8 minute 850° steam oxidation and a 100 minute 850° dry oxidation. The post-oxidation process flows were identical (that of a 4 mask NMOS process). It was discovered that increasing the angle of rotation increased the probability of oxide breakdown at low electric fields while the choice of oxidizing ambient did not have a significant effect.

2. Effects of treatments on breakdown voltage

A summary of the experiment [1] follows:

Run	Oxide	Angle of rotation	Oxide thickness
0C2	wet	0°	17.0 nm
0C3	dry	0°	15.5 nm
4C1	wet	4°	17.5 nm
5C1	dry	5°	15.7 nm
6C1	wet	6°	18.2 nm
7C1	dry	7°	16.0 nm
8C1	wet	8°	18.8 nm

The non-constant oxide thickness values indicate that oxidation rate increases with offset angle; this variation in oxide thickness will necessitate the use of breakdown electric field rather than breakdown voltage as the parameter of interest.

Ramp voltage breakdown statistics were collected for .01 cm^2 capacitors from each wafer. The ramp rate was .4 V/sec and the resolution was .2 V. The raw data is included in Appendix A. The data was characterized by the median value of breakdown electric field and standard deviation [2]. The choice of median rather than mean is somewhat arbitrary when one is using an empirical data distribution. The median has the advantage of being "robust" against values at either tail of the distribution; that is, it more accurately represents the "typical" devices.

Run	Median breakdown field	standard deviation	# of devices tested
0C2	12.35 MV/cm	2.62	81
0C3	9.68	1.98	50
4C1	10.97	2.13	104
5C1	9.94	2.87	43
6C1	8.46	1.53	59
7C1	5.88	1.33	50
8C1	7.77	1.10	52

Fig. 1 illustrates that the magnitudes of the standard deviation and median are correlated. This indicates that the values of breakdown field are not distributed in an IIND manner around each treatment mean. (Recall that mean and median are equal for normal distributions.)

However, an ANOVA analysis to evaluate the effects of the various treatments can be performed when a variance stabilizing transformation is used [3]. The standard deviation is roughly proportional to the 3/2 power of the median (Fig. 2). This indicates that the data points should be transformed to the -1/2 power. Parameters derived from the transformed data sets follow.

Run	$[E^{\frac{-1}{2}}]_{50}$	standard deviation
0C2	.285	.041
0C3	.321	.038
4C1	.302	.029
5C1	.317	.066
6C1	.344	.032
7C1	.413	.040
8C1	.359	.021

Fig. 3 shows no apparent relationship between the transformed medians and standard deviations. An estimate of the within-treatment variance is obtained by calculating $\sum_1^7 \nu_i \sigma_i^2 \; / \; \sum_1^7 \nu_i$. The variance is estimated to be .0015 and the standard deviation .038. The oxide process effect (wet vs dry) is

$$\frac{0C2 + 4C1 + 6C1 + 8C1}{4} - \frac{0C3 + 5C1 + 7C1}{3} \text{ +/- S.D.} \qquad (1)$$

The oxide process effect is calculated to be -.028 +/-.038. This effect is not significant when compared to the within-treatment standard deviation.

The off-axis effect is evaluated by comparing the 0° and 7° treatments (for the wet oxides, an average of the 6° and 8° values is used). The off-axis effect is

$$\frac{0C2 + 0C3}{2} - \frac{7C1 + \frac{6C1 + 8C1}{2}}{2} \text{ +/- S.D.} \qquad (2)$$

The off-axis effect is calculated to be -.079 +/-.038. This effect is a bit larger than 2 S.D. and is thus judged to be significant. It should be noted that since the treatment medians are only separated by about 2 S.D., there is to be expected a fair amount of overlap between 0° and 7° breakdown field values.

3. Spatial distribution of oxide defects

The experimental finding that all of the capacitors on one wafer do not short circuit at the same voltage indicates that there are defects of various severities present. The simplest model for the spatial distribution of defects is the Poisson model. Each value of breakdown field is associated with a particular defect severity. Using the Poisson model, the probability of a capacitor containing an 8 MV/cm type defect is independent of the probability of its containing a 9 MV/cm type defect. The probability that a capacitor contains an 8 MV/cm type defect OR a 9 MV/cm type defect (or both) may be found by convolving the

defects' probability mass functions (pmf) to obtain the joint pmf. The joint pmf for two (or more) poisson random variables is a poisson pmf with parameter equal to the sum of the individual pmf parameters.

The Poisson parameter needed to evaluate the probability that a capacitor fails at a specific breakdown field ("ebd") may be derived as follows.

$P(E_{bd} = ebd) = P$ (at least one ebd–type defect in oxide) * P (no defects in oxide which causeE_{bd} lt ebd)

$$P(E_{bd} = ebd) = (1-e^{-\lambda})e^{-\lambda'} \quad (3)$$

where λ is the parameter for the pmf of the defect associated with ebd and λ' is the parameter for the joint pmf of all defects which cause E_{bd} lt ebd.

Figs. 4-7 show the derived values of the Poisson parameter (λ) at each "ebd" for the four wet oxide wafers. Superimposed are smooth curves indicating the general effect on λ(ebd) of increasing the off-axis angle. One sees that for the 0° case, λ is fairly constant at all "ebd". As the off-axis angle is increased, λ increases approximately linearly with E_{bd} and the slope of this line increases with off-axis angle. As the angle of rotation is increased, a value of λ can not be obtained for the highest values of breakdown field. This does not imply that there are no defects corresponding to those values of E_{bd}; instead, it indicates that there is a very high probability that each capacitor contains a severe defect which masks the effect of less severe defects.

By comparing breakdown statistics for capacitors of different areas, one may determine if the above analysis, based on the assumption that the defects are uniformly distributed (Poisson model), is valid. This type of data is available for the 0C2 and 8C1 runs. The parameter λ', defined above, is set equal to the capacitor area multiplied by the defect density (number of defects per unit area). If the defects are clustered rather than distributed uniformly, the defect density one derives will become smaller as the capacitor area becomes larger. (The following defect density analysis was restricted to fields below 10 MV/cm because above this value the relationship between breakdown field and defect severity is not as clear.)

Fig. 8 shows the derived defect densities for .01 cm^2 and .04 cm^2 capacitors from the 0C2 wafer. There is no evidence of clustering. In fact, the larger capacitors have a slightly higher derived defect density at most breakdown fields. One observes that the defect density curves for the capacitor types are very similar except for a horizontal offset. This might indicate that a constant error was made in measuring the voltage during one or the other set of measurements.

Fig. 9 shows the derived defect densities for .01 cm^2, .000625 cm^2 and .0001 cm^2 capacitors from the 8C1 wafer. There were very few occurrences of breakdown at fields lower than 10 MV/cm for the .0001 cm^2 capacitors; this indicates that this is too small a capacitor area to do a defect-related breakdown study upon. The 8C1 data does show evidence of clustering. Fig. 10 compares the actual .01 cm^2 data with that predicted from the .000625 cm^2 data using both the Poisson and modified Poisson models. The Poisson model predicts

$$P(\text{Failure at a given } V_{bd}) = 1 - e^{-AD_o} \quad (4)$$

where D_o is the defect density derived for a capacitor of area A_o and A is the area of the capacitor for which predictions are being made. The modified Poisson model accounts for clustering, it is described by

$$P(\text{Failure at a given } V_{bd}) = 1 - \exp\left\{-A_o D_o \left[\frac{A}{A_o}\right]^{1-b}\right\} \quad (5)$$

A "b" value of .25 is found to fit this data set reasonably well.

The non-zero value of the cluster parameter (b) indicates that the analysis of defect distribution as a function of E_{bd} (Figs. 4-7) was overly simplistic, at least for the 8C1 wafer. Specifically, by assuming that the defects are independently distributed, we have probably underestimated the actual value of λ (the mean number of defects for a specific ebd), particularly for the defects corresponding to high values of breakdown field (lesser severity defects). However, the modified Poisson model implies that our calculations of λ' (the joint pmf parameter) were correct. Since the cluster parameter is fairly small, Figs. 4-7 remain informative.

4. Conclusions

Off-axis substrates have been conclusively shown to increase the probability that a device will contain a severe oxide defect. Clustering of defects is seen in the off-axis samples but not in the samples fabricated on the (100) plane.

5. Acknowledgments

J. Chung fabricated the devices and performed the breakdown measurements. I thank him for his effort.

References

[1] J.C. Chung, et al., "The effects of off-axis substrate orientation on MOSFET characteristics," IEDM Technical Digest, p. 663, Dec. 1989.

[2] W.B. Joyce, "Generic parameterization of lifetime distributions," IEEE Trans. on Elec. Dev., Vol. 36, No. 7, p. 1389, July 1989.

[3] G. Box et al., "Statistics for experimenters", Wiley Interscience, 1978.

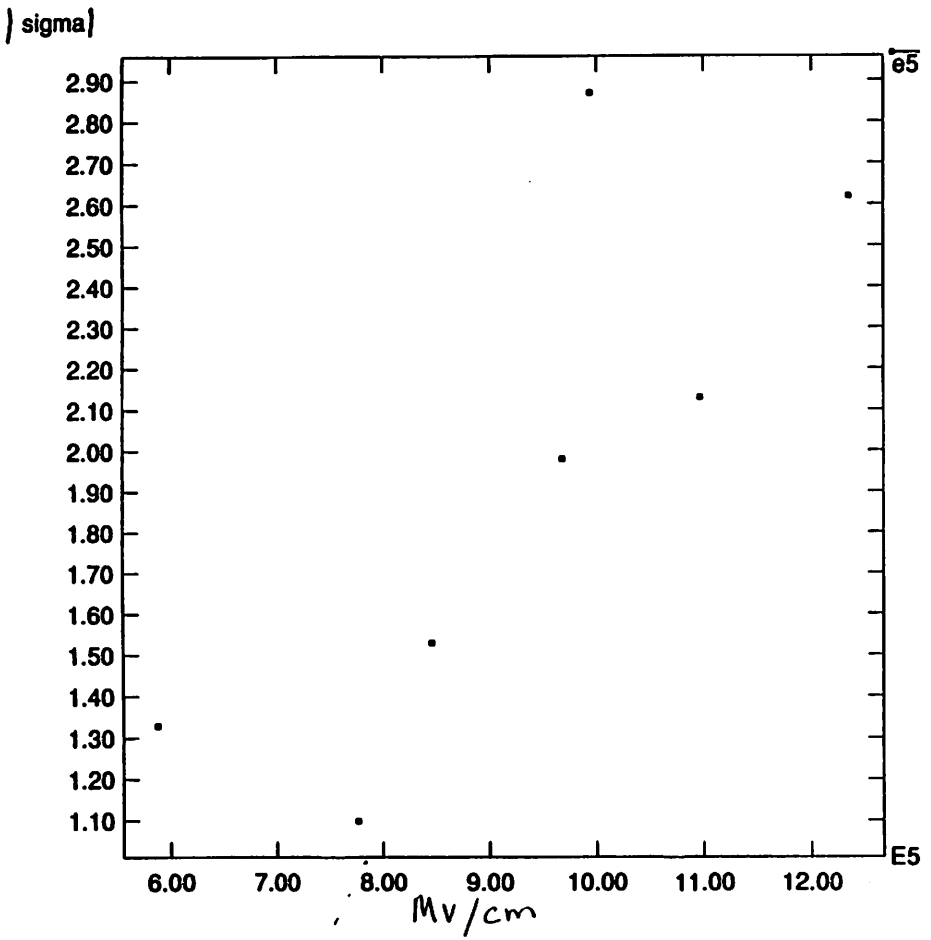


Figure 1: σ is not random with respect to the median value of the breakdown field.

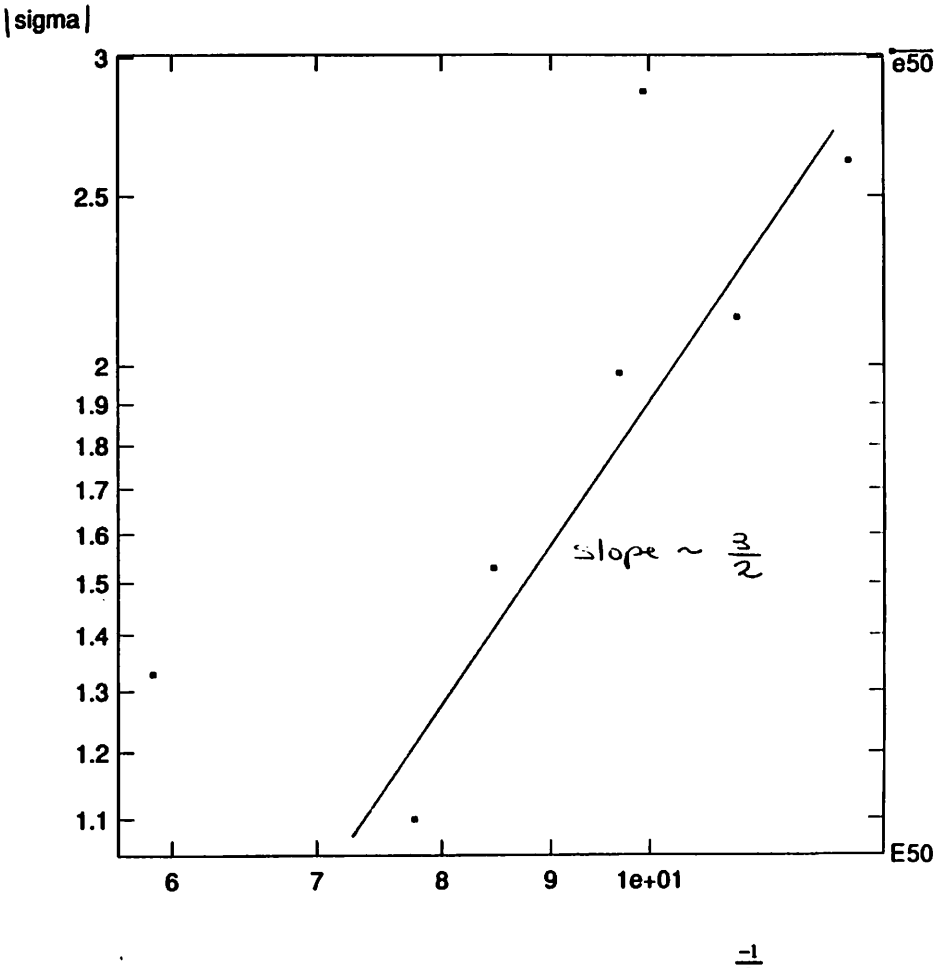


Figure 2: Slope = α, where σ is proportional to $[E^{\frac{-1}{2}}]_{50}$.

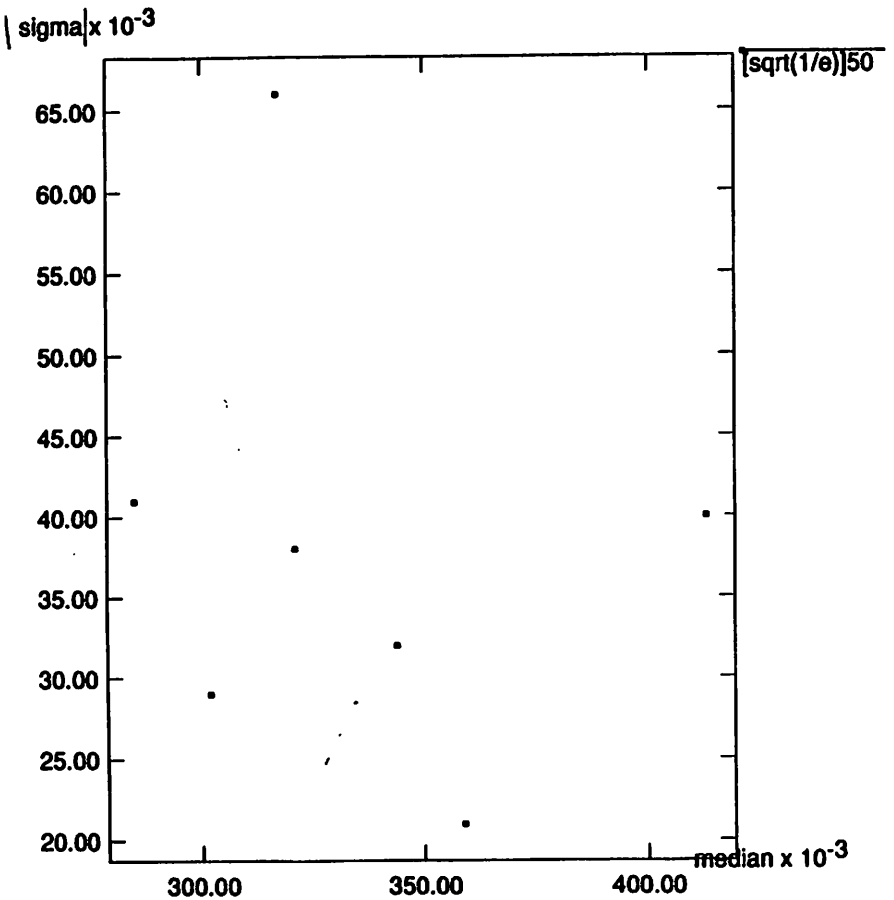


Figure 3: Transformed data shows homoscedacity.

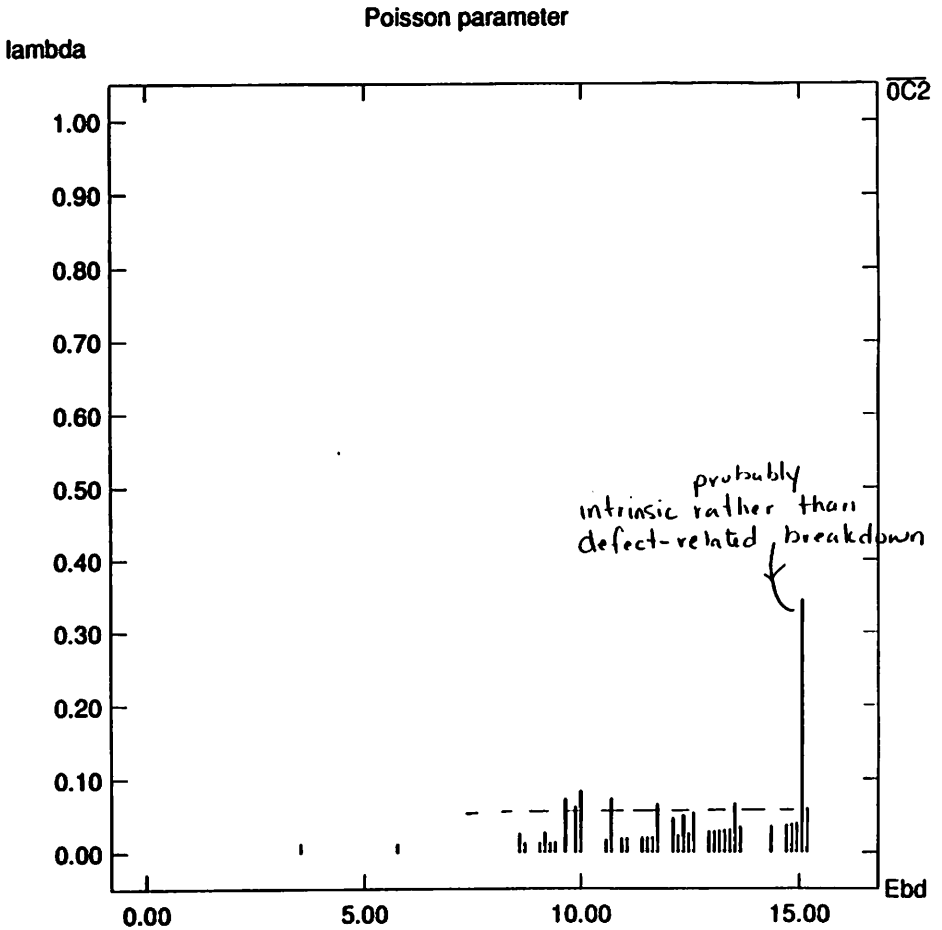


Figure 4: λ (mean) causing specific E_{bd} - 0° offset.

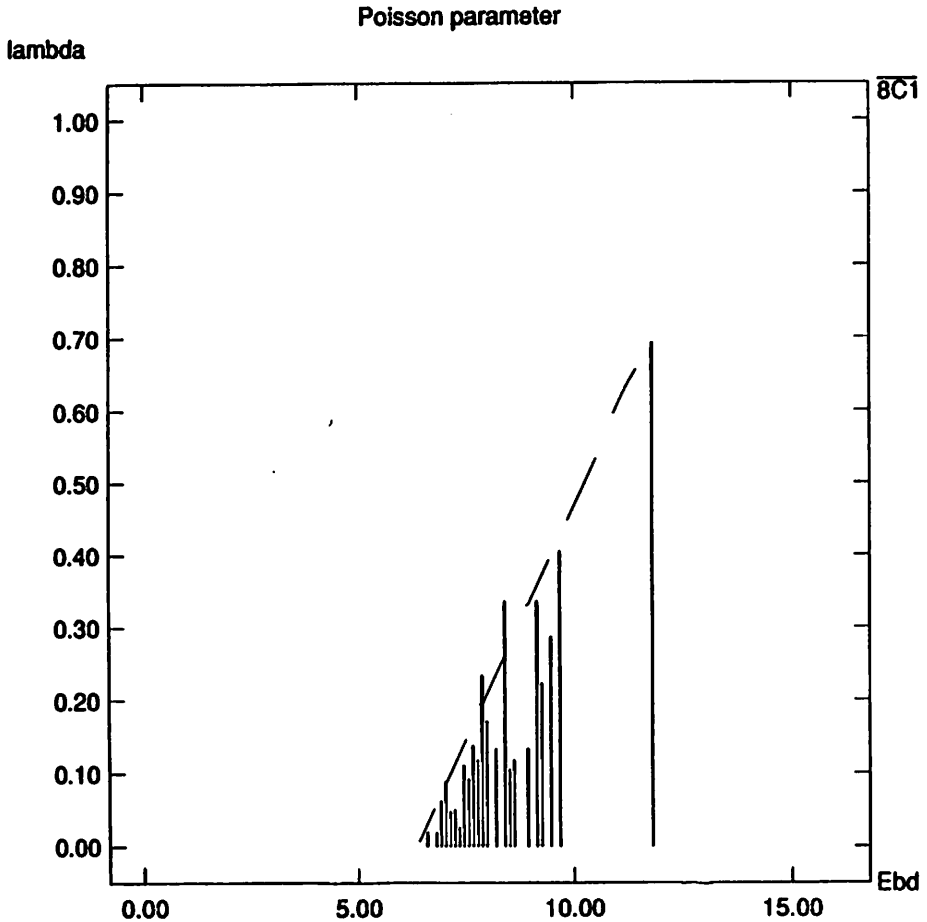


Figure 7: 8° offset.

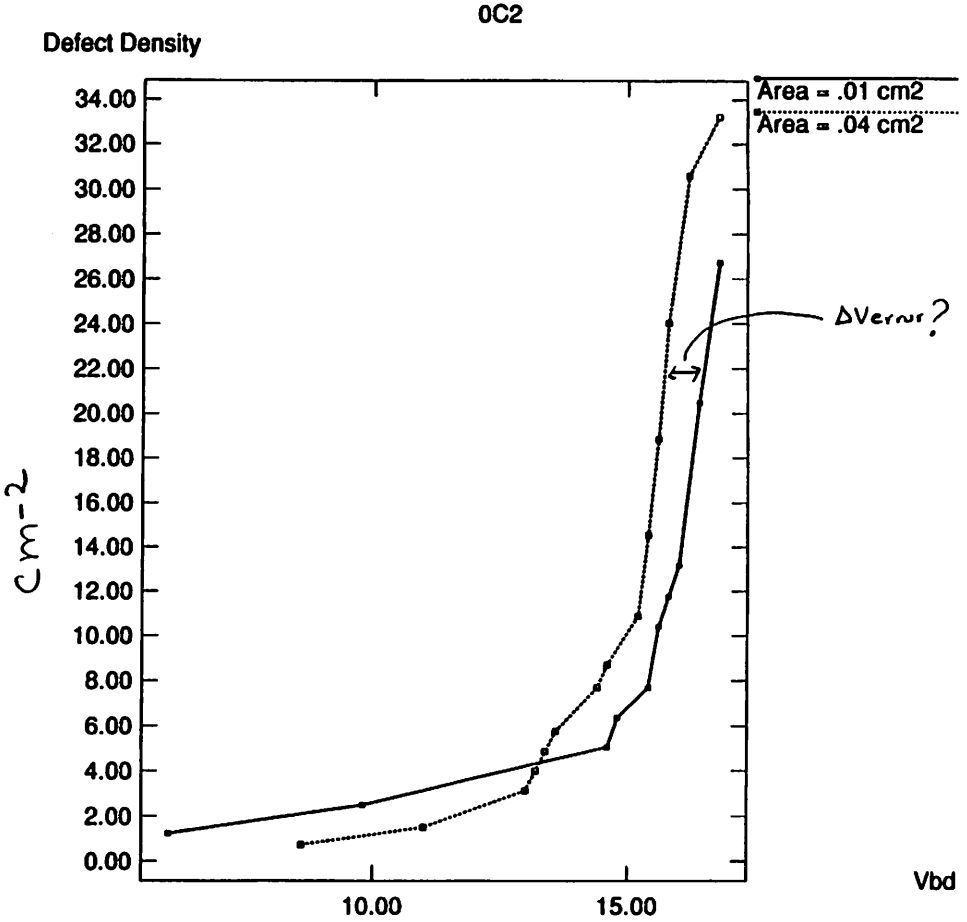


Figure 8: λ' = AD. Derived defect density for on-axis wafer.

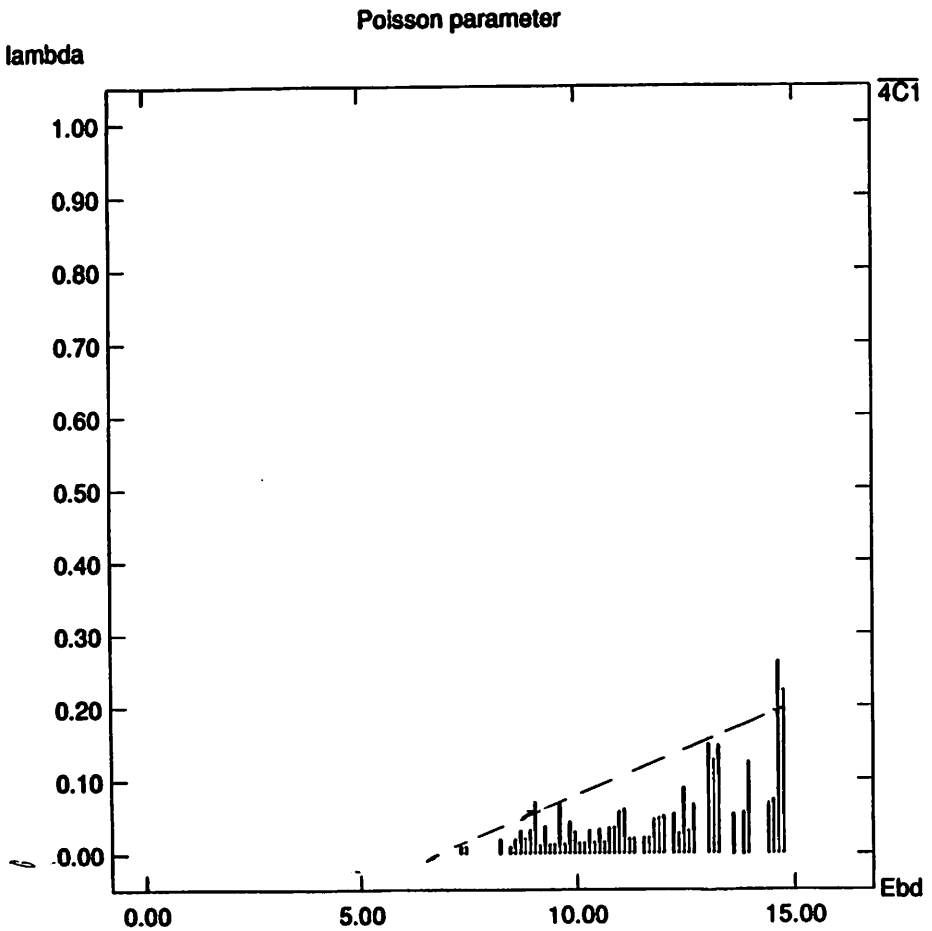


Figure 5: 4° offset.

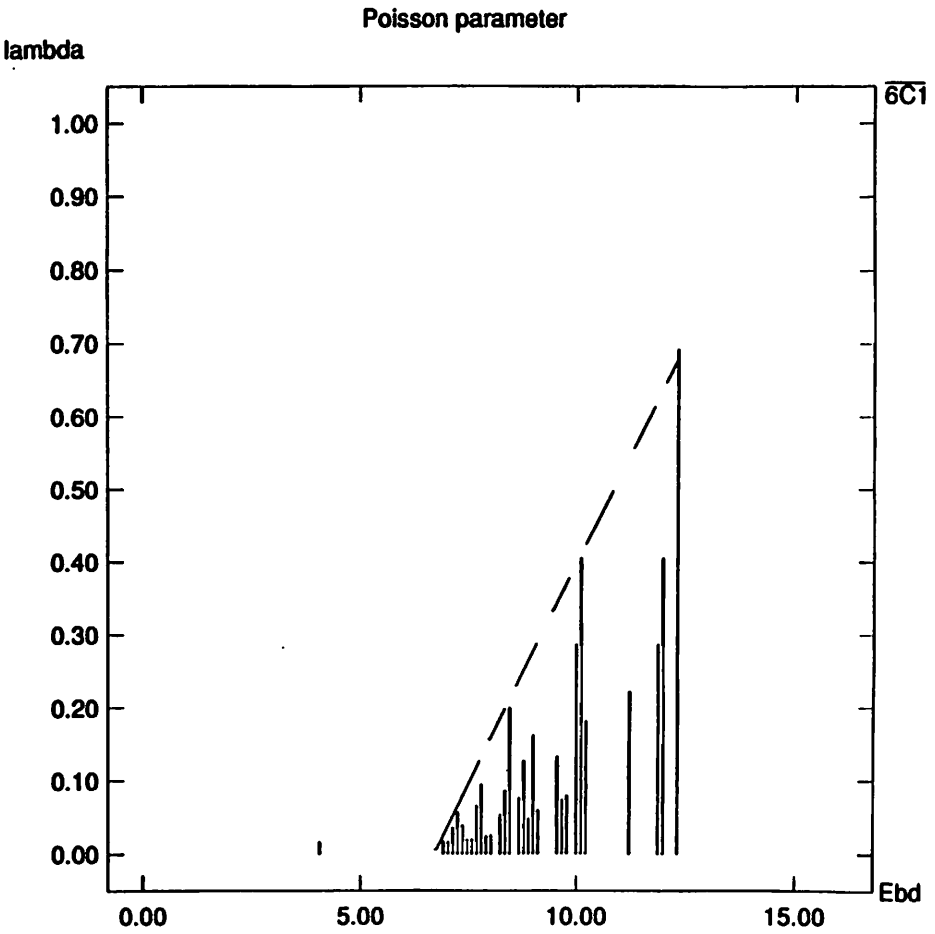


Figure 6: 6° offset.

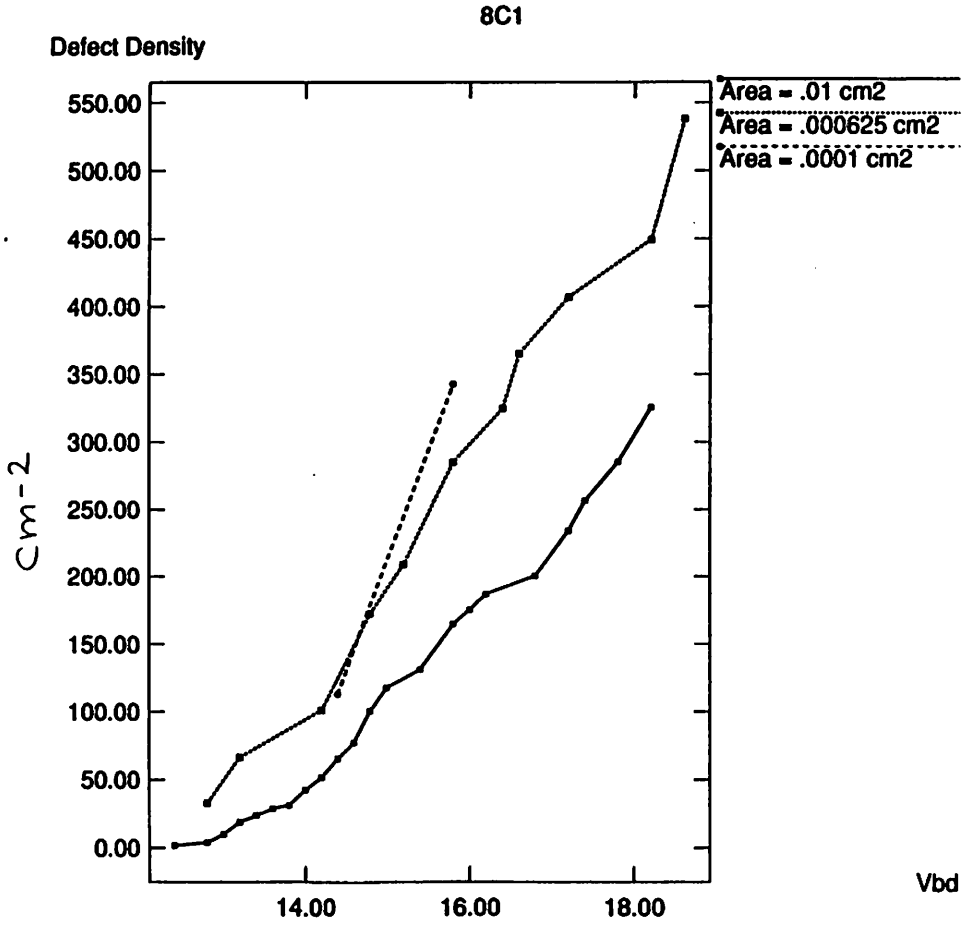


Figure 9: λ' = AD. Derived defect density 8° off-axis wafer evidences clustering..

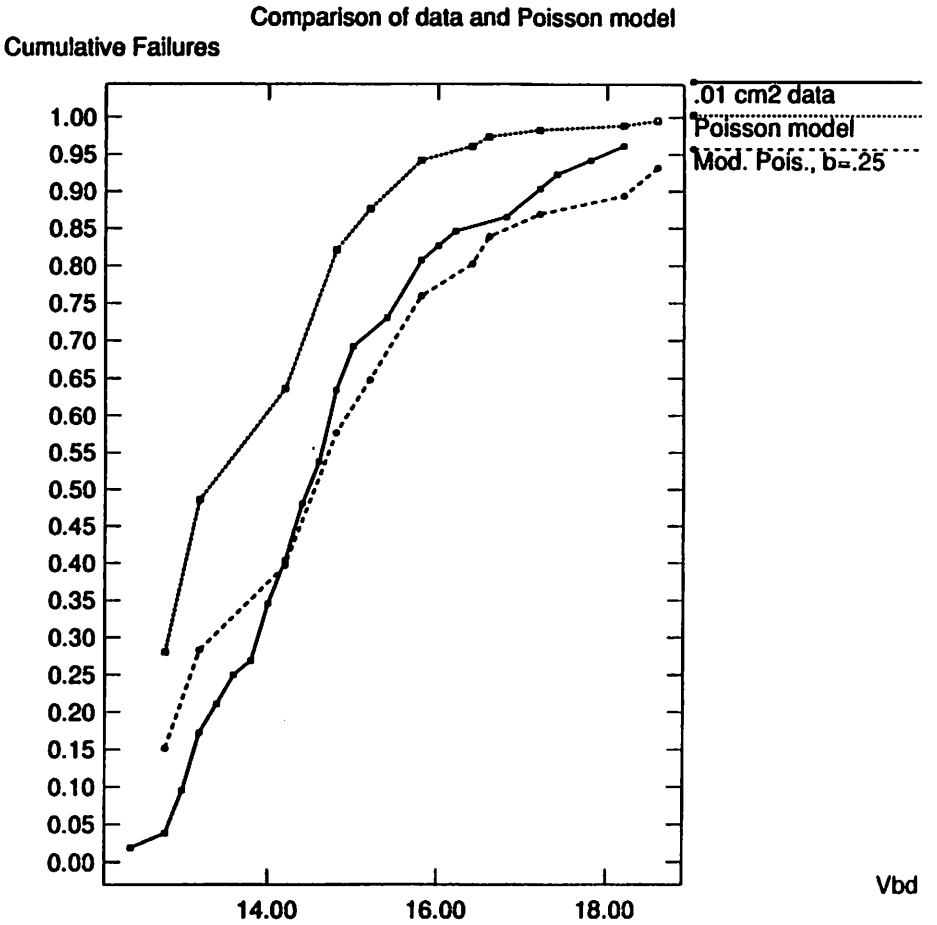
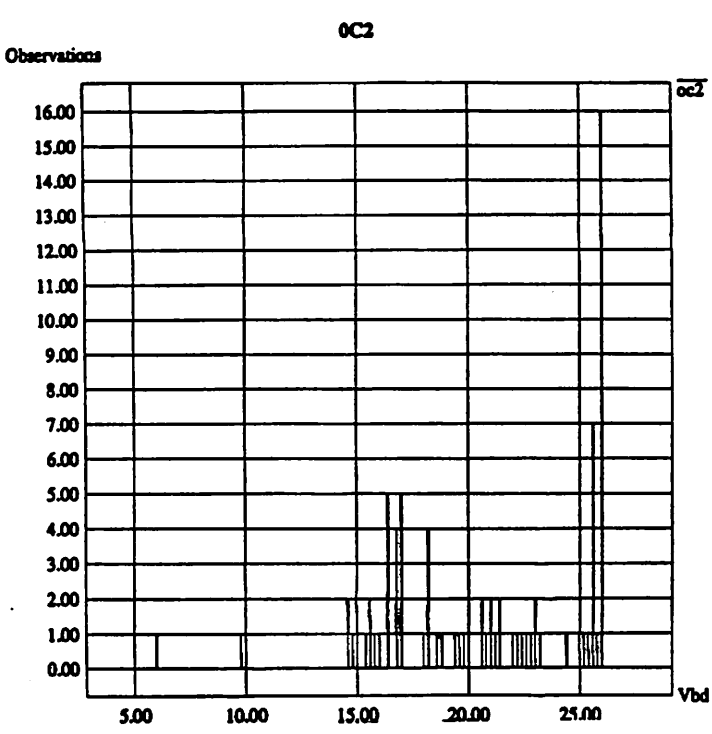
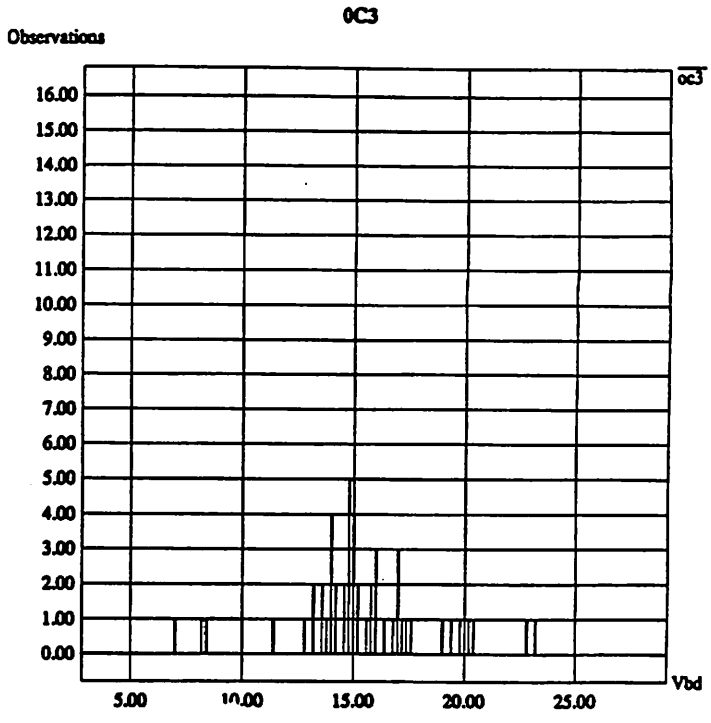
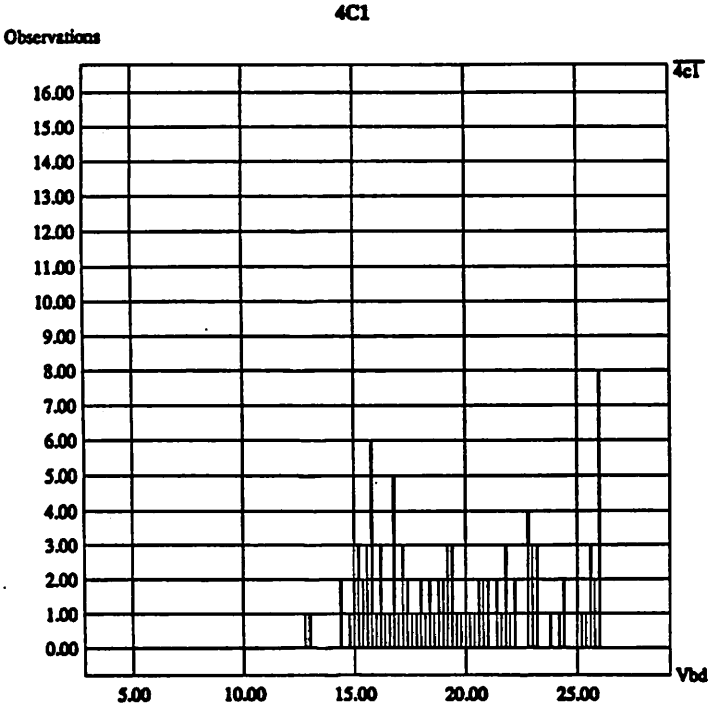


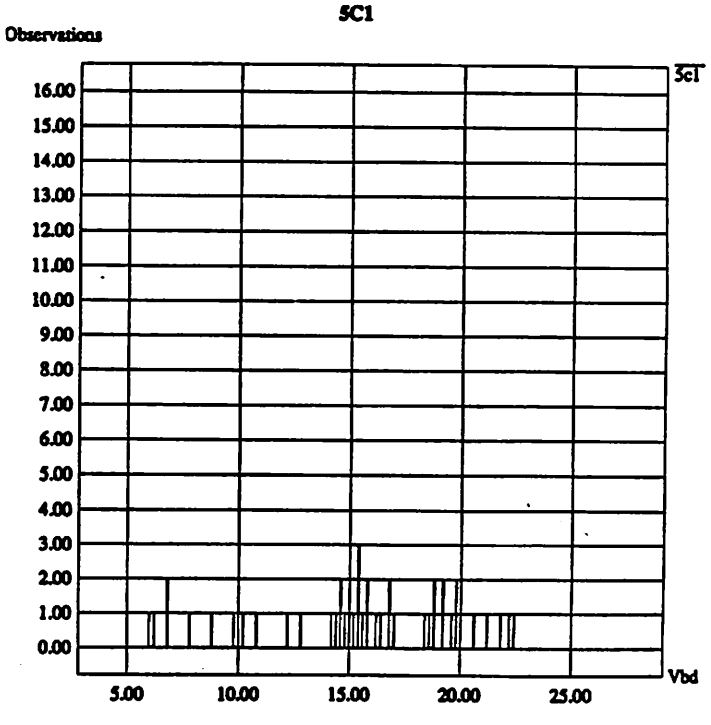
Figure 10: Comparison of .01 cm^2 data and projections made from 0.000625 cm^2 data.

Appendix A

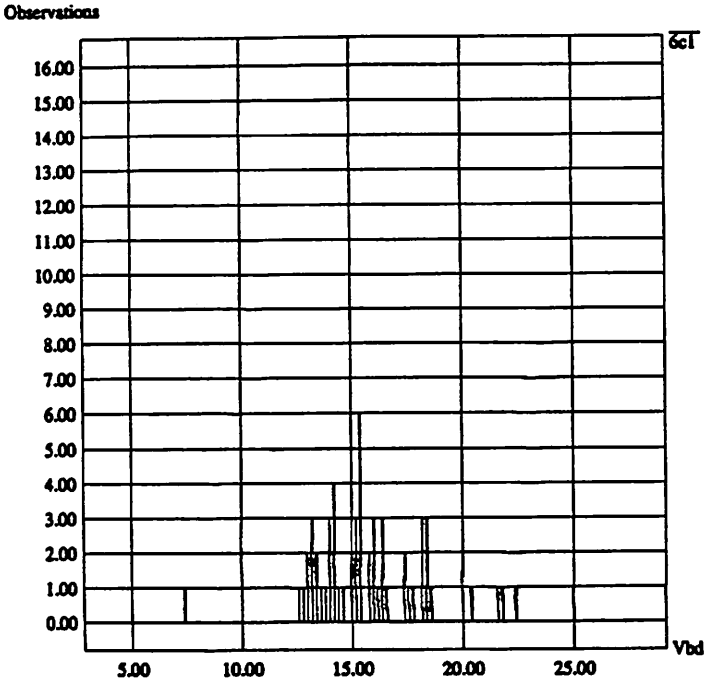

0C2
Observations
oc2
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd


0C3
Observations
oc3
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd

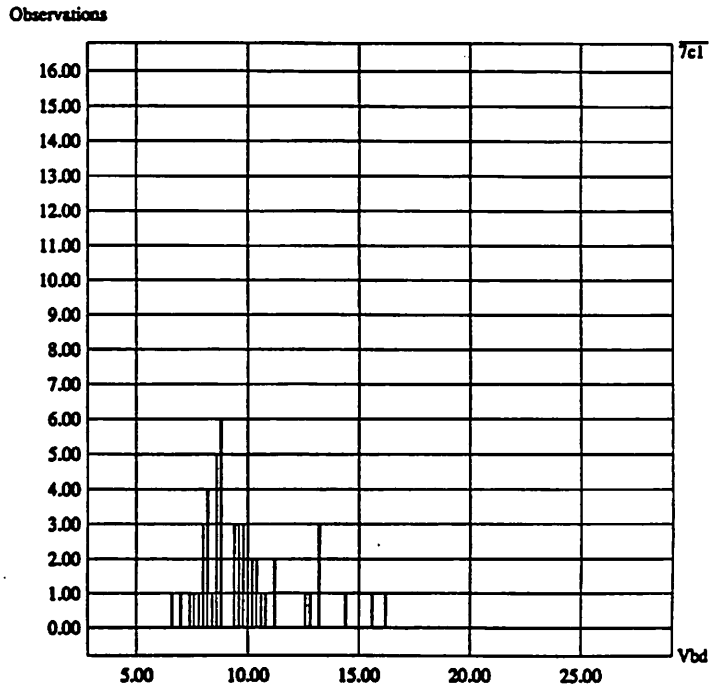

4C1
Observations
4c1
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd


5C1
Observations
5c1
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd

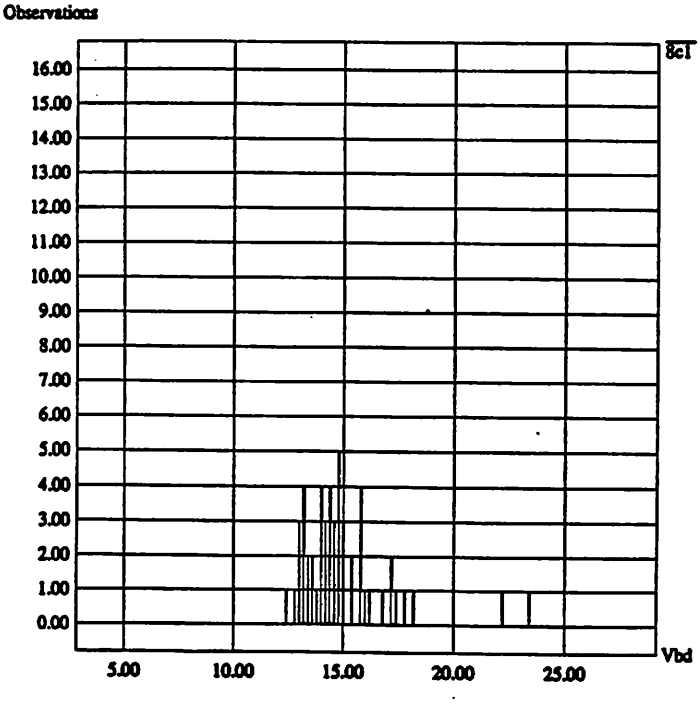
6C1


Observations
6c1
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd

7C1


Observations
7c1
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd

8C1


Observations
8c1
16.00
15.00
14.00
13.00
12.00
11.00
10.00
9.00
8.00
7.00
6.00
5.00
4.00
3.00
2.00
1.00
0.00
5.00
10.00
15.00
20.00
25.00
Vbd

Statistical Experimental Design in Plasma Etch Modeling

Gary S. May

Abstract

The response characterisitics of a CCl_4-based plasma process used to etch doped polysilicon have been examined via a 2^{6-1} fractional factorial experiment folllowed by a Box-Wilson design. The effects of variation in RF power, pressure, electrode spacing, CCl_4 flow, He flow and O_2 flow on several output variables, including etch rate, selectivity, anisotropy, and uniformity were investigated. The screening factorial experiment was designed to isolate the most significant input parameters. From the results of this preliminary investigation, however, it was concluded that each of the six input parameters was significant enough to be modeled. Using this information as a platform from which to proceed, the subsequent phase of the experiment enabled the development of empirical models of etch behavior using response surface methodology.

1. Introduction

Wet etching was the standard method of pattern transfer in early generations of integrated circuits. This stemmed primarily from the fact that etchants with high selectivity to both the substrate and the masking layer were readily available. However, wet etching processes are almost invariably isotropic in nature. Consequently, when the thickness of the film being etched becomes comparable to the minimum patter dimension, the undesirable lateral undercut due to the etch isotropy of wet etchants is no longer tolerable.

In order to overcome the shortcomings of wet etch processes, the technique of ion assisted plasma etching has become widely used in semiconductor manufacturing. Since this method offers the added feature of etch anisotropy, considerable effort has been expended in recent years to develop plasma etch processes. A large portion of this effort has been directed toward thorough characterization of the response of process outputs to variations in input parameters. Such process characterization has necessitated the development of precise models of etch behavior.

Plasma modeling from a fundamental physical standpoint has had limited success. The best physically-based models currently available are capable of describing the chemical kinetics of one-dimensional RF glow discharges [1-4]. These models attempt to derive self-consistent solutions to first-principle equations involving continuity, momentum balance, energy balance and Poisson's relation. This is accomplished by means of computationally expensive numerical simulation methods which typically produce output such as profiles of the distribution of electrons and ions within the plasma sheath. However, although detailed simulation is useful for equipment design and optimization, it is subject to many simplifying assumptions. Due to the extremely complex nature of particle dynamics within a plasma, the connection between these microscopic models and macroscopic parameters such as etch rate has yet to be clearly distinguished.

Since the complexity of practical plasma processes at the equipment level is presently far ahead of theoretical comprehension, other efforts have focused on empirical approaches to plasma modeling involving Response Surface Methods (RSM). These techniques have been used by several authors to obtain statistical models of the etch rates of various thin films. Jenkins et. al. provides a model of the etch rate of p-doped polysilicon in a CF_3Cl/Ar plasma versus pressure, rf power and CF_3Cl fraction [5]. Riley and Hanson, on the other hand, investigated silicon nitride etching in SF_6/He versus the combined SF_6/He flow rate, pressure, power and electrode spacing [6].

However, in these studies, the characterization of many other critical process outputs such as etch uniformity and selectivity has been somewhat overlooked. Therefore, the objective of this work is to obtain a comprehensive set of empirical models for plasma etch rates, anisotropy, nonuniformity and selectivity. These models accurately represent the behavior of a specific piece of equipment under a wide range of etch recipes, thus making them ideal for manufacturing and diagnostic purposes. In particular, this study focuses on the etch characteristics of n^+-doped polysilicon in a CCl_4/He/O_2 plasma. Responses were modeled under the variation of the following six input parameters: RF power, pressure, electrode spacing, and the three gas flows. Etching took place in a Lam Research Autoetch 490 single-wafer plasma system.

2. Experimental Design

A prime example of a fabrication step in which plasma etching has become essential occurs in the definition of polysilicon features for MOS circuits. This process step often requires that a relatively thick polysilicon gate be etched down to a thin slicon dioxide layer. Therefore, high selectivity between poly and SiO_2 is necessary to use a thin gate oxide as an etch stop. In addition, it is desirable that the vertical etch rate of the polysilicon be much greater than its horizontal rate to achieve high etch anisotropy. Finally, good within-wafer uniformity and selectivity to photoresist are also desirable. Carbon tetrachloride has been reported as an anisotropic etchant with a high selectivity for polysilicon in plasma etching [7], thus making it an attractive candidate for this experiment.

The most critical control parameters in plasma etching are RF power, chamber pressure, electrode spacing and gas flow [5-8]. Helium is often added to standard CCl_4 etch recipes in order to enhance etch uniformity. In addition, oxygen is sometimes also introduced into the gas mixture to decrease polymer deposition in the process chamber. The effects of all six process variables must be considered in plasma recipe control. However, RSM techniques are most effective when the number of input factors is limited to six or fewer [5,11]. As a result, it was appropriate to divide the overall experiment into an initial phase to determine the most significant parameters followed by a second phase designed to obtain the statistical response models.

2.1. Screening Experiment

Table I: Range of Input Factors

Parameter	Range	Units
RF Power	300 - 400	watts
Pressure	200 - 300	mtorr
Electrode Spacing	1.2 - 1.8	cm
CCl_4 Flow	100 - 150	sccm
He Flow	50 - 200	sccm
O_2 Flow	10 - 20	sccm

The six factors chosen for the initial screening phase of this experiment along with their respective ranges of variation are shown in Table I. These ranges were chosen to effectively encompass the wide variety of etch recipes currently being utilized in the Berkeley Microfabrication Laboratory. A full factorial experiment to determine all effects and interactions for six factors would require 2^6, or 64 experimental runs. However, in order to reduce the experimental budget, the effects of higher order interactions were neglected and a 2^{6-1} fractional factorial design requiring only 32 runs was performed. The runs were performed in two blocks of 16 trials each in such a way that no main effects or first order interactions were confounded with higher order effects. Three center points were added to the design to provide an estimate of nonlinearity [10]. The randomized design matrix appears in Table II.

Table II: Design Matrix for Screening Experiment

Run	Pressure	RF Power	CCl_4 Flow	He Flow	O_2 Flow	Electrode Gap	Block
1	300	300	100	200	20	1.8	2
2	200	400	100	50	10	1.8	2
3	200	400	150	200	20	1.2	1
4	300	400	150	200	20	1.8	2
5	200	400	150	50	10	1.2	1
6	300	300	150	200	10	1.8	1
7	300	400	100	50	20	1.8	1
8	250	350	125	125	15	1.5	1
9	200	300	150	200	20	1.8	2
10	300	400	150	50	20	1.2	2
11	300	300	100	200	10	1.2	2
12	200	300	150	200	10	1.2	2
13	200	400	100	200	10	1.2	2
14	300	400	150	50	10	1.8	2
15	200	300	100	50	20	1.8	1
16	200	400	100	200	20	1.8	2
17	200	300	100	200	20	1.2	1
18	300	300	150	50	10	1.2	1
19	200	300	100	50	10	1.2	1
20	200	300	150	50	10	1.8	2
21	300	400	150	200	10	1.2	2
22	200	400	100	50	20	1.2	2
23	200	400	150	200	10	1.8	1
24	300	400	100	200	20	1.2	1
25	250	350	125	125	15	1.5	1
26	300	300	100	50	20	1.2	2
27	300	300	100	50	10	1.8	2
28	300	300	150	200	20	1.2	1
29	200	300	150	50	20	1.2	2
30	200	300	100	200	10	1.8	1
31	200	400	150	50	20	1.8	1
32	300	400	100	200	10	1.8	1
33	300	300	150	50	20	1.8	1
34	300	400	100	50	10	1.2	1
35	250	350	125	125	15	1.5	2

2.2. RSM Modeling Experiment

Analysis of the first stage of the experiment revealed significant nonlinearity in all responses, which indicated the necessity of quadratic models. Also, none of the input factors were found to have a statistically insignificant effect on all of the responses of interest. Thus, none were omitted from the response surface models derived in the subsequent phase. In order to obtain these models, it was necessary to augment the data gathered with a second experiment which employed a Central Composite Circumscribed (CCC) Box-Wilson design. In this design, the 2-level factorial "box" was enhanced by further replicated experiments at the center (to provide a measure of error) as well as symmetrically located "star" points [10].

Table III: Additional "Star Point" Recipes for Box-Wilson Experiment

Run	Pressure	RF Power	CCl_4 Flow	He Flow	O_2 Flow	Electrode Gap
36	250	350	125	125	3	1.5
37	250	231	125	125	15	1.5
38	250	350	125	200	15	1.5
39	250	350	125	125	15	0.8
40	369	350	125	125	15	1.5
41	250	350	125	0	15	1.5
42	250	350	125	125	15	1.5
43	250	350	66	125	15	1.5
44	250	350	184	125	15	1.5
45	250	350	125	125	15	1.5
46	250	350	125	125	15	1.5
47	250	350	125	125	15	2.2
48	250	350	125	125	15	1.5
49	250	469	125	125	15	1.5
50	131	350	125	125	15	1.5
51	250	350	125	125	27	1.5
52	250	350	125	125	15	1.5
53	250	350	125	125	15	1.5

A complete CCC design for six factors requires a total of 91 runs. Therefore, in order to reduce the size of the experiment and make use of the results from the screening phase, a half replicate design was again employed. The entire second phase required a total of 18 additional runs. The 18 added recipes are shown in Table III. The circumscribed design was selected as opposed to a inscribed (CCI) design to allow the models to accurately predict the responses over the entire range of the input factor settings [11]. However, in the case of He flow for runs 39 and 41, the necessary star point required recipe settings of 303 and -53 sccm, which are beyond the operational capabilities of the equipment. In this case, the recipe was modified to reflect the maximum/minimum possible parameter settings of the Lam etcher (200 and 0 sccm, respectively). A graphic description of central composite designs appears in Figure 1.

3. Experimental Apparatus and Technique

Etching was performed on a simple test structure designed to measure the vertical etch rates of polysilicon, SiO_2, and photoresist as well as the lateral etch rate of poly. The samples consisted of 4-in diameter silicon wafers with films of thermal SiO_2, phosphorous-doped polysilicon and Kodak 820 photoresist. Approximately 1.2μm of poly was deposited over 5000A of thermal SiO_2 by low-pressure chemical vapor deposition (LPCVD). The poly resistivity was measured at 86.0Ω-cm. Oxide was grown in a steam ambient at 1000 °C. One micron of photoresist was spun on and baked for 60 seconds at 120 °C. Due to the insufficient selectivity of the polysilicon etch rate with respect to that of the photoresist, poly lines for SEM photos were patterned with a mask consisting of low-temperature oxide (LTO) deposited at 450 °C by LPCVD. A cross section showing the critical measurement area is shown in Figure 2.

The etching apparatus consisted of a Lam Research Corporation Autoetch 490 single-wafer parallel-plate system. The etching samples rest on the grounded lower electrode while the upper electrode is

excited by a 13.56 MHz RF generator operating through a matching network. The anodized alumninum electrodes are circular and equal in area. The electrode walls are also composed of aluminum. Process gases are introduced into the chamber through nearly 1000 holes in the upper electrode in "showerhead" fashion. Reactor pressure is monitored with a capacitance manometer and controlled automatically with a throttle valve [12,13]. The etcher was monitored via a real-time statistical process control scheme to ensure consistency in equipment operation throughout the experiment. A schematic diagram of the etching system appears in Figure 3.

Film thickness measurements were performed on five points per wafer (as in Figure 4) both before and after etching using a Nanometrics Nanospec AFT system in conjunction with an Alphastep 200 Automatic Step Profiler. Etch rates were calculated by dividing the difference between the pre- and post-etch thickness by the etch time. The lateral etch rate for poly was measured via SEM. Expressions for the selectivity of the poly with respect to oxide (S_{ox}) and with respect to resist (S_{ph}) along with percent anisotropy (A) and percent nonuniformity (U), respectively, are given below:

$$S_{ox} = \frac{R_p}{R_{ox}} \tag{1}$$

$$S_{ph} = \frac{R_p}{R_{ph}} \tag{2}$$

$$A = \left[1 - \frac{L_p}{R_p}\right] \tag{3}$$

$$U = \frac{|R_{pc} - R_{pe}|}{R_{pc}} * 100 \tag{4}$$

where R_p is the mean vertical poly etch rate over the five points, R_{ox} is the mean oxide etch rate, R_{ph} is the mean resist eatch rate, L_p is the lateral poly etch rate, R_{pc} is the poly etch rate at the center of the wafer, and R_{pe} is the mean poly etch rate of the four points located about one inch from the edge [14].

4. Results and Discussion

After the initial screening experiment, a few of the input factors were found to have an insignificant effects upon individual responses. For example, the electrode gap spacing had little effect on the etch selectivity with respect to oxide. However, no single factor was statistically irrelevent to all five responses of interest. Although it did not appear to affect oxide selectivity, gap spacing did indeed have a dramatic impact upon etch uniformity. Table IV provides an overview of the significance of each main effect resulting from the fractional factorial data. (Since they are extremely time-consuming, the complete set of SEM photos for the anisotropy measurements have been delayed in order to complete the other models in a timely manner. These photos will be taken and the anisotropy data will be analyzed at a later date. Afterwards, an anisotropy model will be similarly derived and appened to this set).

Table IV: Results of Screening Experiment

Factor	Statistical Significance			
	R_p	S_{ox}	S_{ph}	U
Pressure	0.0090	0.0001	0.0001	0.0677
RF Power	0.0001	0.0046	0.0001	0.0493
CCl_4	0.0032	0.0410	0.0001	0.0672
He	0.0001	0.0001	0.0001	0.0002
O_2	0.0043	0.0669	0.0014	0.9581
Gap	0.0185	0.4134	0.0001	0.0107

** Only factors with a significance < 0.05 are considered significant.*

The above results indicate that all six controlled parameters have a significant effect both on etch rate and resist selectivity. On the other hand, oxide selectivity is only impacted by pressure, power, CCl_4 and helium flow. Etch uniformity depends primarily on power, helium flow and gap spacing. The additional 18 runs in the next phase of the experiemnt yielded quadratic models which indicate the precise interaction between input factors and the four responses. These models are discussed below.

4.1. Polysilicon Etch Rate

Fitting a regression model for R_p yielded the following expression:

$$R_p = 3540 - 10.1P + 11.0Rf - 17.8CCl_4 + 11.2He - 1030G - 61.4O_2 \quad (5)$$
$$- 0.034P*He + 7.82P*G + 0.389P*O_2 + 0.085Rf*CCl_4 - 8.36Rf*G - 0.132(CCl_4)^2$$
$$- 0.059CCl_4*He + 12.4CCl_4*G - 0.059He^2$$

where R_p is in A/min and the units of every other parameter are given in Table I. This equation was derived by stepwise regression [11], and it has a standard deviation of +/- 98 A/min. The Analysis of Variance (ANOVA) table for the etch rate model is shown in Table V.

Table V: ANOVA for Poly Etch Rate Model

Source	DF	Sum of Squares	Mean Square	F-Ratio	Significance
Total	52	24717141	475329.63		
Regression	15	21562592	1437506	16.86	0.000
Residual	37	3154549	85258.07		
Lack of Fit	29	2823740	97370.33	2.36	0.103
Error	8	330809	41351.11		

Adjusted $R^2 = 0.821$

The F-test for all the coefficients of the model being equal to zero indicated that this is highly unlikely, since the probability that F(15,37) > 16.86 is negligible. In addition, the F-test for lack of fit reveals little evidence of lack of fit since F(29,8) as large as 2.36 occurs 10.3% of the time. Therefore, most of the error of the model is due to experimental error. The "adjusted R^2 is a parameter between zero and one (with one being optimal) which also measures the goodness of fit.

The etch rate model is fairly complex, but a few interesting relationships are indicated in the contour plots of Figures 5 and 6. In Figure 5, R_p surfaces are plotted against RF power and chamber pressure with all other parameters set at their nominal values. For high process throughput, etch rate should preferably be as high as possible. This occurs at high power and high pressure. In Figure 6, the effects of CCl_4 flow and electrode spacing are explored. Here, it is seen that the highest etch rates occur when the gap is narrow and the flow rate is moderate.

4.2. Etch Uniformity

The uniformity regression model and corresponding ANOVA table are:

$$U = -11.0 - 0.168P + 0.094Rf + 0.714CCl_4 - 0.415He + 11.9G \quad (6)$$
$$- 0.071O_2 + 0.009P^*O_2 - 0.002Rf^*CCl_4 + 0.001Rf^*He$$
$$- 0.001CCl_4^*He + (8e{-}4)He^2 - 1.39G^*O_2 \quad +/- 2.15(\%)$$

Table VI: ANOVA for Etch Uniformity Model

Source	DF	Sum of Squares	Mean Square	F-Ratio	Significance
Total	52	5896.02	113.39		
Regression	12	4255.83	354.65	8.65	0.000
Residual	40	1640.19	41.01		
Lack of Fit	32	1295.73	40.49	0.94	0.588
Error	8	344.46	43.06		

Adjusted $R^2 = 0.638$

Tests for significance reveal that model coefficients are relevant. In addition, the F-test for fit shows no lack of fit. The contours in Figures 7 and 8 describe some results of the uniformity model. In Figure 7, U is plotted against pressure and power. Optimum uniformity is observed at high pressure and low power. Thus, good uniformity is achieved at the expense of high etch rates. The effects of He flow and electrode spacing are observed Figure 8. This plot verifies the initial assumption that helium enhances uniformity, but only up to an optimum value of flow rate beyond which U begins to degrade.

4.3. Oxide Selectivity

The regression model and AVOVA table for S_{ox} are given below:

$$S_{ox} = -9.87 + 0.097P + 0.03Rf - 0.06CCl_4 + 0.03He + 0.079O_2 - (2e{-}4)P^*Rf \quad (7)$$
$$+ (2.9e{-}4)P^*CCl_4 - (3e{-}4)P^*He + (7.4e{-}5)Rf^*He \quad +/- 0.31$$

Table VII: ANOVA for Oxide Selectivity Model

Source	DF	Sum of Squares	Mean Square	F-Ratio	Significance
Total	52	248.70	4.78		
Regression	9	213.26	23.70	28.76	0.000
Residual	43	35.43	0.82		
Lack of Fit	35	31.35	0.90	1.75	0.205
Error	8	4.09	0.51		

Adjusted $R^2 = 0.828$

The F-test for the model possessing coefficients equal to zero indicated that this is highly unlikely, and the F-test for fit showed no evidence that a more complex model is required. A few implications of the oxide selectivity model appear in Figures 9 and 10. Figure 9 shows S_{ox} contours versus RF power and pressure. According to this plot, highest oxide selectivity occurs at high pressure and low power. Thus, a trade-off exists between high etch rate and good selectivity in terms of power. The effects of CCl_4 flow and pressure can be visualized in Figure 10. Greatest oxide selectivity occurs when pressure and CCL_4 flow are both high.

4.4. Photoresist Selectivity

The regression model and ANOVA Table for S_{ph} are:

$$S_{ph} = 7.56 + 0.009P + 0.014Rf - 0.022CCl_4 + 0.006He - 2.59G - 0.099O_2 \quad (8)$$
$$- (5e{-}5)P^*Rf + (1.3e{-}4)P^*CCl_4 - (7e{-}5)P^*He + (3.7e{-}4)P^*O_2 + (2.7e{-}5)Rf^2$$
$$+ (3.6e{-}5)Rf^*He - (5e{-}5)CCl_4^*He + 0.757G^2 \quad +/- 0.09$$

Table VIII: ANOVA for Photoresist Selectivity Model

Source	DF	Sum of Squares	Mean Square	F-Ratio	Significance
Total	52	15.24	0.29		
Regression	14	12.61	0.90	13.02	0.000
Residual	38	2.63	0.07		
Lack of Fit	30	2.42	0.08	3.07	0.050
Error	8	0.21	0.03		

Adjusted $R^2 = 0.764$

Statistical tests for model complexity and fit indicate no reason to doubt the adequacy of the resist selectivity model. The model is visualized in Figures 11 and 12. Figure 11 shows S_{ph} contours versus power and pressure, and Figure 12 shows the effects of CCl_4 flow and pressure. These plots indicate that photoresist selectivity possesses similar trends to that of oxide. This result is not surprising, since both oxide and resist are etched mechanically rather than chemically within the plasma.

5. Conclusion

An economical two-phase experiment has been designed and conducted to characterize the etch rate, uniformity, and selectivity to SiO_2 and photoresist of n^+-doped polysilicon versus a comprehensive set of controlling parameters. These parameters were fit to quadratic response surface models. The models can be used for a variety of manufacturing purposes, including recipe generation, process control, and diagnosis.

6. Future Work

SEM photos for anisotropy measurements are still pending. Therefore, the complete set of models is presently unavailable. However, this data will be compiled, analyzed and added to this study in the near future.

7. Acknowledgement

The author wishes to thank Jiahua Huang for her assistance in fabricating the test structure for this experiment and Hai-Fang Guo for her aid in equipment monitoring during etching. This research was supported by the Semiconductor Research Corporation, the National Science Foundation and the California MICRO program.

References

[1] T. J. Cotler, M. S. Barnes, and M. E. Elta, "A Monte Carlo Microtopography Model for Investigating Plasma/Reactive Ion Ecth Profile Evolution," *J. Vac. Sci. Tech. B*, vol. 6, no. 2, Mar/Apr, 1988.

[2] M. S. Barnes, T. J. Cotler, and M. E. Elta, "Large-Signal Time-Domain Modeling of Low-Pressure RF Glow Discharges," *J. Appl. Phys.*, vol. 61, no. 1, January, 1987.

[3] D. B. Graves, "Modeling Plasma-Enhanced CVD Reactors for Semiconductor Fabrication," *Short Course on Chemical Vapor Deposition*, University Extension, UC-Berkeley, August 7-9, 1989.

[4] A. P. Paranjpe, J. P. McVittie, and S. A. Self, "Numerical Simulation of 13.56 MHz Symmetric Parallel Plate RF Glow Discharges in Argon," *Proc. 41st Gas. Elec. Conf.*, October, 1988.

[5] M. W. Jenkins, M. T. Mocella, K. D. Allen, and H. H. Sawin, "Modeling Plasma Etching Processes Using Response Surface Methodology," *Solid State Tech.*, April, 1986.

[6] P. E. Riley and D. A. Hanson, "Study of Etch Rate Characteristics of SF_6/He Plasmas by Response-Surface Methodology: Effects of Interelectrode Spacing," *IEEE Trans. Semicon. Manufac.*, vol. 2, no. 4, November, 1989.

[7] D. H. Bower, "Planar Plasma Etching of Polysilicon Using CCl_4 and NF_3," *J. Electrochem. Soc.*, vol. 129, no. 4, April, 1982.

[8] C. B. Zarowin and R. S. Horwath, "Control of Plasma Etch Profiles with Plasma Sheath Electric Field and RF Power Density," *J. Electrochem. Soc.*, vol. 129, no. 11, November, 1982.

[9] S. E. Bernacki and B. B. Kosicki, "Controlled Film Formation During CCl_4 Plasma Etching," *J. Electrochem. Soc.*, vol. 131, no. 8, August, 1984.

[10] G. E. P. Box, W. B. Hunter, and J. S. Hunter, *Statistics for Experimenters*, New York: Wiley, 1978.

[11] *RS/Discover User's Guide*, BBN Software Products Corporation, June 1988.

[12] *Autoetch Plasma Etch System Operation and Maintenance Manual*, Lam Research Corporation, March, 1985.

[13] "Advanced Dry Etching of Aluminum and its Alloys," *Solid State Tech.*, April, 1986.

[14] S. Wolf and R. N. Tauber, *Silicon Processing for the VLSI Era*, Sunset Beach: Lattice Press, 1987.

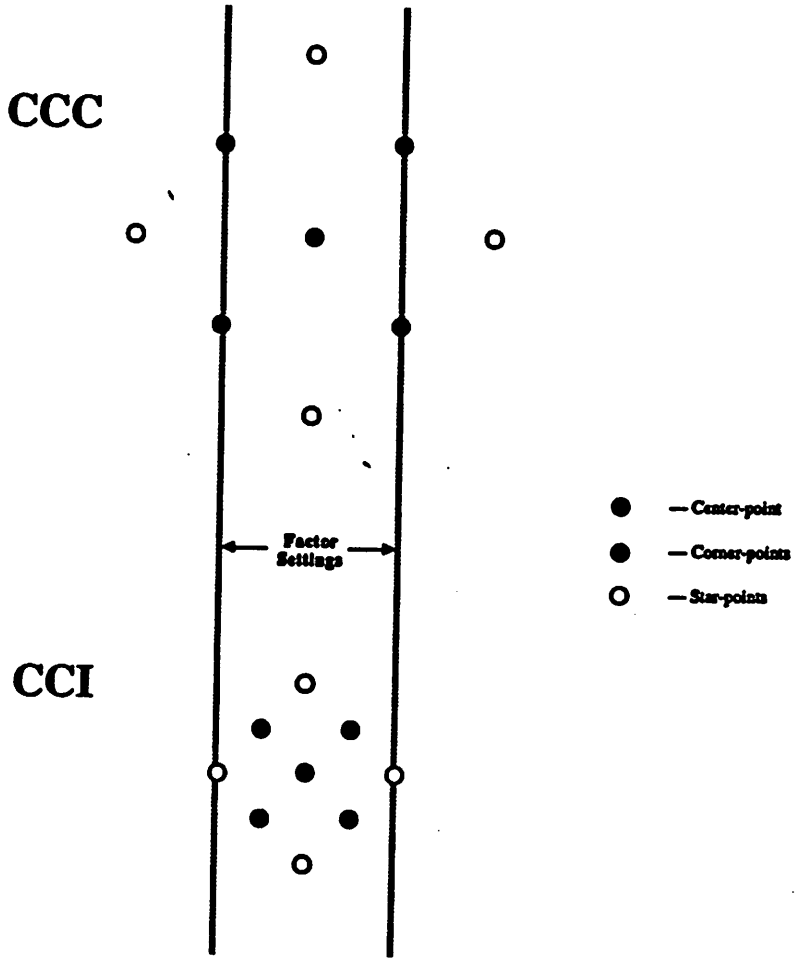


Figure 1: Central Composite Box-Wilson experimental designs [11].

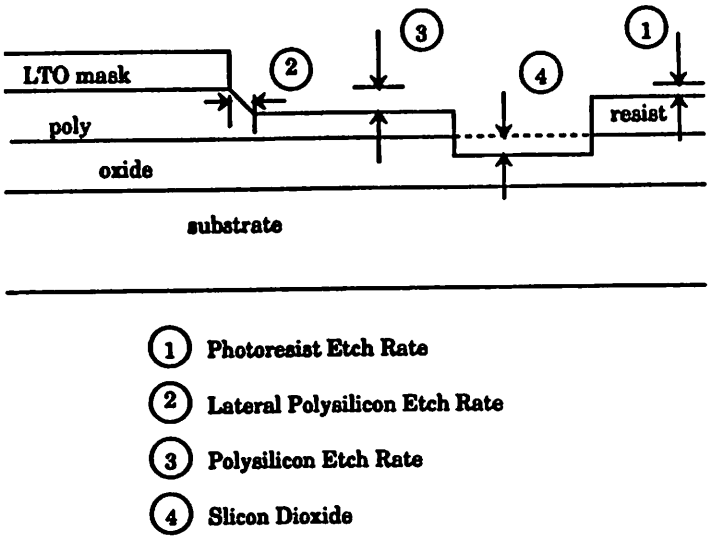


Figure 2: Cross section of test structure describing the measurements of interest.

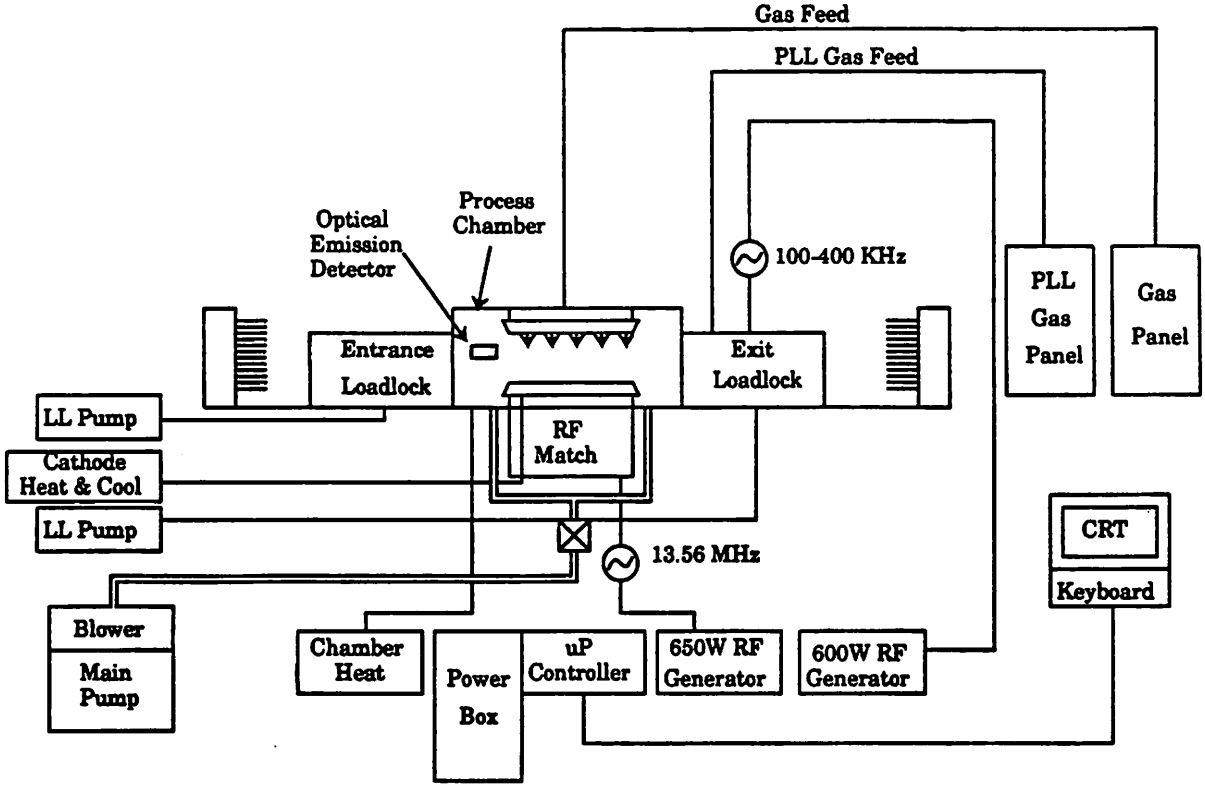


Figure 3: Schematic diagram of Lam Autoetch 490 plasma etching system [13].

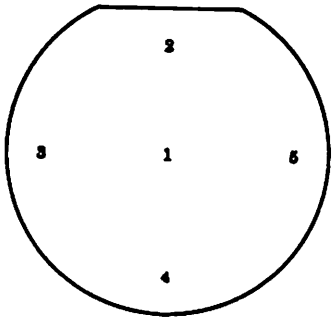


Figure 4: Wafer Measuremnt Sites.

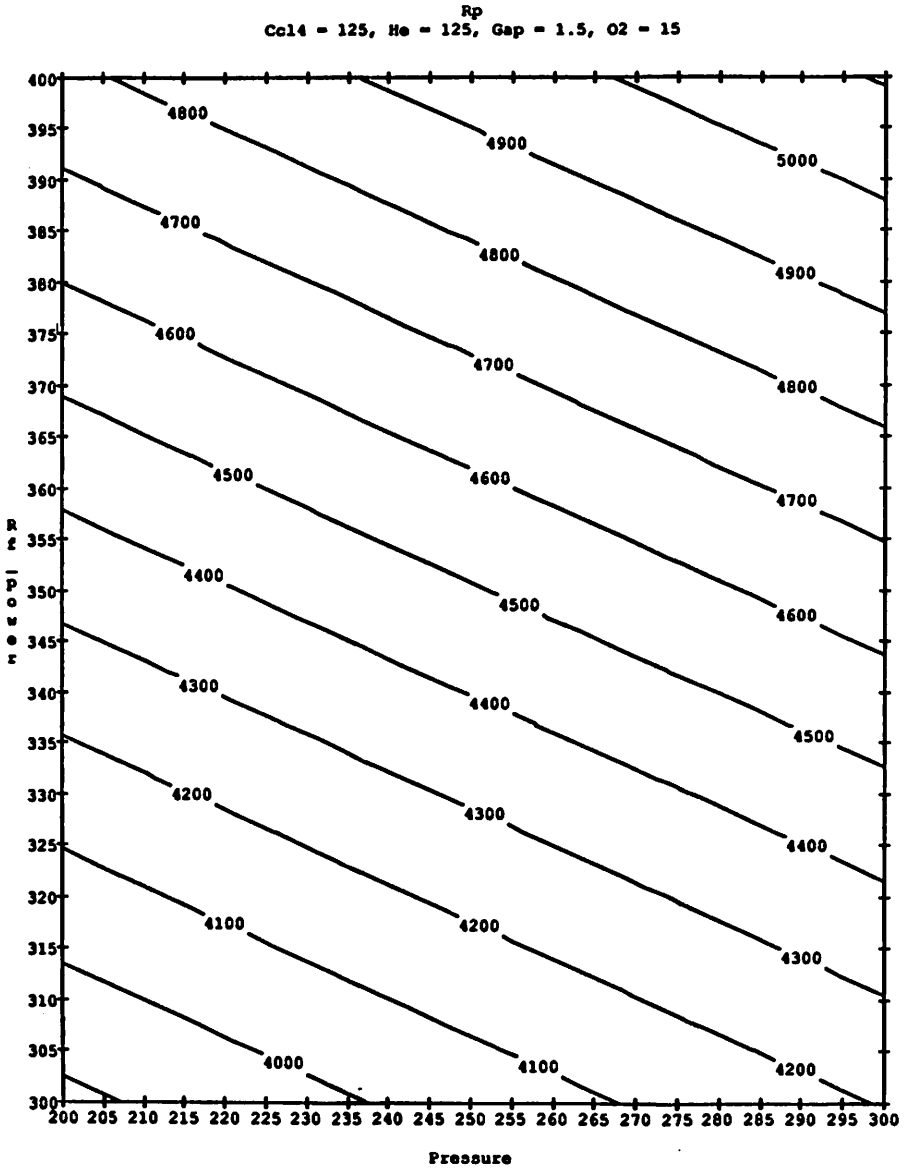


Figure 5: Contour plot of polysilicon etch rate versus RF power and pressure.

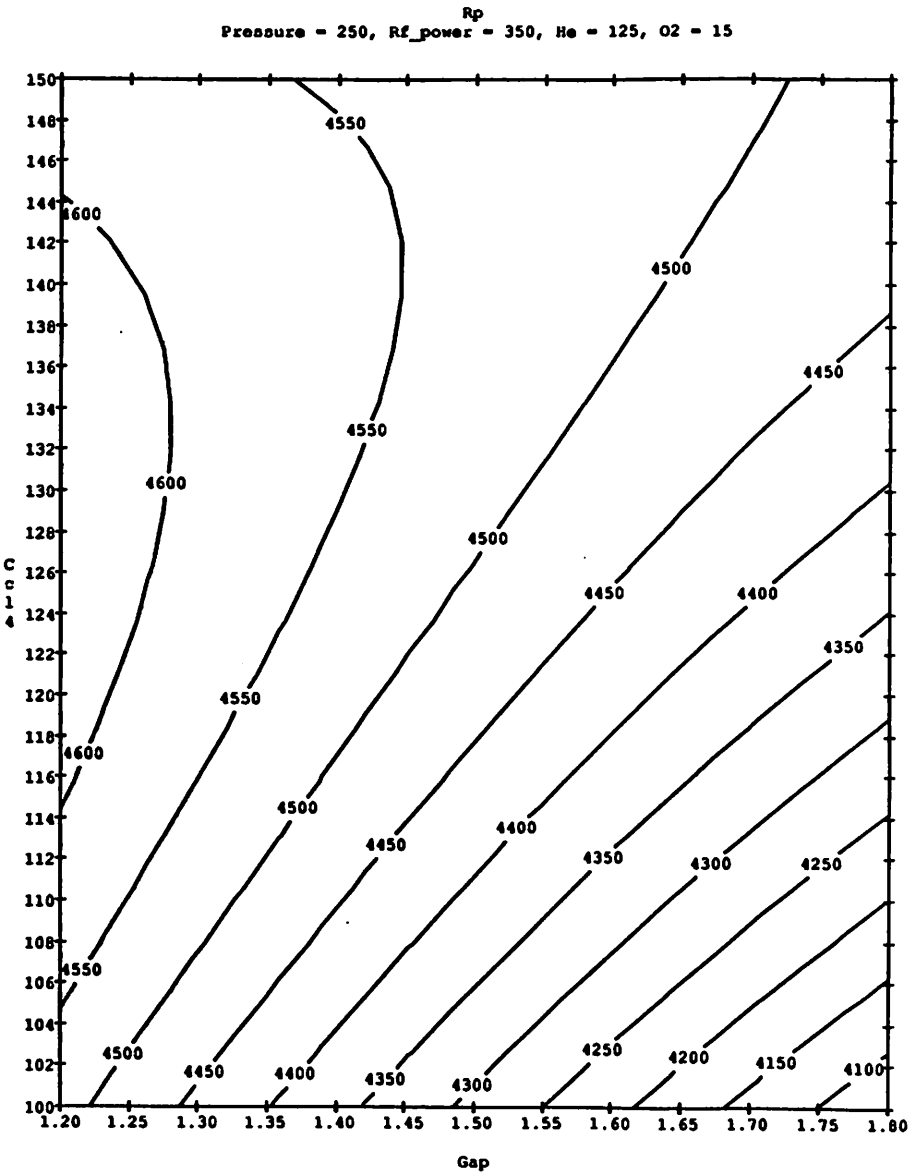


Figure 6: Contour plot of polysilicon etch rate versus CCl_4 flow and electrode spacing.

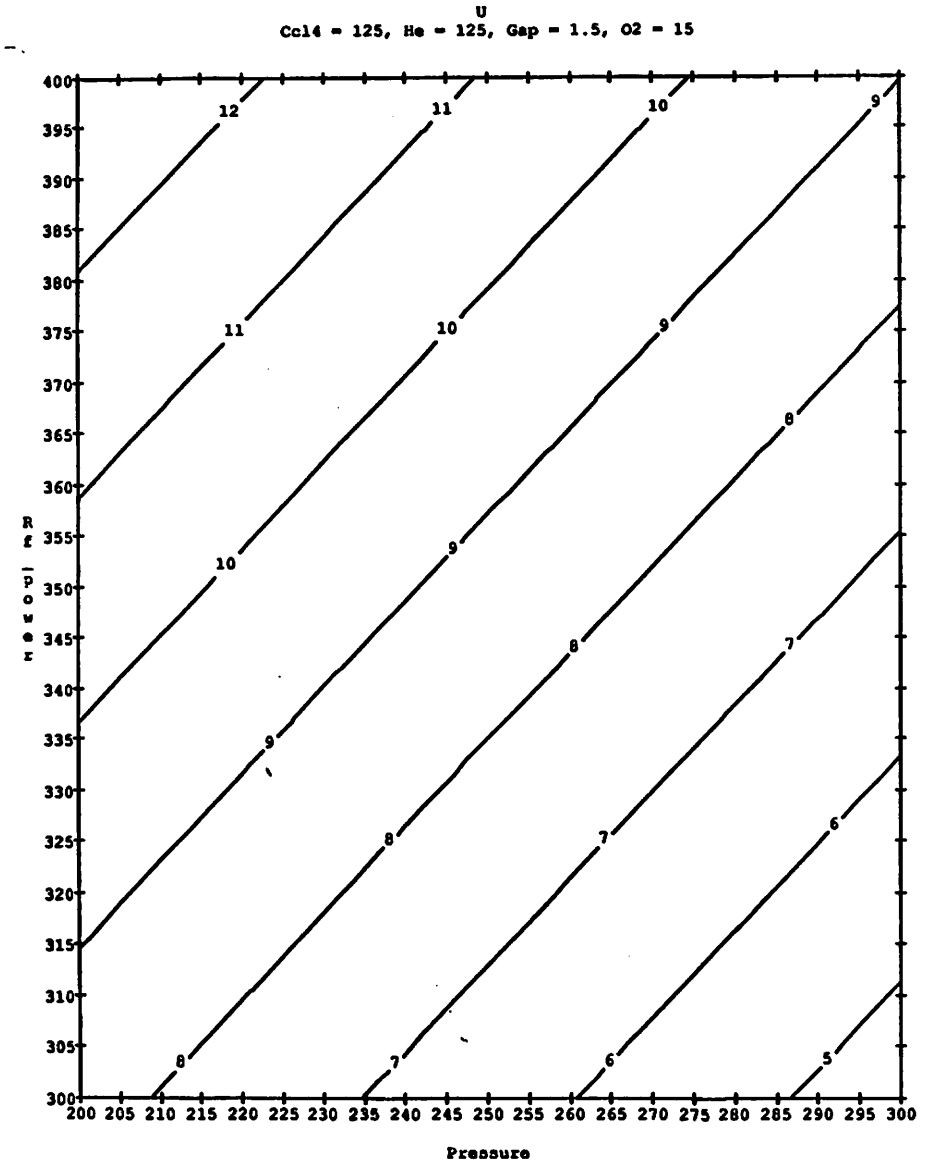


Figure 7: Contour plot of etch uniformity versus RF power and pressure.

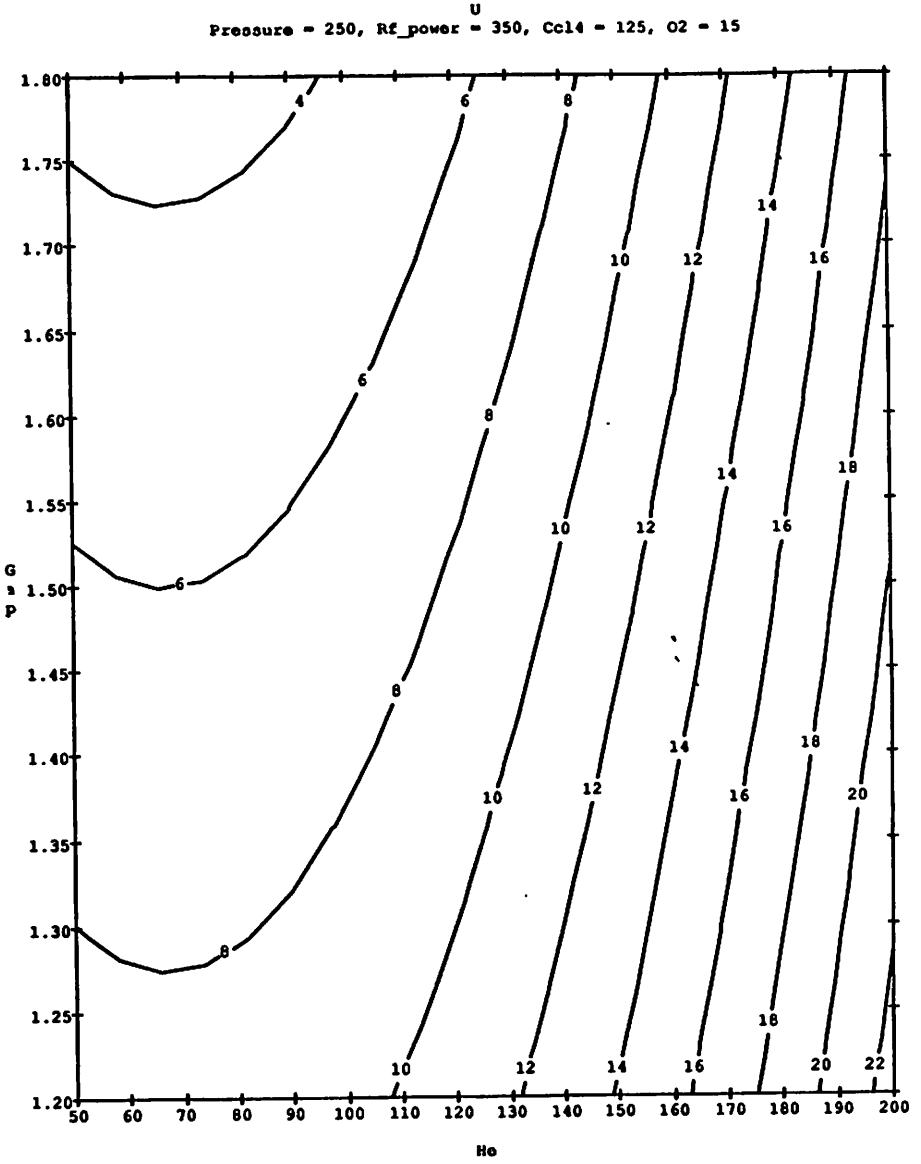


Figure 8: Contour plot of etch uniformity versus electrode spacing and He flow.

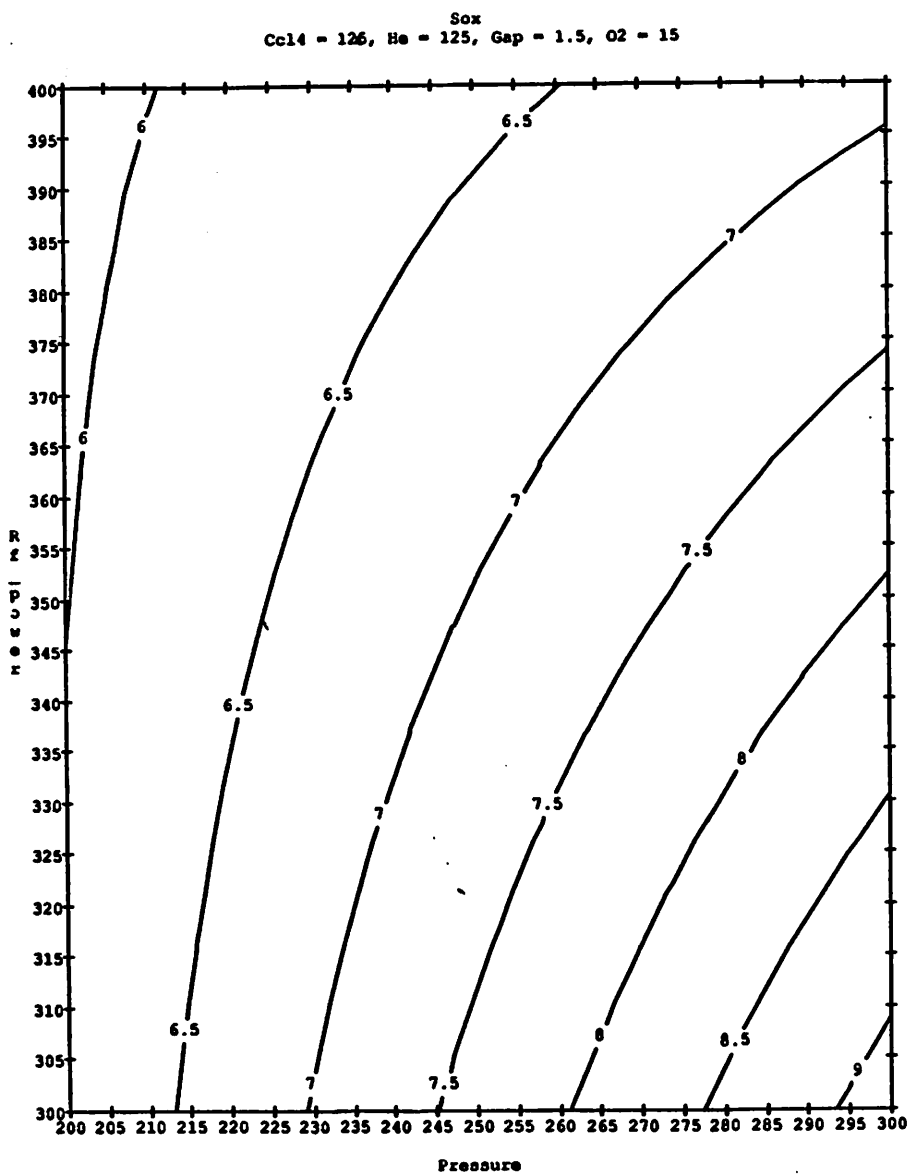


Figure 9: Contour plot of oxide selectivity versus RF power and pressure.

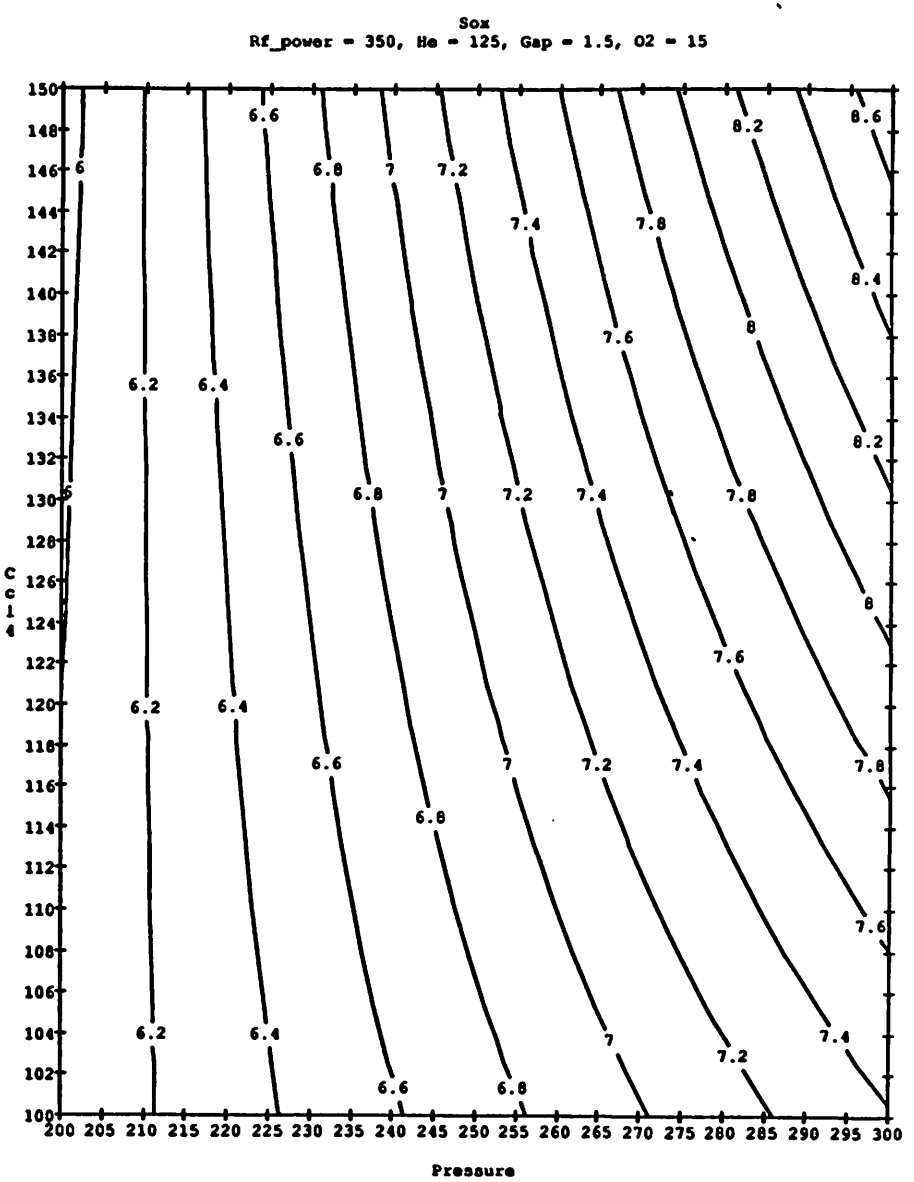


Figure 10: Contour plot of oxide selectivity versus CCl_4 flow and pressure.

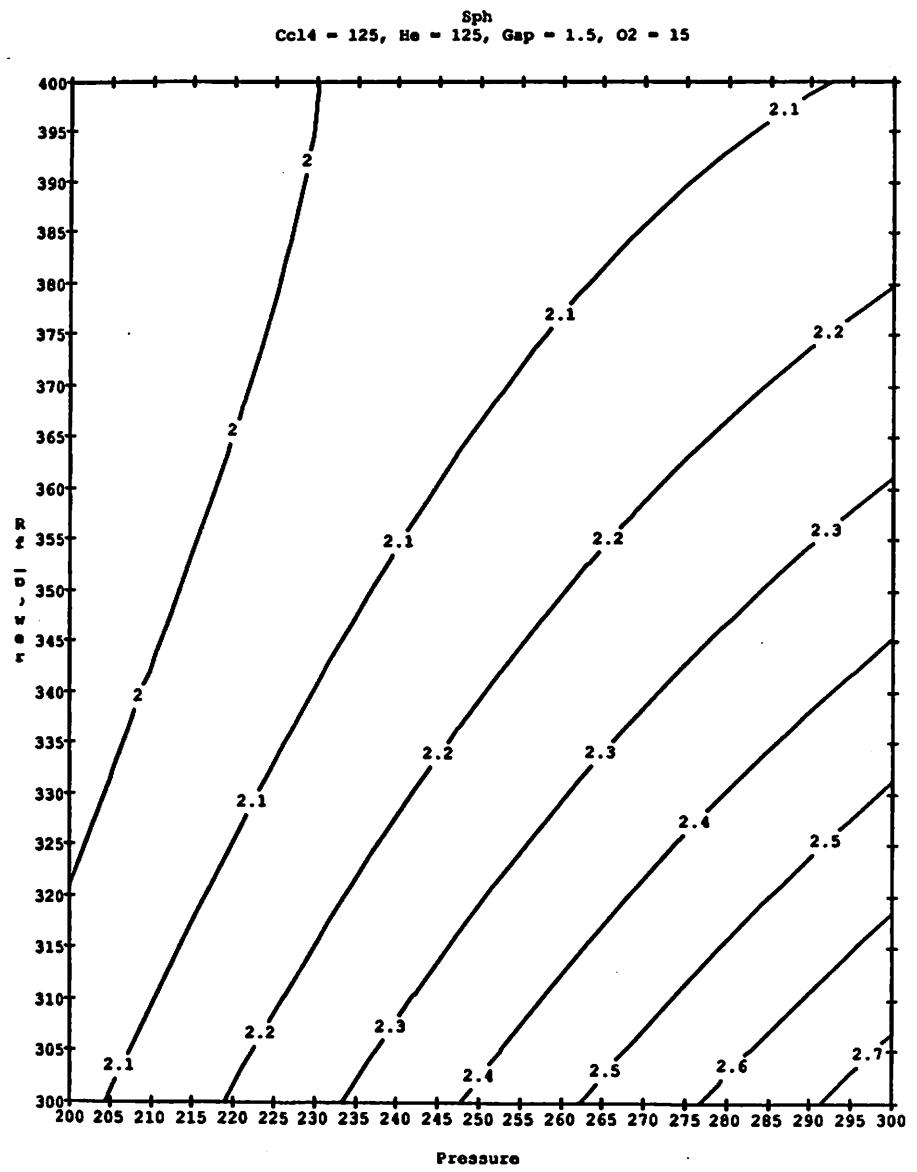


Figure 11: Contour plot of resist selectivity versus RF power and pressure.

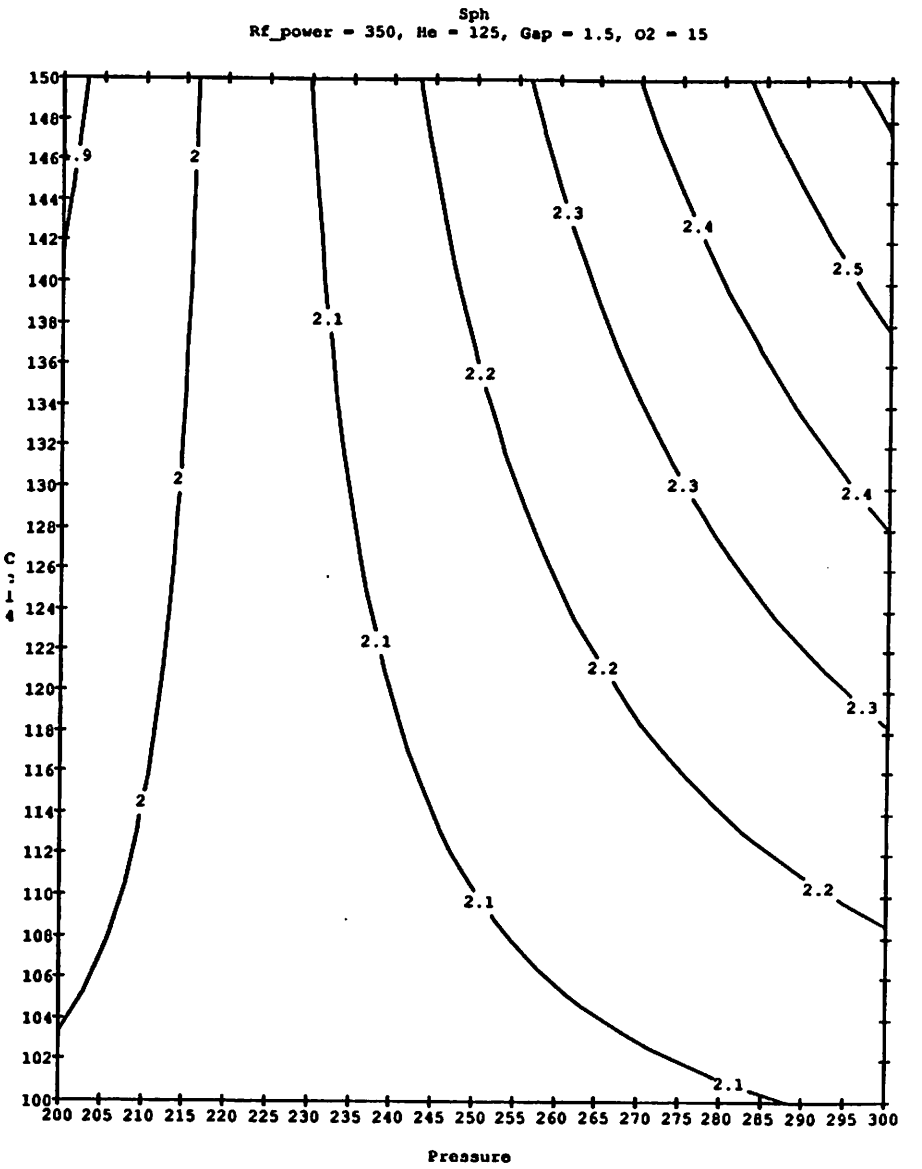


Figure 12: Contour plot of resist selectivity versus CCl_4 and pressure.

Parametric Yield Analysis of CMOS EPROMs

Eric D. Boskin

Abstract

This report describes the statistical comparison of two EPROM designs. This comparison is based on the geometrical representation of their respective yield bodies. Spice3 was used to evaluate the sensitivities of the two circuits to some of the most prominent process variations.

1. Introduction

The performance yield loss of analog integrated circuits due to variation of the fabrication parameters is of significant concern in the semiconductor industry. Statistically based methods have been developed to understand and quantify the variation of these parameters, and their effect on circuit performance. This project applies many of these techniques in the study of the design and performance yield of a CMOS EPROM. Specifically, the design trade-off between a static pullup and the use of precharging on the bit lines will be investigated.

2. Methodology

The Spice circuit simulator will be used to study the effect of parameter variation on performance. The MOS transistor model in Spice contains certain parameters which are directly related to physical device parameters. These model parameters can be varied, in repeated runs of the simulator, with the distribution seen in the manufacturing environment (Monte Carlo analysis). Each simulation run represents one manufactured die. The range of performance seen in the simulation results will be very close to the performance spread of manufactured parts, so the simulation results can be used to analyze and predict performance yield.

Statistical methods will also be used to determine the sensitivity of the circuits to fabrication parameter variation. The most significant parameters will be used to create a model for the yield body of the circuit through linear regression.

2.1. Statistical Model for Fabrication Variation

Three fabrication parameters will be varied. These are the change in channel length (Ld), the oxide thickness (Tox) and the substrate doping level, (Sub). The three parameters were varied for both the n-channel and the p-channel transistors (i.e. Ntox and Ptox). One value for each of these six parameters will be generated for each run. This variation will simulate global variation between die, wafers and lots. Further, a local variation (intradie) will be introduced by a small variation in these parameters between the matched transistors in the sense amplifier circuitry. One local value for each parameter was also generated for each simulation run, for a total of twelve varying parameters.

Gaussian distributions will be used for the probability density functions of the fab parameters. The statistical model will take into account parameters with global variation and parameters with local variation. Further, the correlation between parameters will also be accounted for. A random number generator with a gaussian distribution is the basis for creating values for each of the parameters for each simulation. The random number generator takes the mean and standard deviation of a distribution as input parameters, and generates a random number from that gaussian distribution.

Global variation is represented by one call to the random number generator with the global standard deviation of the parameter, to establish a value for that die. Then, local variation is modeled with a second call to the random number generator using the first value as the mean, and a smaller, local standard deviation.

The local standard deviation was estimated to be twenty-five percent of the global standard deviation. This method was used to generate global and local values for the variables Nsub and Psub, which are independent from any other parameter. It was also used to generate the four values related to oxide thickness on one die, as the oxide thickness for n and p-channel transistors on any die show only small, local variation.

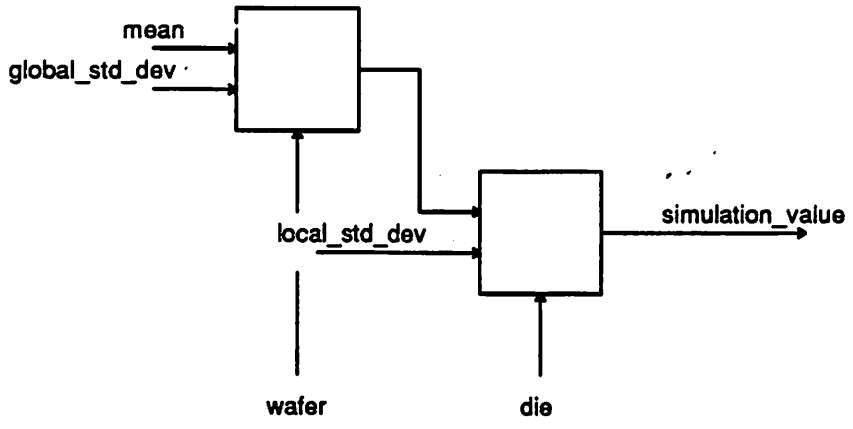


Figure 1 - Model of Statistical Variation

Parameters which exhibit correlation, such as the line width variation between n-channel and p-channel transistors, require a more sophisticated model. In this case the mean, standard deviation and correlation for the parameters is known. The method for generating these parameters is based on the multivariate statistical technique of principal component analysis. More specifically, in order to generate two correlated variables, X and Y, with given means (μ) and global standard deviations (σ), we used the formulas:

$$X = \sigma_X \sqrt{\frac{1}{a^2 + b^2}} (aV + bW) + \mu_X$$

$$Y = \sigma_Y \sqrt{\frac{1}{c^2 + d^2}} (cV + dZ) + \mu_Y$$

where V,W, and Z are unit normal, random gaussian numbers, and a,b,c and d are a function of the correlation (ρ). This resulted in a correlation coefficient:

$$\rho_{x,y} = \frac{ac}{\sqrt{(a^2 + b^2)(c^2 + d^2)}}$$

The values for a,b and c were chosen accordingly. A correlation coefficient of 0.75 between Nld and Pld was used. There is a high correlation between these parameters because their value is determined mostly by a shared process step, polysilicon etch. After finding the global values for Nld and Pld, these equations can be applied recursively to find the local values by using the local standard deviations, and the global values as the new means, as depicted in Figure 1.

Recently, work has been done characterizing the local variation of transistor parameters. Variation of threshold voltage, drain current (Spice parameter KP) and the body effect coefficient (Spice parameter BETA) have been modeled in terms of the size of transistors and their distance from each other. The variation in the circuit parameters corresponds to variation in the fabrication parameters postulated here.

Experimental work is necessary to establish relevant statistics about local parameter variation. Pelgrom [8] provides an interesting framework for that work. The statistical results could be used to generate better estimates of local variation for a specific fabrication line. This result could be used directly in the generation of the input parameter distribution for the Monte Carlo analysis.

2.1.1. Input Parameter Space

The input parameter space is the range of values each input parameter is allowed to take in any simulation. For gaussian distributions, each input parameter is described by a given mean and a global and local standard deviation. The values used in this project are given in Table I. These are typical values for a 1.2 micron CMOS process, which is currently used in high volume EPROM manufacturing.

Parameter	Mean	Global Std. Dev.	Local Std. Dev.	Units
Oxide thickness	40.0	1.67	0.42	nm
NMOS linewidth variation	0.3	0.1	0.025	microns
PMOS linewidth variation	0.25	0.083	0.02	microns
Doping	5.5×10^{17}	1.67×10^{17}	0.42×10^{17}	cm^{-3}

Table I: Input Parameter Space

2.2. Calculation of Performance Yield

Monte Carlo simulation techniques are used to evaluate the performance yield of two possible circuit designs. Given the performance specifications, the Spice results can be checked to see if each circuit succeeded or failed to meet all the specifications. The performance yield is

$$\text{Perf\_Yield} = \frac{(\text{\# of passing circuits})}{(\text{total \# of simulations})}$$

2.2.1. Use of the Yield Body in Binning Parts

Normally, each performance specification has a value, which defines the acceptable limit for a part to meet that specification. For example, a 25 nsec EPROM would have the value of 25 nsec for the Read Access Time specification. The manufacturer might also sell parts with 30 and 35 nsec maximum access times. These specification ranges, or bins, are used by manufacturers because fabricated ICs exhibit performance spread.

The performance yield prediction technique developed here can also be used to predict the number of parts a manufacturer will have in each bin. The percentage of parts in each bin can be calculated from the Monte Carlo simulation results. The yield bodies for adjacent bins will be adjacent regions in performance space.

2.3. Calculation of EPROM Sensitivity to Fab Parameters

The first order sensitivity of a circuit can be estimated from the change in performance for a unit change in an input parameter. The sensitivity is the percent change in performance from nominal per the given input parameter change.

2.4. Linear Model for Yield Body

The yield body is the area in input Parameter space where the resultant circuit will pass all the performance criteria. The yield body in this analysis will be a polytope in the twelve dimensional input space. It can be examined graphically through the used of projections onto the plane of two input parameters.

An estimate of the yield body can be generated by first assuming each performance specification generates one surface of the polytope in input space. If we postulate that the surface can be described as a linear combination of the input parameters, we can generate a linear model for the yield body of the EPROM. The model has the form:

$$\text{Performance} = A + B(\text{Nld}) + C(\text{Ntox}) + D(\text{Nsub}) + E(\text{Pld}) + \cdots$$

There will be one equation for each specification. This model has been successful in the analysis of digital circuits [2], however here it is being applied in an analog circuit. Although it is possible that a quadratic model is necessary, the linear model will be used for its simplicity.

3. Implementation

This analysis is based on the circuit model for an EPROM shown in Figure 2, which is a simplified schematic of the model used for Spice simulation. Figure 2 includes the static pullup on the bit line. The p-channel transistor with W/L=4/2, pulling up on node (11), is the static pullup. The static pullup transistor brings the bit line to a logic one when the EPROM cell is off. Figure 3 shows a diagram of the precharging

scheme. This circuit includes Address Transition Detection to generate a precharge pulse which aids bit-line precharging. The precharge pulse is about 5 nsec wide after any address transition.

The 500 point Monte Carlo simulation was generated and run on a Decstation 3100. Shell scripts, utilizing awk and the C language pre-processor, generated 500 Spice decks with varying transistor models according to the statistical model described in section 2.1. The Monte Carlo required approximately 5 CPU hours to complete.

3.1. Linear Model for the Yield Body

The analysis was done for the static and precharge circuits. The six fabrication parameters which had the greatest on performance were chosen for the analysis. (The sensitivity of the circuit performance to fabrication parameter variation will be discussed in section 4.2.) These are Nld, Ntox, Nsub, Pld, Psub and Nld_local. These six variables were normalized and then put through Principal Component Analysis to form independent variables for linear regression. Five principal components were enough to account for 96 percent of the total internal variation. The EPROM performance specifications used are shown in Table II. These specifications apply to the 1 bit EPROM model used for simulation.

Specification	Upper Limit
Power	2.1 mW
Read 0 Access Time	23.4 nsec
Read 1 Access Time	23.4 nsec

Table II: Performance Specifications

4. Results

The main results of the analysis will be discussed. These include the performance yield of the two circuits, the circuits' sensitivity to parameter variation, the model for the yield body and the results of the circuit optimization.

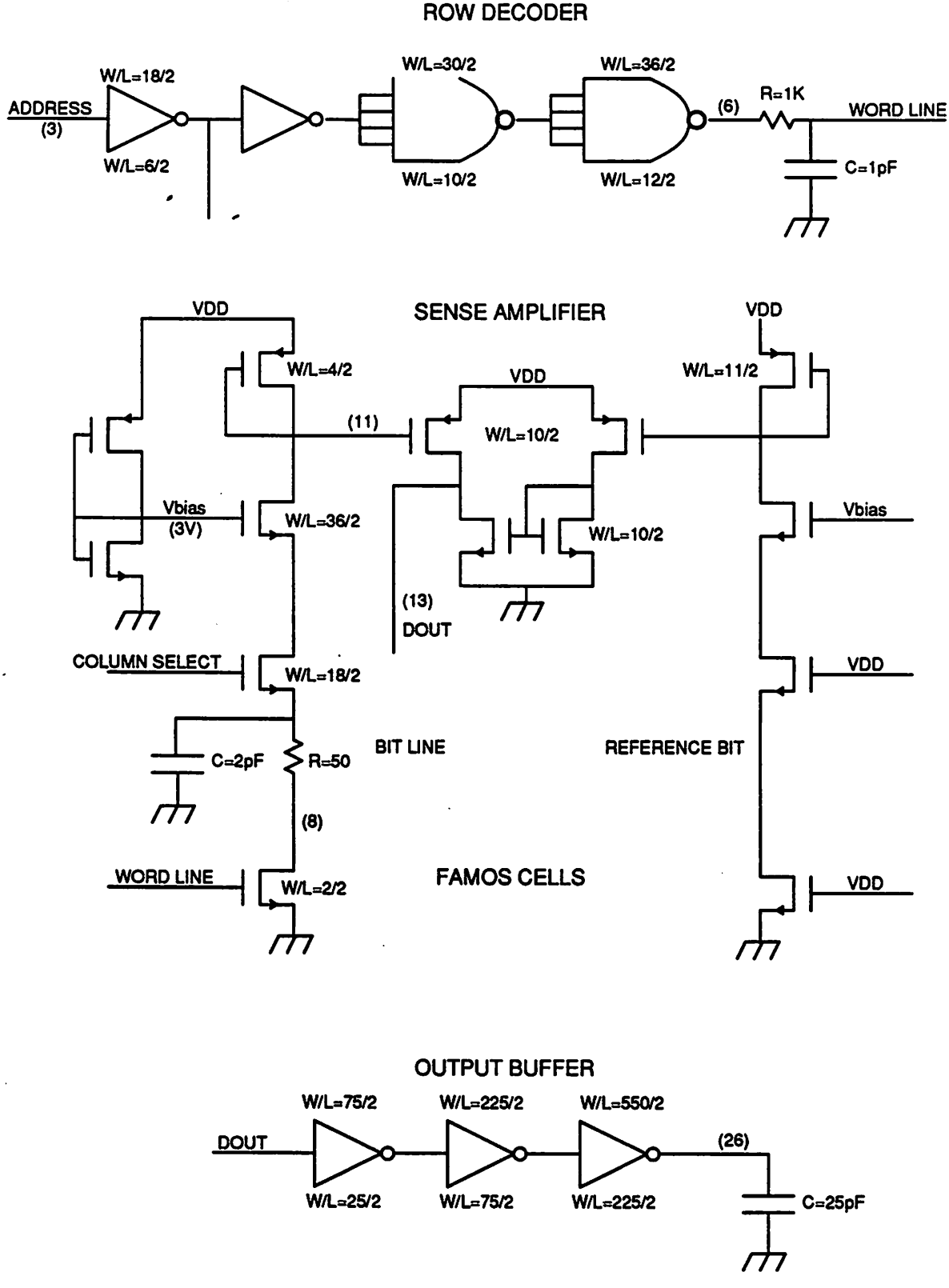
4.1. Prediction of the Monte Carlo Simulation for Performance Yield

Appendices A and B show the access time results of the Monte Carlo simulations of the static pullup and precharge circuits, respectively. The two circuits were simulated 500 times, with the same set of parameter variation. Read 0 is the time it takes to read a programmed cell, that is, with the memory cell pulling down the bit line. Read 1 is the time when the bit line is high. The read access time specification on a part would be the larger of the two times.

The interesting result is that the static design produces a slightly higher performance yield given a fast access time specification, but the precharge design produces a higher performance yield at a slower access time specification. This can be seen in Appendix C, which shows the simulation results in histogram form. Specifically, the static pullup has a performance yield of 18.2 percent at 21 ns access time and 76.6 percent at 25 ns. The precharge circuit has yields of 16.2 percent and 78.4 percent at the two speeds. This performance yield prediction is an important result of this analysis technique, although here the result is not statistically significant.

Also note that the precharge design is more sensitive to process variation, as seen in the wider distribution of performances for the same parameter variation. However, the circuit performance criteria which varies the most, Read 1 delay, is never the limiting value for the speed of the precharge circuit, where Read 0 is slower. So, one benefit of the precharge circuit, is that one of the performance specifications does not effect the yield. This will potentially simplify the testing procedure of the product.

Figure 2 - EPROM Circuit Model with Static Pullup



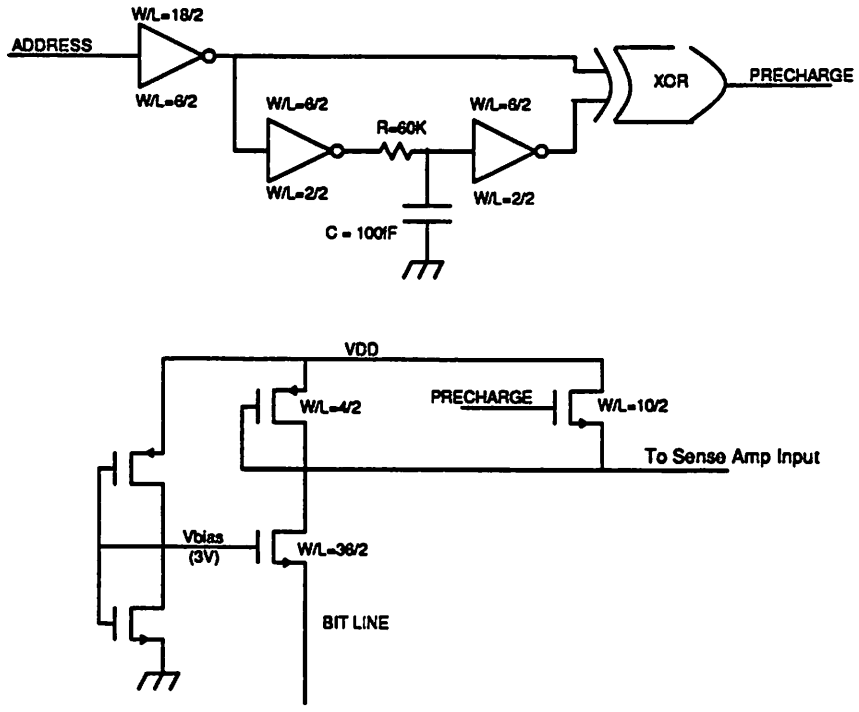


Figure 3 - Bit Line Precharge Circuitry

4.2. 4.2 Circuit Sensitivity to Fabrication Variation

Table III quantifies the performance sensitivity of the two circuits to the twelve fabrication parameters. It shows the percent change from nominal delay for Read 0 and Read 1 Access Times for three sigma changes in all input parameters. There are two entries for each parameter, one corresponding to a change to it's plus three sigma limit (denoted 'hi' in Table III), and one corresponding to a change to the minus three sigma limit (denoted 'lo'). In other words:

$$\text{Sensitivity} = \frac{t_{3\sigma\ input} - t_{nominal}}{t_{nominal}} * 100\ \%$$

It now becomes clear how the six input parameters for Principal Component Analysis were chosen. They are the parameters which have the highest sensitivities in both designs.

The most important result, in addition to the selection of the parameters for Principal Component Analysis, is the increased sensitivity of the precharge circuit for Read 1. Also notice the large sensitivities of some of the local variation, such as nld_local.

4.3. Yield Body Equations and Projections

The result of the Principal Component Analysis is shown in Table IV. The Principal Components are linear combinations of the input parameters. These linear combinations are uncorrelated with each other, and are used for linear regression. A linear regression was done for each performance constraint for both the static pullup and precharge circuits. An example of the regression results is shown in Tables V and VI. Table V shows the coefficients generated for Read 0 delay for the static pullup circuit, and Table VI shows the Analysis of Variance Table.

Static Circuit			Precharge Circuit	
Variable	Sens of Read 0	Sens of Read 1	Sens of Read 0	Sens of Read 1
(in % change from nominal delay)				
Nominal Delay	22.2	22.4	23.0	15.0
nld hi	-14.946846	-15.683648	-15.093441	-27.891643
nld lo	15.891987	16.627216	19.129866	56.642769
ntox hi	7.520718	3.296313	9.867941	30.101246
ntox lo	-8.272898	-4.109828	-7.362490	-13.071564
nsub hi	5.062577	-1.171403	7.351490	27.601801
nsub lo	-6.931844	10.051883	-5.718565	-21.953356
pld hi	-9.385125	-17.485166	-5.020423	6.183117
pld lo	15.675400	21.700668	16.591416	12.606497
ptox hi	-0.343937	1.530004	-1.239726	6.666775
ptox lo	1.045132	-1.638262	1.945563	-0.876574
psub hi	0.089686	5.697237	2.368283	-4.474786
psub lo	-1.314953	-7.616061	2.052906	14.541392
nld_l hi	-0.199531	-1.217042	1.428413	-2.922239
nld_l lo	-4.015016	9.960784	-3.803739	38.448105
ntox_l hi	-2.519746	5.416097	-2.740701	25.924131
ntox_l lo	2.885328	-5.506483	3.275720	-12.822298
nsub_l hi	-2.476546	4.284182	-2.628619	25.744210
nsub_l lo	2.986489	-5.570484	4.057728	-13.692478
pld_l hi	0.483303	0.084503	-0.191948	-1.360167
pld_l lo	-1.193363	0.633947	-0.434502	2.765803
ptox_l hi	0.111286	-0.128091	-0.003652	1.650075
ptox_l lo	0.021150	0.186432	-0.021912	0.189314
psub_l hi	-0.889431	0.275792	-0.054345	2.370735
psub_l lo	0.136126	0.175066	-0.145037	-0.216061

Table III - First Order Sensitivities

Parameter	Coeff1	Coeff2	Coeff3	Coeff4	Coeff5
nld	0.69453	-0.02964	-0.053303	-0.122391	0.018589
ntox	0.062938	0.189548	-0.796532	0.502942	0.267476
nsub	0.079627	0.908271	0.254647	0.190265	-0.259428
pld	0.697986	-0.026514	0.005215	-0.112795	-0.007503
psub	0.14145	-0.301527	0.46987	0.815682	0.048916
nld_local	-0.011624	0.215907	0.27757	-0.133448	0.926477
Variance	0.187657	0.128945	0.118086	0.11196	0.101417
% of total var	27.811492	19.11007	17.50075	16.592803	15.030285
Cumulative %	27.811492	46.921561	64.422312	81.015115	96.0454

Table IV - Principal Component Coefficients

Parameter	Coeff	Std. Err	T. Value	Sig
CONSTANT	22.090711	0.020703	1067.01795	0.0001
pco1	-4.265672	0.047735	-89.361343	0.0001
pco2	2.053502	0.057586	35.659593	0.0001
pco3	-0.384407	0.060176	-6.388076	0.0001
pco4	1.960178	0.0618	31.717967	0.0001
pco5	0.675375	0.064933	10.401083	0.0001

Table V - Least Squares Regression for Read 0 Delay, Static Pullup

Source	Sum of Squares	df	Mean Square	F Value	Signif. level
Regression	2221.6685	5	444.3337	2082.4151	2.2204e-16
Error	105.4068	494	0.2133		
R-Squared	0.9547				
Adjusted R-Squared	0.9542				

Table VI - Analysis of Variance Table

During the residuals check, it was seen that the residuals for the power equation for the static pullup were not randomly distributed. Therefore, a quadratic model was fitted and the results are shown in Table VII.

Parameter	Coefficient	Standard error	T value	Signif. level
CONSTANT	1.976475	0.000390	5066.137744	0.000100
pco1	0.150232	0.000472	318.233116	0.000100
pco2	-0.087723	0.000583	-150.480379	0.000100
pco3	0.024678	0.000603	40.941383	0.000100
pco4	-0.164328	0.000616	-266.889315	0.000100
pco5	-0.003553	0.000649	-5.476820	0.000100
pco1**2	0.032975	0.000767	43.012301	0.000100
pco1*pco2	-0.010978	0.001307	-8.400517	0.000100
pco1*pco3	0.006947	0.001473	4.716252	0.000100
pco1*pco4	-0.022612	0.001445	-15.651587	0.000100
pco1*pco5	-0.001416	0.001516	-0.933862	0.350845
pco2**2	0.024284	0.001108	21.922794	0.000100
pco2*pco3	0.009427	0.001628	5.790682	0.000100
pco2*pco4	0.006959	0.001728	4.026329	0.000100
pco2*pco5	-0.013881	0.001796	-7.730116	0.000100
pco3**2	0.012940	0.001336	9.684334	0.000100
pco3*pco4	0.010109	0.001846	5.476969	0.000100
pco3*pco5	-0.009267	0.001927	-4.810113	0.000100
pco4**2	0.020986	0.001375	15.258998	0.000100
pco4*pco5	-0.003049	0.001803	-1.691058	0.091476
pco5**2	0.003514	0.001472	2.386623	0.017391

Table VII - Regression of the Power Equation Using a Quadratic Fit

Appendices D through G show the results of the Monte Carlo simulation projected onto planes of two input variables. Both circuit results are shown projected onto two planes, Nld and Ntox, and Nld and Pld. Each point represents one run of the simulator, at that point in input space. The point will be a solid square if that circuit passed all performance specifications, and an outlined square if the circuit failed any specification.

These regression results are plotted as lines on these figures. Note that the regression generated an expression for delay or power in terms of principal components of the normalized values for the varying parameters, Nld, Ntox, Nsub, etc. By setting the delay equal to the performance limit (i.e. Read 0 access time to 23.4 ns), and setting the ten other variables to their mean (which is zero, due to the normalization), and expanding out the principal components back into input parameter space, the lines shown in the figures are generated. The lines shown are projections of the yield body into a two dimensional space. The regression was done on normalized data, so the axes were labeled in terms of the distance from the mean expressed in number of standard deviations.

The lines generated by the access time specification generally fall on a sharp division for passing and failing circuits. This is less true for the power constraint. Certainly, the simplified analysis generates an interesting projection of the yield body. The quadratic regression for power is also plotted on Figure 10. It is not significantly different from the linear model.

These lines show the projection of the twelve dimensional yield body into two dimensions. This projection is done as if all other (ten) variables were simulated at their mean value. Since this was not the case, the inaccuracy of the linear model is due to both the inaccuracy of the model, and our simplified projection, ignoring the variation of the other parameters. Nevertheless, this simplification allows us to get a graphical view of the importance of various fabrication parameters.

Note that the yield body for the precharge model is slightly smaller than that for the static pullup. This is another sign of its increased sensitivity. It is also clear that one specification for the precharge model, Read 1 delay, has no effect on yield, and that the yield body is not centered around the process mean (Nld of 300 nm, Pld of 250 nm and ntox of 40 nm). If it were possible to move the center point of the fab parameters to the center point of the yield body, yield could be increased. This technique is known as Design Centering.

4.3.1. Alternative Projections of the Yield Body

The projection of the yield body as done above gives an interesting, but slightly incomplete picture of how well the linear model has established the boundary of the true yield body. This measure can be improved by projecting the yield body onto a plane perpendicular to the constraints. Projection onto this plane would give a better picture of which side of the constraint plane the simulation results were on. The projection as done above suffers from a 'shadowing' problem since the constraint plane is hitting the projection plane at an angle.

For each constraint, expressed as a linear combination of the principal components:

$$\text{Performance} = A + B(pco1) + C(pco2) + D(pco3) + E(pco4) + F(pco5)$$

the vector [A,B,C,D,E,F] is normal to the plane defined by the constraint, and becomes a basis vector for the desired plane of projection. For a space of dimension m, m-1 basis vectors are needed to uniquely define a plane. In this case, three constraints are not enough to define a unique plane in our 5 dimensional component space. Further, each basis vector must be independent to define the plane. Here, we are in an underdetermined situation, and must create additional constraints to define the projection plane. This analysis was not done for this project.

4.4. Performance Yield Optimization

The modeling of performance yield allows it to be added as a criterion for optimization. One interesting direction would be to express performance yield as a function of the circuit and fabrication parameters, and use that as an optimization constraint.

Another, graphical technique, is based on the normal to the constraint planes. For each constraint plane, the normal to that plane establishes the direction of maximum sensitivity of the circuit to the input parameters. Relating the principal component space back into input parameter space would allow the calculation of the sharpest yield derivatives of the input parameters. This information can be used for performance yield optimization.

5. Conclusions

This analysis clearly showed the power of statistical design tools to explore many performance related features of a design which are not normally considered while doing worst case design. However, in this case, there was not a particularly significant difference in these two circuits. The lack of sensitivity to input parameters of the static pullup makes it a more manufacturable circuit, regardless of the slight performance yield disadvantage at slower access times.

5.1. Ideas for Current Research

This project is part of an ongoing research project in statistical circuit design. Two extensions of this work are currently being pursued. First, the EPROM circuit model is being enhanced to better represent a commercial EPROM design. Secondly, formal mathematical techniques for generating the yield body, the planes of projection parallel to the constraints and the normals to the yield body planes are being investigated. The information contained in a circuit's yield body can be used to understand why the circuit will

fail to meet performance specifications in a varying manufacturing environment. This information will be used to generate a test pattern to monitor the performance yield of the circuit.

5.2. Ideas for Further Research

An extension of this analysis technique would be to characterize the effect of fab parameter variation on the variation of high level circuit parameters for small analog circuit blocks. Instead of doing a worst case design for a small circuit block, characterization could predict the spread of analog circuit performance for a given manufacturing variation. Macromodels could be developed for use in the Spice simulator.

6. Acknowledgement

We are grateful to Mr. George Korsh and to Dr. Gust Perlengos from ATMEL for many helful discussions. We are also grateful to ATMEL and to the state of California for funding this project under the California Competive Technologies Program.

References

[1] E.M. Butler, "Realistic Design Using Large-Change Sensitivities and Performance Contours," IEEE Trans. on Circuit Theory, Vol. CT-18, No. 1, pp. 58-66, January, 1971.

[2] P. Cox, P. Yang, S. Mahant-Shetti, P. Chatterjee, "Statistical Modeling for Efficient Parametric Yield Estimation of MOS VLSI Circuits," IEEE Journal of Solid-State Circuits, Vol. SC-20, No. 1, pp. 391-398, February, 1985.

[3] D.E. Hocevar, P. Cox and P. Yang, "Parametric Yield Optimization for MOS Circuit Blocks," IEEE Trans. on CAD, Vol. 7, No. 6, pp. 645-658, June, 1988.

[4] S. Inohira, T. Shinmi, M. Nagata, T. Toyabe, K. Iida, "A Statistical Model Including Parameter Matching for Analog Integrated Circuits Simulation," IEEE Trans. on CAD, Vol. CAD-4, No. 4, pp. 621-628, October, 1985.

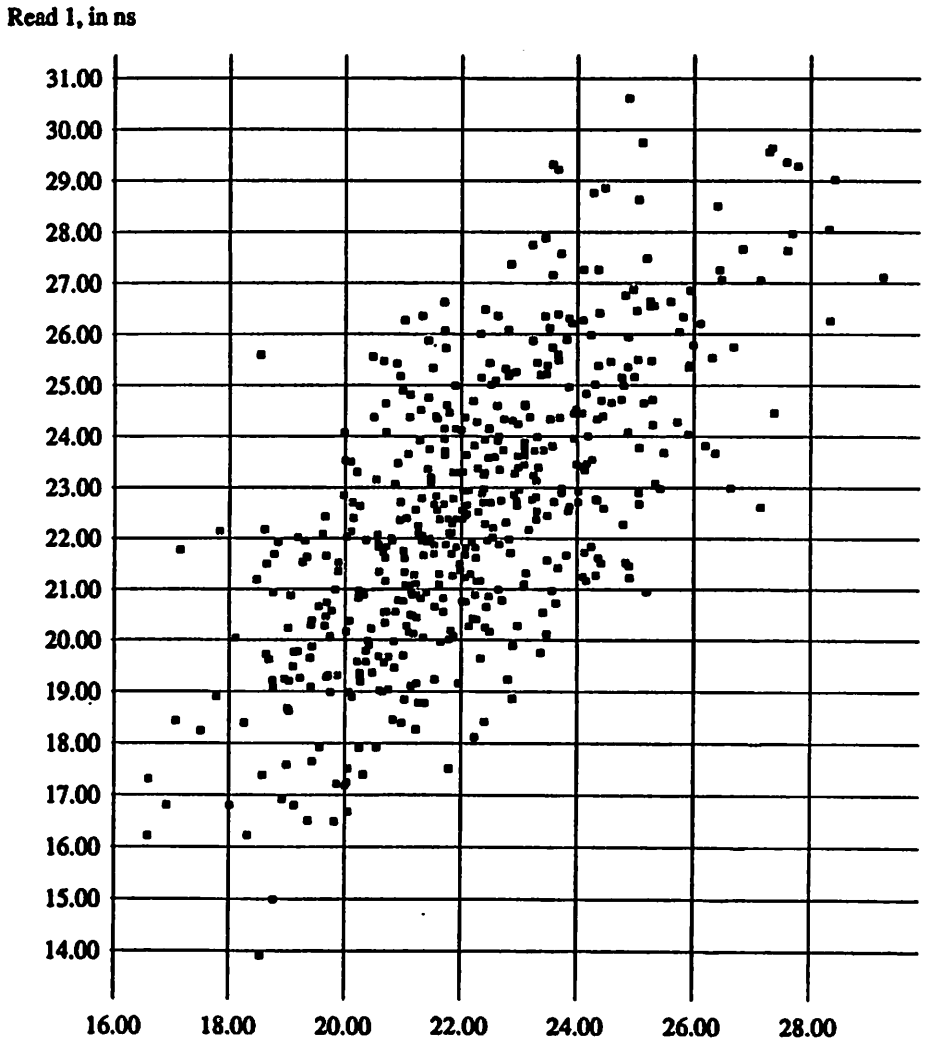
[5] Visvanathan, V., "Variational Analysis of Intergrated Circuits," Proc. of International Conference on CAD, pp. 228-231, November, 1986.

[6] T.K. Yu, S.M. Kang, I.N.Hajj and T.N. Trick, "Statistical Performance Modeling and Parametric Yield Estimation of MOS VLSI" SRC report C87319, October 1987.

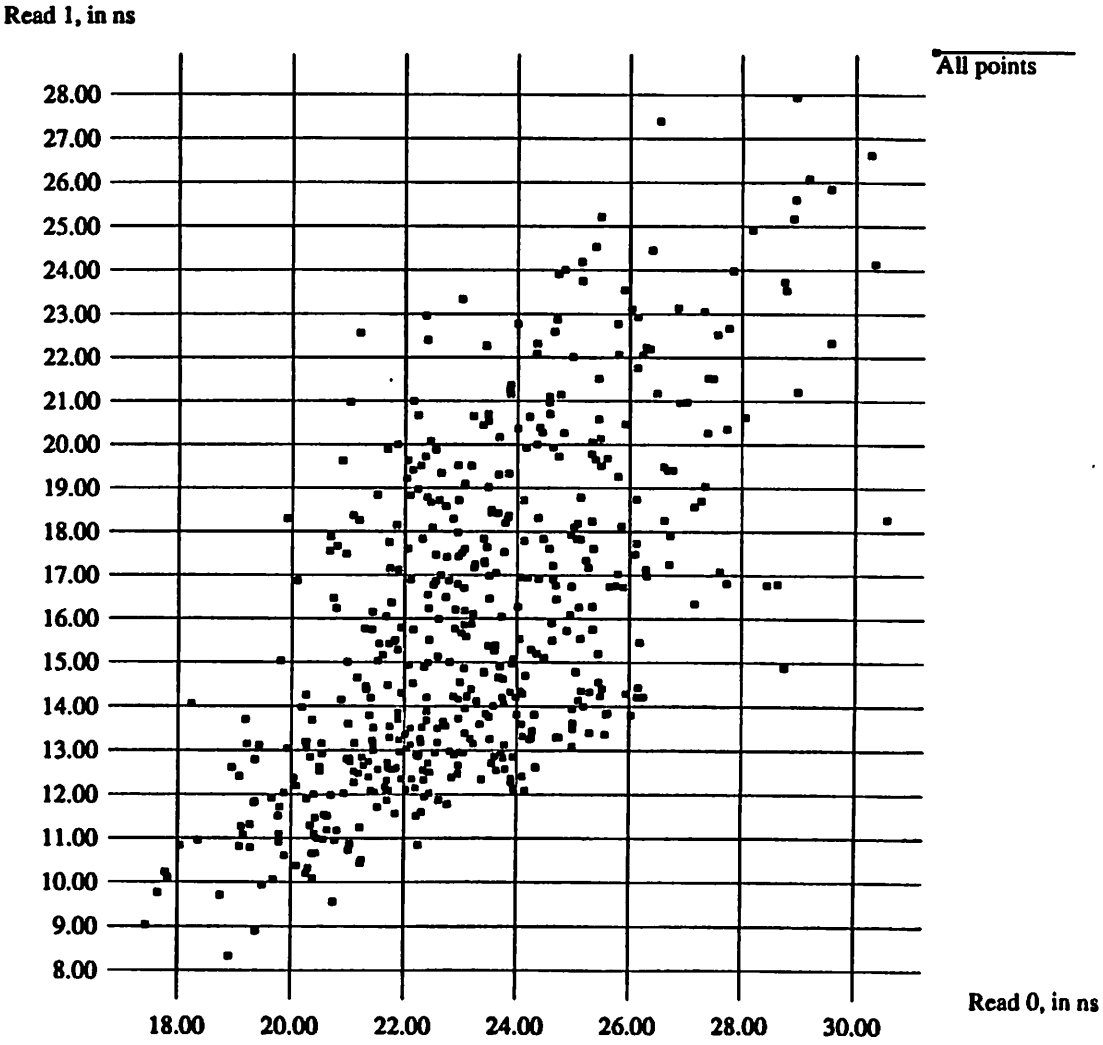
[7] D. Hoff, et al, "A 23-ns 256K EPROM with Double-Layer Metal and Address Transition Detection," IEEE JSSC, Vol. 24, No. 5, pp. 1250-1258, October 1989.

[8] M. Pelgrom, A. Duinmaijer, A. Welbers, "Matching Properties of MOS Transistors," IEEE JSSC, Vol. 24, No. 5, pp. 1433-1440, October, 1989.

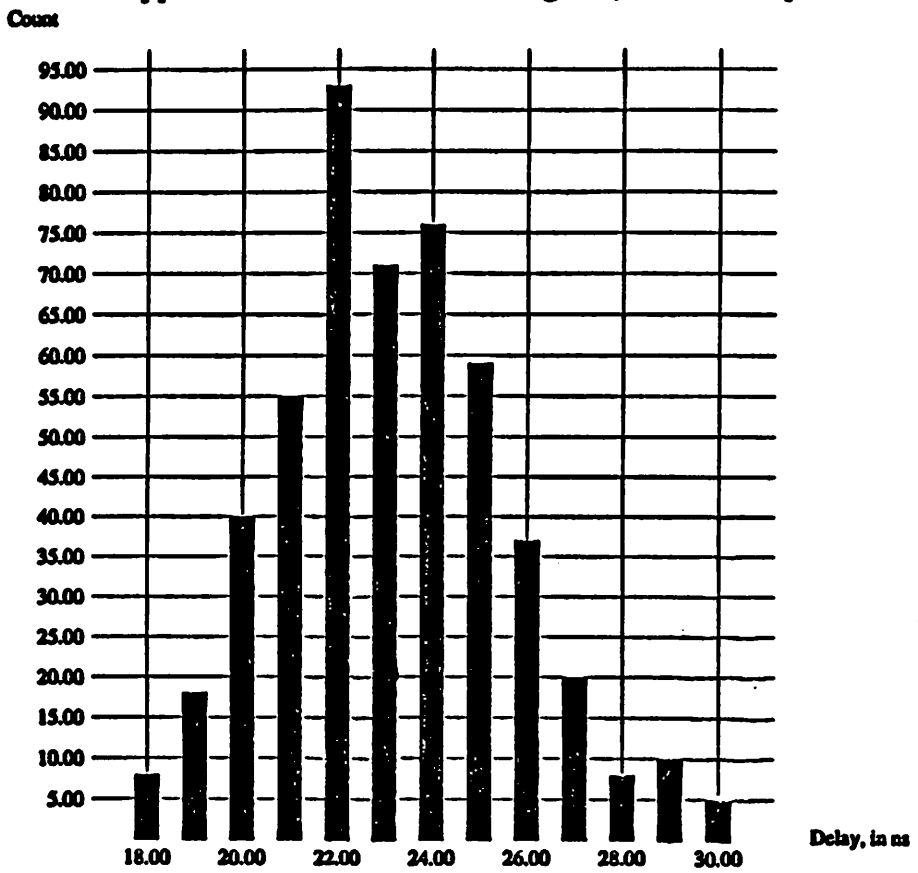
Appendix A - Access Time with Static Pullup



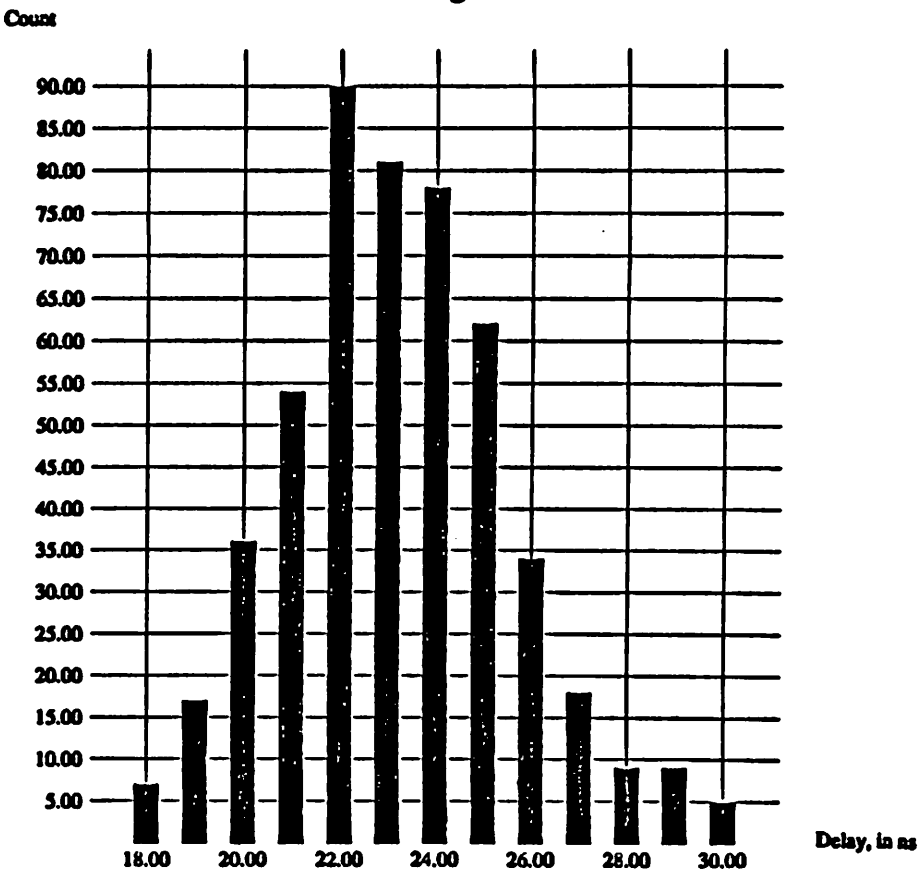
Appendix B - Access Time with Precharge



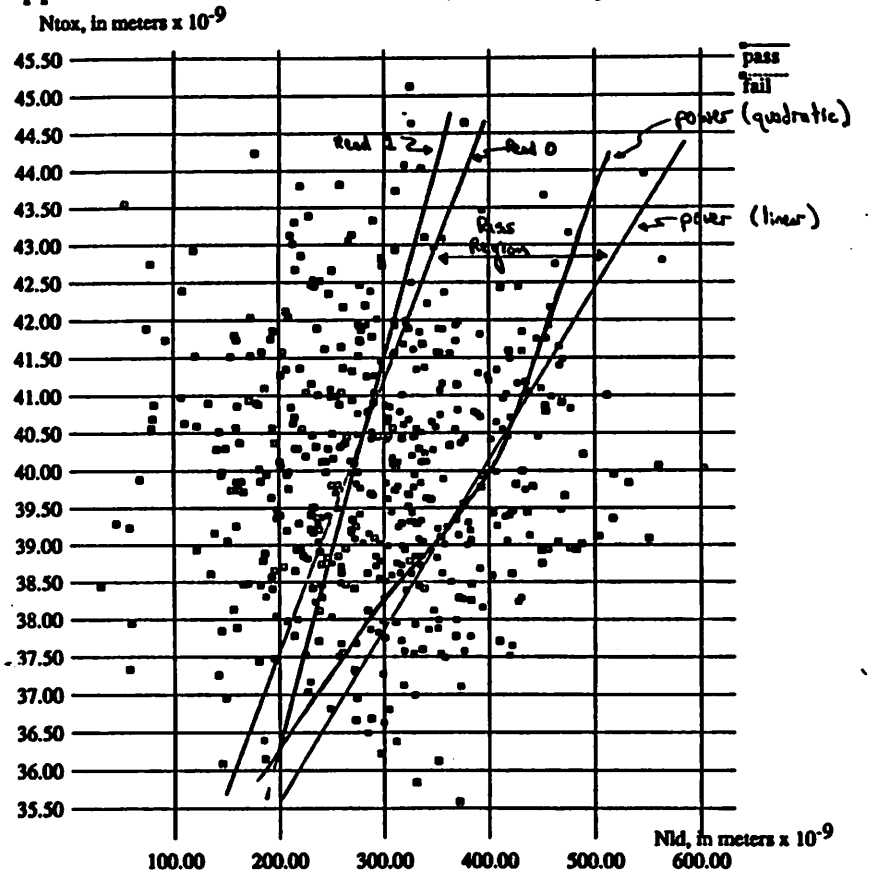
Appendix C - Access Time Histogram, Static Pullup



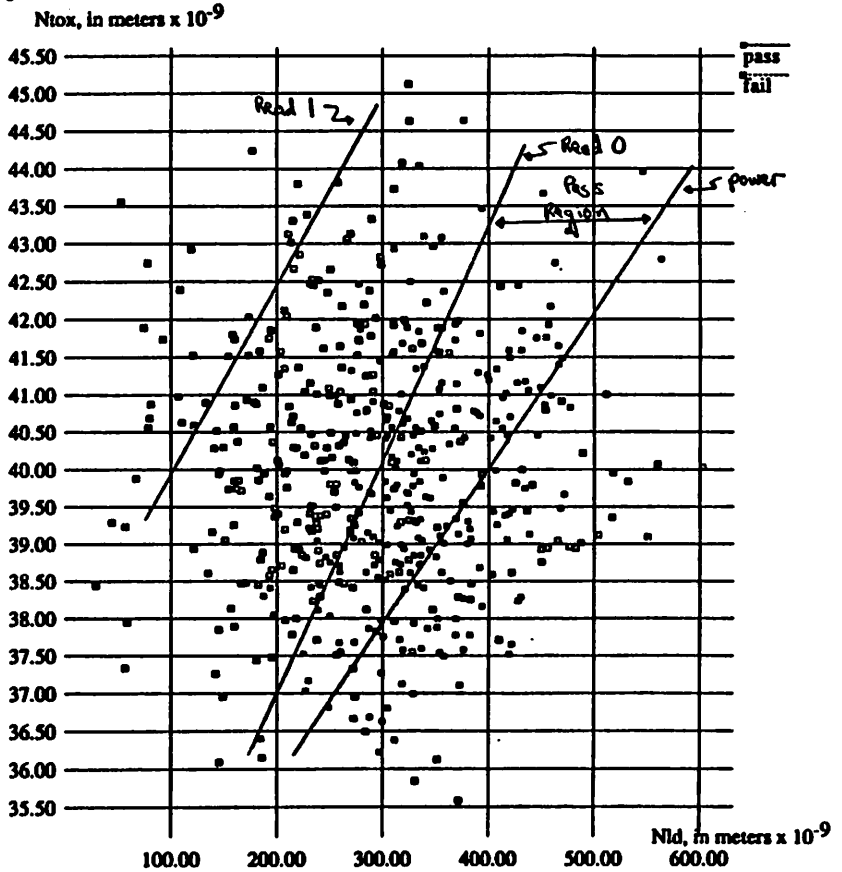
Precharge Circuit



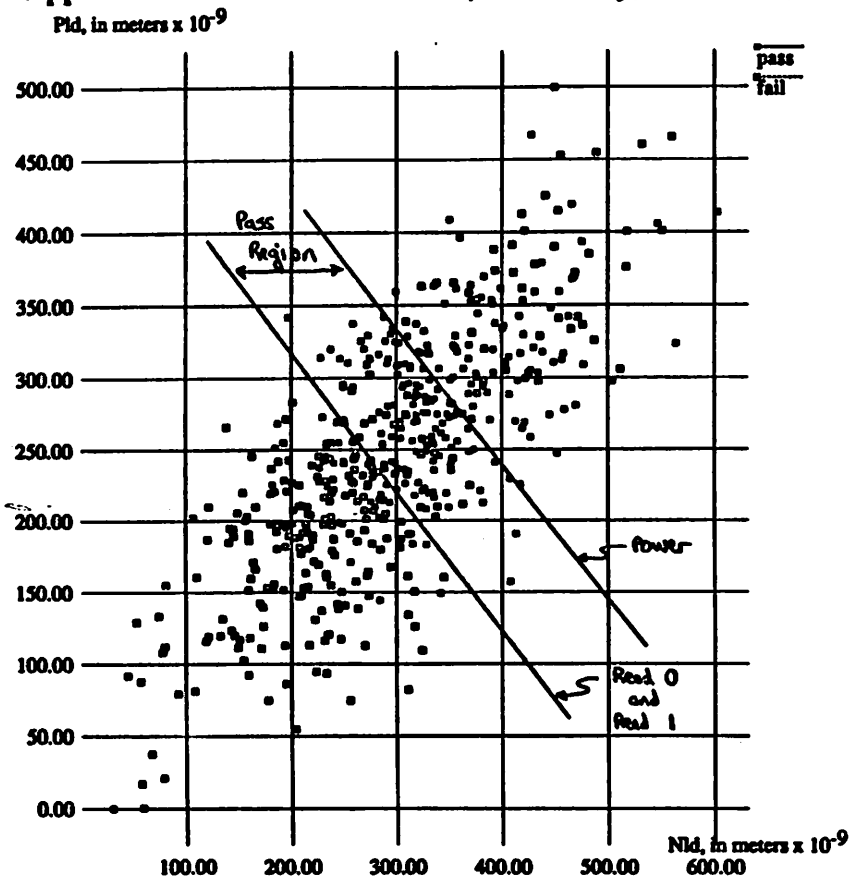
Appendix D - Monte Carlo Result, Yield Body, Nld vs Ntox, Static



Appendix E - Monte Carlo Result, Yield Body, Nld vs Ntox, Precharge



Appendix F - Monte Carlo Result, Yield Body, Nld vs Pld, Static



Appendix G - Monte Carlo Result, Yield Body, Nld vs Pld, Precharge