# SEQUENTIAL OPTIMIZATION IN SEMICONDUCTOR MANUFACTURING

by

John Thomson

Memorandum No. UCB/ERL M92/118

29 October 1992

# SEQUENTIAL OPTIMIZATION IN SEMICONDUCTOR MANUFACTURING

by

John Thomson

Memorandum No. UCB/ERL M92/118

29 October 1992

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering
University of California, Berkeley
94720

# SEQUENTIAL OPTIMIZATION IN SEMICONDUCTOR MANUFACTURING

by

John Thomson

Memorandum No. UCB/ERL M92/118

29 October 1992

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering
University of California, Berkeley
94720

# SEQUENTIAL OPTIMIZATION IN SEMICONDUCTOR MANUFACTURING

by

John Thomson

## ABSTRACT

The optimal and timely control of semiconductor manufacturing equipment is crucial for a successful fabrication line. Responding to the inevitable equipment shifts or changes in output specifications must be made on-line, without having to stop the process and re-characterize the equipment. Sequential optimization approaches attempt to accomplish this for single equipment steps whose output specifications are known. In addition, since typical processes require numerous steps, some compensation must be made for the interaction of consecutive steps.

This thesis will overview the important issues in sequential optimization, and describe two different implementations: evolutionary operation and Ultramax. A dynamic specification strategy for multi-step processes will then be presented.

## ACKNOWLEDGEMENTS

I would like to thank my research advisor, Professor Costas Spanos, for his guidance and helpful suggestions throughout the course of this project. I also thank Professor A. R. Neureuther for his valuable comments.

The members of the BCAM group, Bart Bombay, Eric Boskin, Raymond Chen, Zeina Daoud, Mehdi Hosseini, Sovarong Leang, Sherry Lee, Hao-Cheng Liu, Lauren Massa-Lochridge and Eddie Wen have made my time at Berkeley much more interesting than it otherwise could have been. The friendships of many other students have been and continue to be invaluable.

I would like to acknowledge the work of Gary May for creating the structure of the C++ equipment models used in this work and also of Sovarong Leang for conducting the experiments to characterize the equipment in the photolithography workcell. Eric Braun assisted with the generation of the response surface plots. Carlos Moreno from Ultramax Corporation has provided crucial information on using the Ultramax software.

I am grateful to my parents who have always been a tremendous source of encouragement and support.

This research has been supported by the Alberta Heritage Scholarship Fund, the Teagle Foundation, the University of California Regents and the Semiconductor Research Corporation.

## Table of Contents

## List of Figures

# Chapter 1 Introduction

## 1.1 Background and Motivation

The increasing complexity and speed requirements of modern integrated circuits have placed stringent demands on the performance of manufacturing equipment. Determining the optimal input settings and output specifications for each process step has become a non-trivial task. The control algorithms must continuously respond to shifts in equipment behavior or changes to output specifications. When a change takes place in a high volume manufacturing environment, re-characterizing the fabrication line through a series of off-line experiments is often not feasible. To avoid disrupting the product flow, the necessary adaptations must occur immediately.

Design of experiments (DOE) is a useful tool that has been applied on several manufacturing processes. Traditional DOE examines a wide variety of input combinations in order to deduce the process response. Such a technique is useful for developing initial equipment settings.

To ensure that a process is running optimally in a production environment, experiments may be performed while useful product is being produced. During such a sequential design of experiments, a compromise is made between exploring alternative recipes for enhanced performance, and continuously running the process at the same operating point to ensure adequate results. In this document, we report on the implementation of an evolutionary operation (EVOP) algorithm that uses fractional factorial experiments. We have also experimented with Ultramax [1], a sequential optimizer that has found application in many manufacturing environments. Ultramax uses a variant of the EVOP algorithm to generate advice for process runs.

Sequential design of experiments requires a measure of performance to optimize. For single equipment processes, this measure of performance can be easily derived from the given output specifications. For manufacturing sequences that consist of numerous steps, the specifications for the final step are given, but the optimal specifications for the intermediate steps must be determined. It is important to find a consistent set of specifications among all processing steps since they are dependent upon each other. Determining the best target range for each step is affected by shifts in the equipment response and changes in the final output specifications. A dynamic specification strategy for reacting to these changes is necessary. Such a strategy has been developed and is also presented in this document.

## 1.2 Thesis Organization

The remainder of this report is organized as follows. Chapter 2 will discuss generic issues in sequential optimization. In chapter 3, the details of EVOP, its implementation and simulation results will be presented. Chapter 4 overviews the structure of Ultramax and a comparison to EVOP. The approach taken to optimize a multi-step manufacturing sequence is contained in Chapter 5. Finally, simulated results and conclusions are provided in Chapter 6.

# Chapter 2 Issues in Sequential Optimization

This chapter introduces a variety of fundamental topics in sequential optimization that will be referenced in the remainder of the report.

## 2.1 Equipment Models

Accurate equipment models are a vital part of a manufacturing control system [2]. Physical models can be developed using theoretical principles, but quite often the underlying assumptions may not be valid. Instead, two other approaches dominate the modeling techniques evident in modern semiconductor manufacturing. The first method uses quadratic response surface models. The second method employs semi-empirical models where the functional form is influenced by theoretical considerations, and the various coefficients are tuned to match experimental results.

Once the form of the model has been determined, the coefficients are found by performing a regression using historical data. In order to accurately represent equipment whose response is constantly changing, the regression is typically weighted to emphasize more recent data. In this analysis, it is advantageous to scale the inputs and outputs to be within reasonable ranges so that numerical round-off errors are minimized.

## 2.2 Feedback and Feed-Forward Control

Feedback control is a popular technique to improve the performance of manufacturing equipment. Models of the equipment are generated and then used to determine the optimal settings of the inputs. Feed-forward control corrects for processing variations in previous steps. Parameters of the incoming wafers are examined and then the controllable inputs are adjusted. A more thorough discussion of feedback and feed-forward control is presented in [3].

## 2.3 Process Capability and Process Observability

Process capability ($C_{pk}$) is an indicator of the suitability of a manufacturing step for a given task. Under the assumption that the output is normally distributed and under statistical control, $C_{pk}$ is defined to be

$$C_{pk} = \frac{\min(USL-\mu, \mu-LSL)}{3\sigma_{exp}} \quad (1)$$

where USL and LSL are the upper and lower specification limits, $\mu$ is the mean output value and $\sigma_{exp}$ is the standard error of the output. For a given process, the maximum attainable capability is achieved when the inputs are fixed to values which will produce outputs at the center of the specifications.

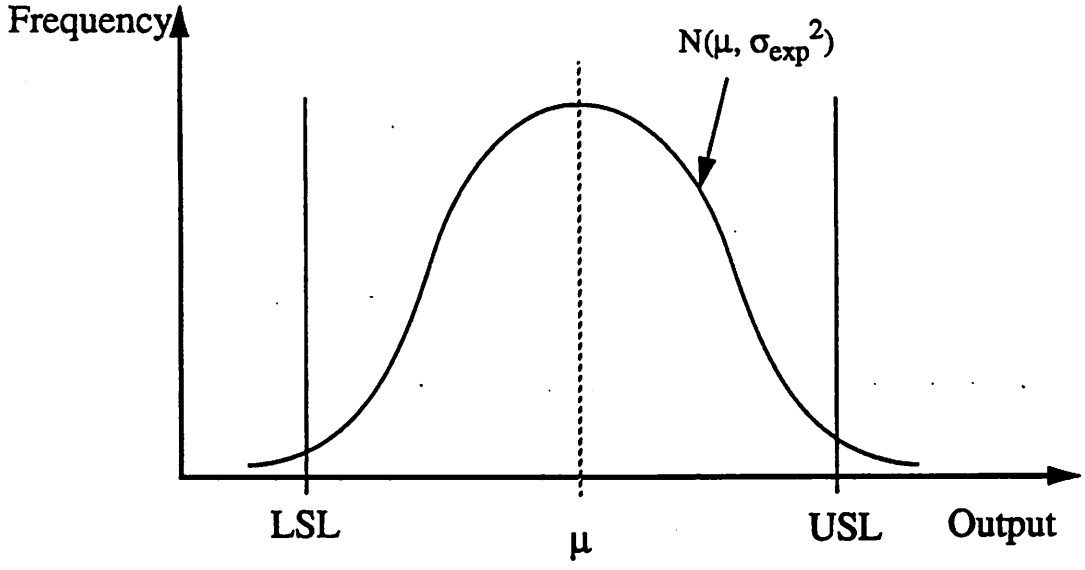


Figure 1. Process Capability

Observability refers to the ability to infer the equipment behavior and to recognize when changes have occurred. Sequential design of experiments achieves observability by continuously introducing changes to the inputs and monitoring the output, even if the current operating point is at the optimum. A local model of the process is constructed and

later used for optimization. As shown in figure 2, if the deviations introduced are insignificant, then the effects of altering input parameters will be masked by noise, decreasing the observability. Conversely, if large deviations are introduced, then the effects of the inputs will become visible, but then the process capability will be reduced. Thus, observability and capability are conflicting goals.

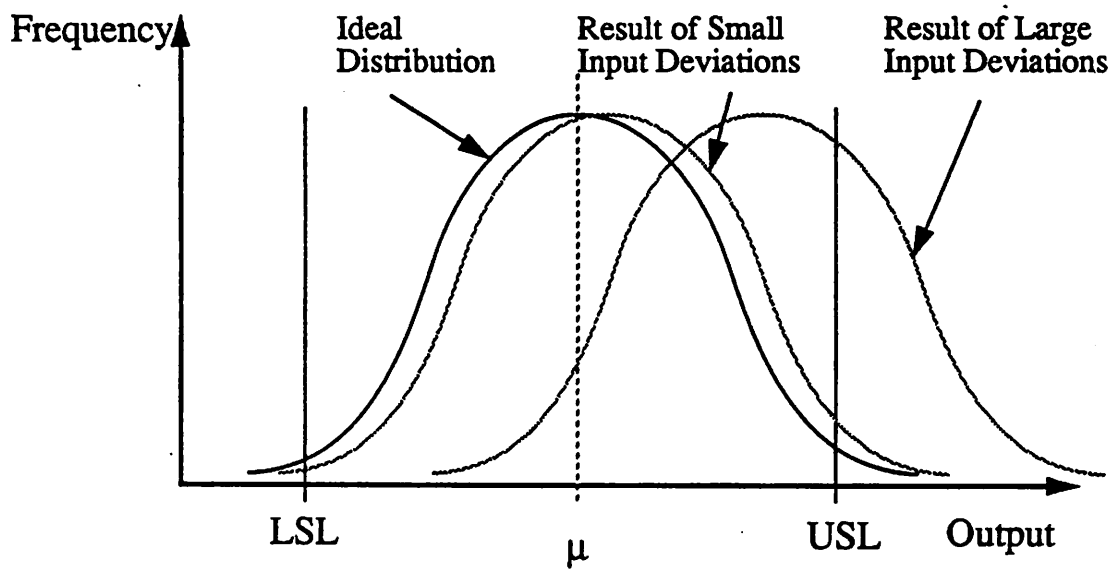


Figure 2. Capability and Observability Trade-off For an Optimized Process

Ideally, when the current operating point is at its optimum value, no deviations should be introduced and therefore no degradation in process capability will occur. In practice, small deviations are inserted so that changes in equipment behavior can be detected. If the current process capability is relatively low, then large deviations should be employed to increase the observability and to quickly shift the operating point to maximize the capability. Large deviations should be used for initial equipment characterization or as a reaction to equipment shifts.

## 2.4 Cost Functions

Manufacturing steps typically have a target and specifications for each of their outputs. The target is the ideal output value and the specifications indicate the range for the output to be considered acceptable. It is important to ensure that all outputs are within these ranges; otherwise, unacceptable products will be processed. One method to help influence all outputs to be within specifications is by using a cost function. The primary purpose of the cost function is to collapse the information of an output vector into a single measure of performance to be minimized by the controller. A relatively low cost indicates that the outputs are close to their targets.

A cost function should be chosen so that attention is focused on outputs which are currently farthest from their targets since these outputs most seriously degrade the overall process capability. On the other hand, outputs which are currently close to their targets should not be completely ignored in the cost function, otherwise they would be allowed to drift away from their targets without affecting the cost. It is important to ensure that all outputs are within specification, since a single output outside of its specifications will result in an unacceptable product.

### 2.4.1 Choice of the Cost Function

Numerous cost functions are possible. For a single output, Taguchi [4] recommends a cost function of the form:

$$\text{Cost} = k(y - y_{target})^2 \tag{2}$$

where $y$ is the output, $y_{target}$ is the target value, and $k$ is a constant.

Using such a function promotes an effort for continuous process improvement even when the output is within specifications. Typically, the scaling factor $k$ is chosen so that the cost is equal to 1 when the output approaches its specification, as shown in figure 3.

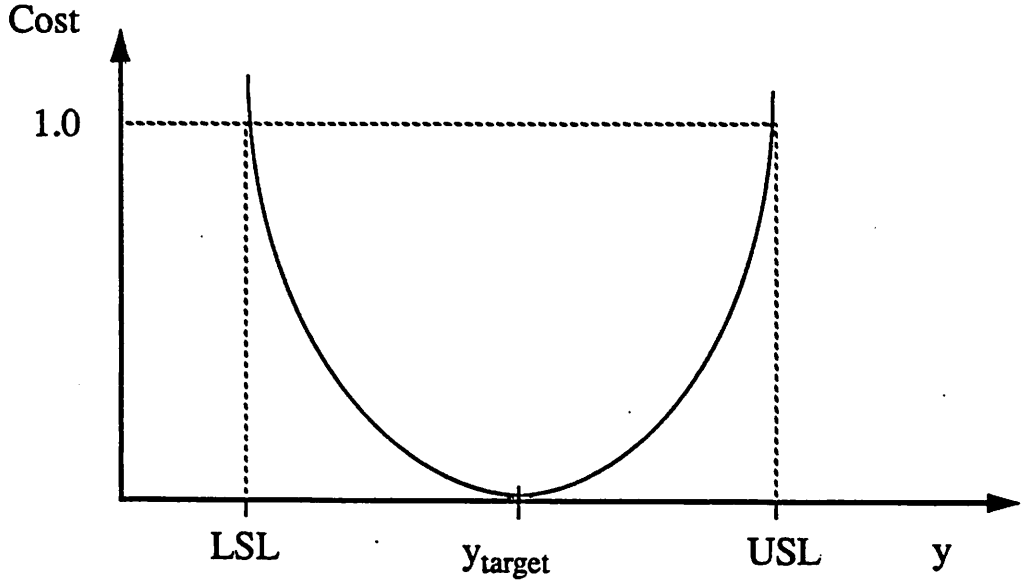


Figure 3. Quadratic Cost Function for a Single Output

This is done to maintain consistency across all equipment and to simplify the performance analysis of any step. In this way, the human operator or the CIM software can immediately infer the performance of the equipment simply by examining the magnitude of the cost, without the need for scaling. A quadratic, as opposed to linear, cost function reflects the idea that unit changes in the output affect the cost differently, depending on the distance to the target. Deviations in outputs which are close to their target affect the cost only to a limited degree because the outputs are already well within their specifications and small changes have negligible effeet on the process capability. Conversely, deviations in outputs which are far from their target affect the cost to a large degree because these outputs are limiting the process capability. Any change in these outputs should have a noticeable effect on the cost.

For multiple outputs whose specifications are independently defined, the total cost is the sum of the costs of each output:

$$\text{Cost} = \sum_{i=1}^{n} k_i (y_i - y_i^{\text{target}})^2 \qquad (3)$$

where $n$ is the number of outputs. In this context, 'independence' implies that the acceptable range for each output is not a function of the values of other outputs. These so called 'box constraints' specify that each output must be between an upper and lower specification limit. As with one dimension, each $k_i$ is chosen so that if an output is equal to its specifications, the contribution to the cost will be 1. In two dimensions, the resulting equal-cost lines of Equation 3 appear in figure 4.

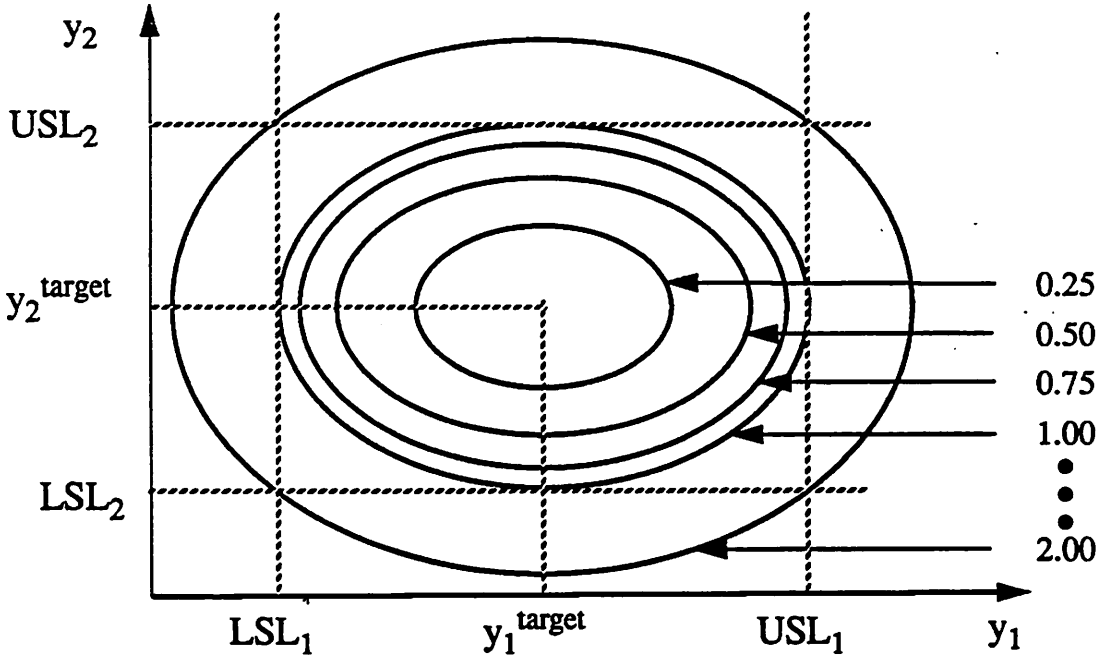


Figure 4. Lines of Equal Cost for the Quadratic Function

For a given output, the distance between its upper and lower specification limit is directly related to the importance of the output. The narrower the specifications in relation to the routine variation of this output, the more important the output.

The multidimensional quadratic cost function has one disadvantage. In general, given the cost, it is not possible to determine if all specifications have been met. If the cost is less than one, then all outputs have met their specifications. If the cost is greater than the number of dimensions, then at least one of the outputs did not meet its specifications. However, if the cost is greater than one but less than the number of dimensions, then it is impossible to determine if all specifications have been met by simply referring to the cost. The solution is to confirm the specifications directly or to use a different cost function as described in the next section.

### 2.4.2 Alternative Cost Function

An alternative function can be used to remove the uncertainty evident in the quadratic cost function:

$$\text{Cost} = \max\left(k_i\left[y_i - y_i^{\text{target}}\right]^2\right), \quad i = 1 \text{ to } n \tag{4}$$

where $n$ is the total number of outputs. The equal-cost lines for two dimensions are shown in figure 5. From the figure, it is obvious that the cost is less than one if and only if all specifications are met. However, this cost function introduces a more serious problem, since it focuses exclusively on the output which is farthest from its target and ignores the other outputs. Therefore, while bringing the "worst" output closer to its target, the other outputs can drift away from their targets unnoticed, with no effect on the cost until one of them becomes the worst output. This will drastically decrease the process capability, making the above formulation of the maximum cost function unsuitable for control purposes. Thus, the quadratic cost function is still preferable for most control situations.

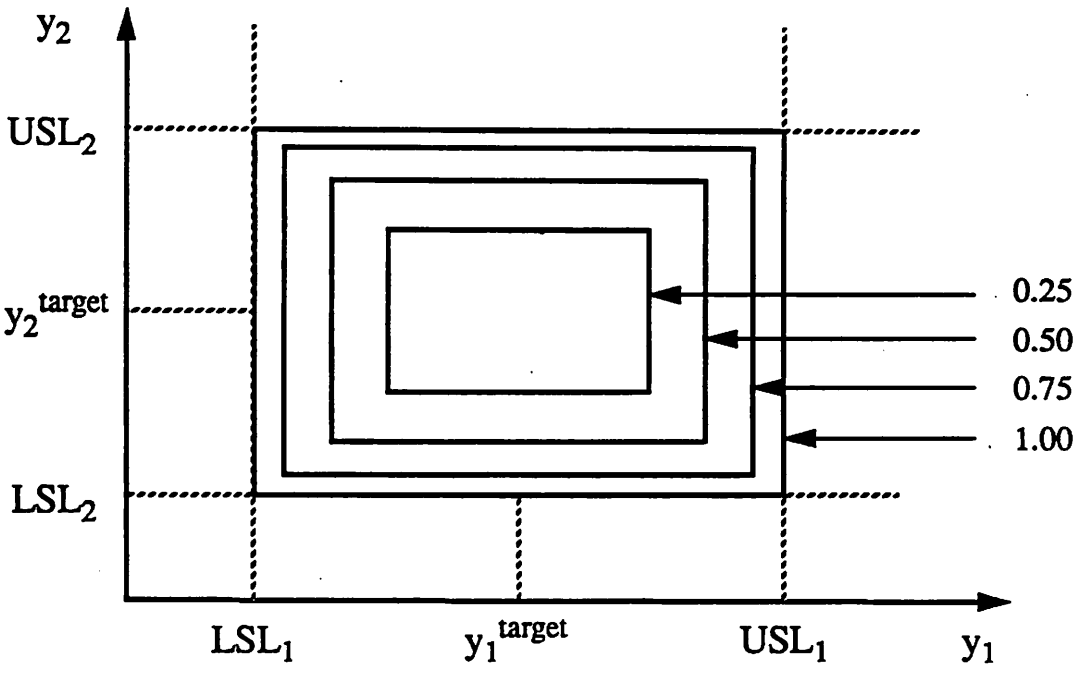


Figure 5. Lines of Equal Cost for the Maximum Function

### 2.4.3 Further Requirements of the Cost Function

The preceding discussion on cost functions has assumed that the specifications for the outputs are independently set. This assumption is usually true for the final outputs of a manufacturing sequence, but not necessarily true for the outputs of the intermediate steps. Figure 6 shows how dependent specifications form an oblique angle with the output axes. In these situations, interaction terms must be included in the cost function. The quadratic cost function in Equation 3 now takes the more general form:

$$\text{Cost} = \sum_{i=1}^{n}\sum_{j=i}^{n} k_{ij}(y_i - y_i^{\text{target}})(y_j - y_j^{\text{target}}) \qquad (5)$$

Equation 5 supports pairs of parallel linear specifications of the form,

$$b_1 < c_1y_1 + c_2y_2 + \ldots + c_ny_n < b_2 \qquad (6)$$

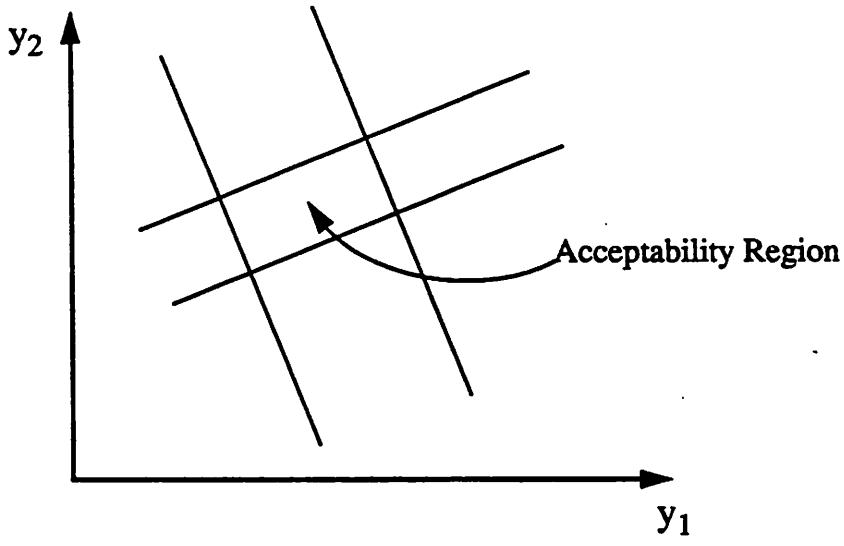


Figure 6. Pairs of Orthogonal Dependent Specifications

where **y** is the output vector and **b** and **c** are constants[1]. Graphically, the cost function is very similar to figure 4; the only difference is that the ellipses are now rotated through a certain angle.

The $k_{ij}$ coefficients are chosen by considering the cost associated with each pair of parallel specifications. If the output vector just meets the specifications, the cost associated with the pair is 1. If the output vector is midway between the pair of specifications, the cost associated with the pair is 0. For the total cost function, the cost from each pair of parallel specifications is combined to yield the total cost.

The remainder of this work uses pairs of parallel linear specifications which are orthogonal to each other, but not necessarily orthogonal to the output axes. As shown in figure 7, the resulting acceptability region is an orthogonal box which is used to approximate the true acceptability region.

Of course, in general, the true acceptability region can take any arbitrary shape. Though it would be possible to approximate the acceptability region with specifications

1. In this report, lower case bold letters designate a column vector.

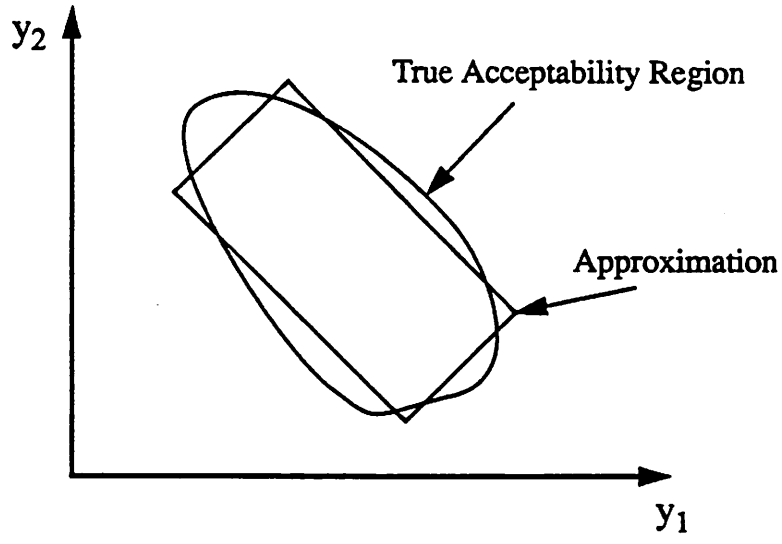


Figure 7. Approximation of True Acceptability Region

that are not parallel or not orthogonal, the derivation of cost functions as described in Chapter 5 would not longer be valid. Fortunately, simulations have shown that the box approximation will normally suffice. However, if the output of a processing step is a highly non-linear function of the input, then the box may be a poor approximation. The solution to this problem is discussed in Chapter 6.

This chapter has overviewed some of the important issues in sequential optimization. The next chapter will describe an implementation of a sequential optimizer which relies on many of the ideas presented so far.

# Chapter 3 Evolutionary Operation

## 3.1 Introduction

Evolutionary operation (EVOP) is a popular process control technique that is useful in optimizing equipment performance during production runs. Factorial experiments centered around the current operating point are constructed and from the results, the operating point may be adjusted if a favorable effect on the output is likely. When running the experiment, only small deviations may be introduced to the inputs in order for the process capability to remain acceptable. However, if the deviations are made too small, then the effects of the input variables will be invisible due to the routine variation of the process.

EVOP requires a single performance measure to be minimized or maximized. The quadratic cost function described in Chapter 2 is a natural choice. EVOP attempts to position the operating point at its optimal value even for noisy, dynamic environments. Unlike traditional off-line experimental designs, EVOP is applied on a sequential run-by-run basis during normal production runs. This chapter describes such an EVOP scheme as it has been implemented for the photoresist spin coat and bake station in the Berkeley Microfabrication Laboratory.

## 3.2 Methodology

### 3.2.1 Design of Experiment

All EVOP approaches use the common idea of a structured factorial experiment, but substantial flexibility still exists in the design of the experiment and the actions taken as a result of the experiment. When performing EVOP, a decision must be made regarding the magnitude of the deviations introduced to the inputs. If the ideal operating point is far from the current operating point, then large deviations are needed to shift the inputs as

quickly as possible. This is especially true if the current operating point is at a relatively insensitive location of the response surface or if appreciable noise exists, since small input deviations will have negligible effect. At the other extreme, if the current operating point is at its optimal location and the process is sensitive to the inputs, then the deviations introduced must be small in order for an acceptable capability to be maintained.

Although full factorial experiments are conceptually simpler, fractional factorial designs are encouraged especially if the number of inputs is larger than 3 or 4. By using fractional factorials, the important information is often deduced using much fewer runs than comparable full factorial designs. However, high order fractional designs run the risk of excessive confounding of effects [5] which may lead to incorrect conclusions. As a minimum, the resolution of the design must be at least III so that first order effects are not confounded with each other. It is important to realize that resolution III designs are still not immune to first order effects confounding with second order effects. If second order effects are considered to be significant, based on operator experience or theoretical foundations, then higher resolution designs should be utilized.

### 3.2.2 Estimation of Effects

Figure 8 shows a $2^2$ full factorial design which would be used for a system with only two inputs. After the equipment is run at each of the five locations, a *cycle* is said to have been completed, and a decision is made whether or not to change the current operating point. To make this decision, the *effects* of the inputs are calculated. A first order effect for an input variable is defined to be the average change in the output when going from the input's low value to its high value. For example, referring to figure 8, the effect of input 1 would be calculated as:

$$\text{Eff}_{x_1} = \frac{y_3+y_4}{2} - \frac{y_2+y_5}{2} \qquad (7)$$

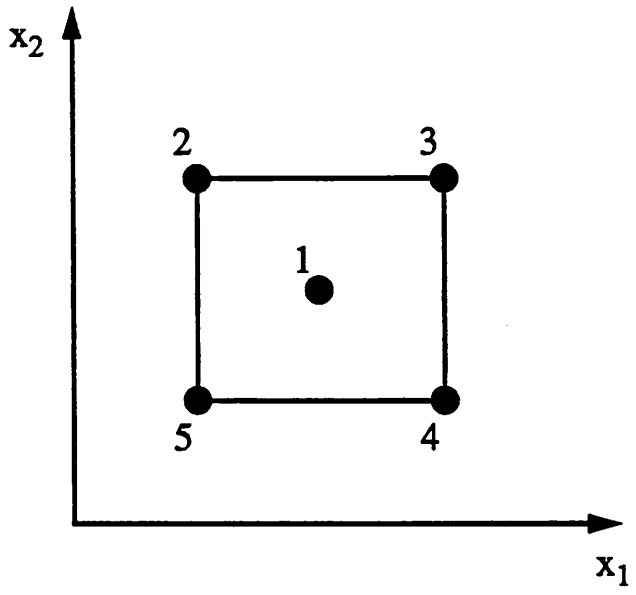


Figure 8. $2^2$ Factorial Design with Center Point

With $p$ runs per cycle, excluding the center point, the standard error of the estimation of the first order effects after $q$ cycles at a particular factorial location is

$$\sigma_{effect} = \frac{2\sigma_{exp}}{\sqrt{pq}} \tag{8}$$

Assuming that the replication noise of a process is normally distributed with standard deviation $\sigma_{exp}$, the 95% confidence interval for an effect is $+/-2\sigma_{effect}$. If the estimate of an effect lies outside of this interval, then the effect is considered to be statistically significant.

The change in mean effect, $CIM_{eff}$, is defined to be the difference in the average response at the factorial locations of the experiment minus the average response at the center point. For the 2-input case shown in figure 8,

$$CIMeff = \frac{1}{5}(\bar{y}_2 + \bar{y}_3 + \bar{y}_4 + \bar{y}_5 - 4\bar{y}_1) \tag{9}$$

The standard deviation of the CIM effect is

$$\sigma_{CIMeff} = \sqrt{\frac{p}{(p+1)q}}\sigma_{exp} \tag{10}$$

The CIM effect is used to determine if a minimum or maximum has been reached in the response surface. A significantly positive CIM effect indicates that a minimum may have been reached. A significantly negative CIM effect indicates that a maximum may have been reached. These conclusions are only valid if no first order effects are significant.

The numbers assigned to the points in figure 8 are not related to the order in which the experiment is actually run. Regardless of the order chosen, it is not possible to keep time effects unconfounded with both first order effects for any given cycle. It is possible to devise an ordering which will block time effects over many cycles, but for simplicity, a random experimental order is used for each cycle.

### 3.2.3 Shifting the Experiment

If a first order effect is significant, then the position of the factorial is moved for the next cycle. The location of the new factorial is dependent upon which effects are significant, whether the effects are positive or negative, and whether the output is being minimized or maximized. For example, if an effect is significantly positive and the output is being minimized, then the input will be decreased for the next cycle. Each first order effect is examined independently. Note that interaction effects need not be calculated, since the correct direction to move can be determined from first order effects alone. The only exception occurs when the current operating point is at a 'saddle point' of the response surface where a first order effect is positive on one side of the factorial experiment and negative on the other. Under these rare circumstances, using the information gained from calculating the interaction effects may incrementally improve the performance of EVOP.

After any given cycle, the center point of the next cycle will be shifted positively, negatively or not at all, for each input. Thus, for $k$ inputs, there are $3^k$ possible locations for the center of the next cycle as shown in figure 9. (Locations 1 through 5 are the settings used for the previous cycle.) It is quite possible that no effects will be significant, resulting in a cycle repeated at the same location.

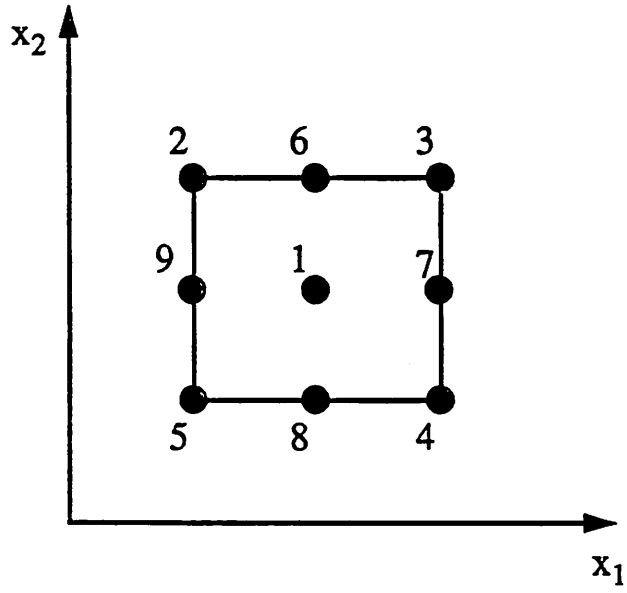


Figure 9. Possible Locations for a Center Point for the Next Cycle

Some implementations of EVOP unnecessarily restrict the possible new positions of the center point to be one of the locations of the previous experiment: positions 1 through 5. Doing this results in a reduced number of possible new locations, but actually makes the move decisions more complicated. For example, if the output is to be minimized, the effect of $x_1$ is negative and the effect of $x_2$ is negligible, we are motivated to increase $x_1$, implying that positions 3 and 4 are candidates for the next center point. We must make the choice between 3 and 4 arbitrarily, or wait for the effect $x_2$ or the interaction effect to become significant in order to make an intelligent decision. However, waiting for other effects to become significant will slow the response time of EVOP appreciably.

### 3.2.4 Estimation of Experimental Error

Testing the significance of the effects requires an estimate of the experimental error. This can be acquired by using points which are replicated during a repeated cycle. Any time a cycle is repeated, a new estimate of the experimental error is obtained by taking the difference, $d_i$, of the last output average at a particular location and the new value.

$$d_i = \overline{y}_i - y_{i+1} \tag{11}$$

It can be shown that the experimental error is related to the standard deviation of the differences by the following equation, assuming that the process has not shifted in the middle of a cycle.

$$\sigma_{exp} = \sigma_{diff}\sqrt{\frac{q-1}{q}} \tag{12}$$

$q$ is the number of cycles run, $\sigma_{diff}$ is the standard deviation of the differences, and $\sigma_{exp}$ is the experimental error. An estimate of $\sigma_{diff}$ can be determined from the differences as

$$\hat{\sigma}_{diff} = \frac{\sqrt{\dfrac{\sum_{i=1}^{m} d_i^2 - \dfrac{\left(\sum_{i=1}^{m} d_i\right)^2}{q}}{q-1}}}{c_4} \tag{13}$$

where $m$ is the number of runs per cycle and $c_4$ is defined below.

$$c_4 = \sqrt{\frac{2}{q-1}}\frac{\Gamma\left(\frac{q}{2}\right)}{\Gamma\left(\frac{q-1}{2}\right)} \tag{14}$$

$\Gamma(k) = (k-1)!$, for integer k. $\Gamma(k) = (k-1)\,\Gamma(k-1)$. $\Gamma(1/2) = \sqrt{\pi}$

When repeated cycles are needed using a fractional arrangement, each repetition should use the same fraction. Different fractions could be used for each cycle, at the expense of having to make new estimates of $\sigma_{exp}$.

It is possible to use the range of the differences to estimate the experimental error. This is computationally simpler, but the computing power available in modern-day computers makes this difference unnoticeable. Further, as the number of runs per cycle increases, the relative efficiency of the range estimator diminishes, so it should only be used in problems with small dimensionality.

Once an estimate of the experimental error is obtained from a repeated cycle, it is combined with the previous estimate using an exponentially weighted moving average to form the new estimate to be used. This weighted average is used to reflect that recent estimates are more important than older ones.

### 3.2.5 Restrictions to Shifting the Experiment

In general, if any effect becomes significant after only one cycle at a particular location, then the experiment is moved for the next cycle and no updated estimate of the noise is obtained. We call this situation a *quick move*. There are two instances when a quick move is not allowed. First, since the experimental error is estimated only through replication, at least two cycles will always be run at the starting location in order for the initial noise estimate to be derived.

The second exception applies when several consecutive quick moves have been performed. If this were to occur, then the estimate of the error would not be updated and instead would be based only on relatively old data. Thus, a cycle is inserted to update the estimate of the error. In a dynamic system where the noise level is constantly changing, it

is crucial to maintain an accurate estimate of the noise at all times. But even in static systems where the noise level of the outputs is constant, the sensitivity of the quadratic cost function to noise is variable, since it will be dependent on the distance of the outputs to their targets. The addition of noise to outputs which are close to their targets will have a small impact on the cost since the cost function is in its flat region. The addition of noise to outputs which are away from their targets will have a large impact on the cost since the cost function is in a steep region. Thus, even for static systems, updated estimates of the error are necessary.

To ensure that a current estimate of the error is used, a limit has been set on the maximum number of consecutive quick moves allowed. This heuristic was developed to guard against the situation where $\hat{\sigma}_{exp}$ is underestimating the true $\sigma_{exp}$. Without the heuristic, numerous consecutive quick moves may occur without having a chance to update $\hat{\sigma}_{exp}$, causing moves to be made as a result of noise only. Having an overestimate of $\sigma_{exp}$ is not a problem since it will be more difficult to shift the experiment and the resulting repeated cycles will help produce updated estimates of $\sigma_{exp}$.

## 3.3 Implementation

The algorithms described above have been implemented using C++ and have been combined with equipment models developed by the Berkeley Computer Aided Manufacturing group.

Various parameters of the experiment are specified by the user. These include:

- number of inputs and outputs
- degree of fractionation
- range of the inputs
- the starting center point for the inputs
- specifications for the outputs
- step size

The step size specifies the distance between the center point and the factorial points along each input direction, as a fraction of the range of each input.

The other parameters that may be modified are the maximum number of consecutive quick moves and the forgetting factor used in the exponentially weighted average calculation for the estimate of the noise.

Simulated optimization runs were completed on the Eaton photoresist spin and bake station, but any equipment can be simulated with trivial modification to the code. The simulations determine the number of runs required to find the optimum as a function of the step size and the type of the experimental design. Once the optimum has been found, the effects of continuously changing the recipe on the cost function have been analyzed. Generators for the fractional factorials have been taken from [5].

## 3.4 Simulated Optimization of the Spin-Coat Procedure

A few parameters were set before the simulation began. Specification limits were set to be 12190 - 12610 Å for the photoresist thickness and 35.5 - 44.5% for the photoresist peak reflectance. The peak reflectance is observed by scanning wavelengths in the neighborhood of the exposure wavelength. Refer to [9] for details.

Normally distributed errors with sigmas of 70 Å and 1.5% were added to the thickness and reflectance respectively. With the specifications set to be three standard errors away, the best attainable capability for each output would be approximately 1. The starting center point for the factorial experiment was 5200 rpm for spin speed (SPS), 30 seconds for spin time (SPT), 115 °C for bake temperature (BTE) and 90 seconds for bake time

(BTI). Further, the maximum number of consecutive quick moves was 5 and the forgetting factor was 0.5. The equipment models used were taken from [9] and are repeated here.

$$T = -13814 + \frac{2.54 \cdot 10^6}{\sqrt{SPS}} + \frac{1.95 \cdot 10^7}{BTE\sqrt{SPS}} - 3.78BTI - 0.28SPT - \frac{6.16 \cdot 10^7}{SPS} \quad (15)$$

$$R = 134.4 - 0.046SPS + 0.32SPT - 0.17BTE + 0.023BTI \quad (16)$$
$$-4.34 \cdot 10^{-5}\,(SPS \cdot SPT) + (5.19 \cdot 10^{-5})\,(SPS \cdot BTE) - (1.07 \cdot 10^{-3})\,(SPT \cdot BTE)$$
$$-(4.11 \cdot 10^{-4})\,(SPT \cdot BTI) + (5.15 \cdot 10^{-6})\,(SPS)^2$$

The result of using full and half factorials with various step sizes is shown in Table 1.

**TABLE 1.**

| Type of Factorial | Step Size | Runs to reach optimum | Average cost after optimum is reached | Thickness Cpk after optimum is reached | Reflectance Cpk after optimum is reached |
|---|---|---|---|---|---|
| Full | 0.01 | 1200 | 0.261 | 0.907 | 0.981 |
| | 0.02 | 400 | 0.341 | 0.735 | 0.939 |
| | 0.05 | 100 | 0.765 | 0.433 | 0.894 |
| | 0.10 | 50 | 2.787 | 0.211 | 0.698 |
| Half | 0.01 | 900 | 0.298 | 0.871 | 0.996 |
| | 0.02 | 250 | 0.355 | 0.717 | 0.976 |
| | 0.05 | 75 | 0.850 | 0.409 | 0.894 |
| | 0.10 | 30 | 2.710 | 0.214 | 0.711 |

As the step size increases, fewer runs are required to find the optimum, but after the optimum is reached, the average cost is larger and the process capabilities degrade. For the given values of the specifications and noise levels, running the system continuously at its optimum without introducing the deviations required for the factorial experiment results in an average cost of 0.239. Note that when the step size is small, the average cost is only slightly above the cost obtained when no deviations are introduced. Thus, such a small

step size would have a minimal impact on the process capability. Using a large step size around the optimum degrades the process capability substantially.

However, a small step size requires a very large number of runs to reach the optimum. If the current operating point is far from the optimum, such as when the original optimizations are being done at start-up or if the equipment response has shifted, large step sizes are desirable. In a system without any noise, doubling the step size will cut the number of runs required to find the optimum in half. In a system with noise, doubling the step size will cut the runs required by more than half.

Figure 10 shows the cost as a function of run number for a variety of step sizes. The simulations were done using half fractional designs.

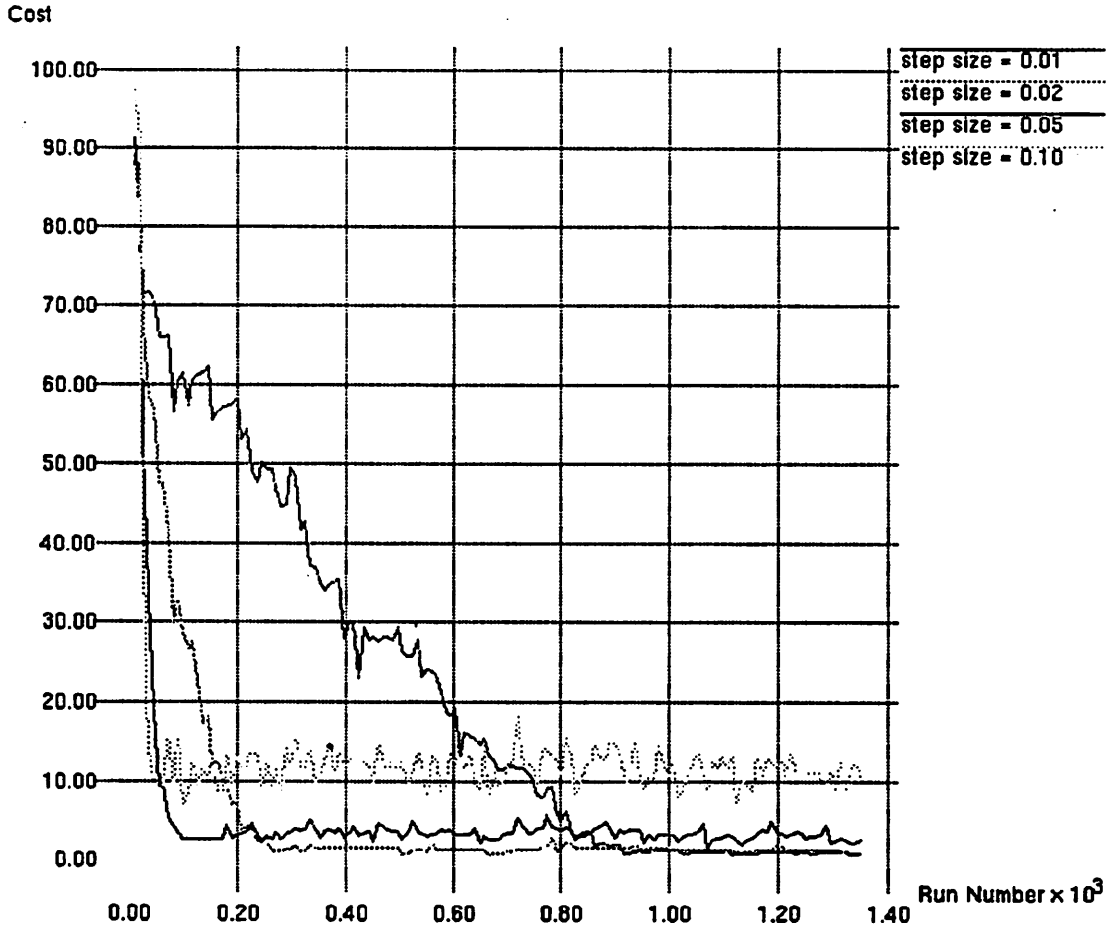


Figure 10. Effect of Step Size on Performance

The average costs around the optimum for a full factorial and its half fraction are virtually the same, but the number of runs to reach the optimum are significantly different, as expected. If the half fraction was equally effective as the full factorial in determining the direction to move, then both approaches would take the same number of cycles. This is true when the step size is large since the effects are much larger than the noise level. But as the step size decreases, the half fraction takes more cycles than the full factorial, but still fewer total runs. In the limiting case where the step size is made arbitrarily small, we would expect the half fraction to take the same number of runs as the full factorial to find the optimum.

## 3.5 Conclusion

An EVOP software package has been written and applied to the Eaton photoresist spin and bake station. Simulations have been run to verify operation of the software and also to examine the impact of fractional factorials and step size.

Clearly, the ideal scenario would be to have a variable step size depending on the current conditions: a large step size when movement is required, a small step size when the optimum has been found. It is important to note that even after the optimum is found, the recipe does not become fixed. Instead, EVOP continues so that adaptations to shifts in the equipment can be made.

Several enhancements could be made to decrease the response time of EVOP to changes in the equipment. For example, if large effects are calculated, then the position of the center point for the next factorial could be shifted by an amount larger than the step size used within a single factorial. In addition, if several moves are currently being made in a certain direction, then the step size in that direction could be increased as well.

When an extra cycle is inserted to update $\hat{\sigma}_{exp}$, the response is delayed. If extra cycles need to be inserted frequently and the factorial is large, one alternative would be to simply

repeat the runs at the center point to update the estimate instead of repeating the entire factorial. This would reduce the total number of runs required.

Even with the proposed enhancements, the response time of EVOP may not be small enough to meet the demands of a given manufacturing environment. Instead, a more rapid sequential optimization approach may be necessary. Such an approach is presented in the next chapter.

# Chapter 4 Rapid Sequential Optimization

## 4.1 Introduction

One of the drawbacks of EVOP is that information from runs at previous cycles is ignored. Consequently, each cycle of factorial experiments consumes several runs before a decision is made to move the operating point. To maintain an optimal yield in a dynamic processing environment, the response time to shifts in the equipment must be less than EVOP can deliver. A different approach must be used.

## 4.2 Ultramax

Ultramax is a commercial software package that has been developed to perform rapid sequential optimization [1]. It can be applied during process development, but its intent is to optimize a process in an ongoing production environment. The goal of Ultramax is to find the optimum input settings in a minimum number of runs and also to respond to equipment shifts after the optimum has been found. Like EVOP, it is based on the concept of introducing deviations to the inputs while continuing to manufacture acceptable products. Ultramax can be configured to vary the inputs to a small degree, which would be useful when the optimum operating point has been found. Alternatively, when shifts in the equipment have occurred, large deviations can be introduced so that the new optimum can be found faster.

EVOP suffers from the fact that many runs are necessary to construct a factorial experiment. A shift in the operating point is only made after a factorial experiment has been completed. The advantage of Ultramax is that since factorial experiments are not used, the current operating point can be adjusted after every run. This results in a shorter response time.

All combinations of inputs and outputs from previous runs are used to derive equipment models. The regression uses an exponentially weighted approach so that recent data can be considered more important. By constructing equipment models, Ultramax is able to locate regions of the input space that are likely to yield favorable outputs. As new data is generated, the models are continuously updated to reflect current operating conditions. Figure 11 shows the incorporation of Ultramax into a control system.

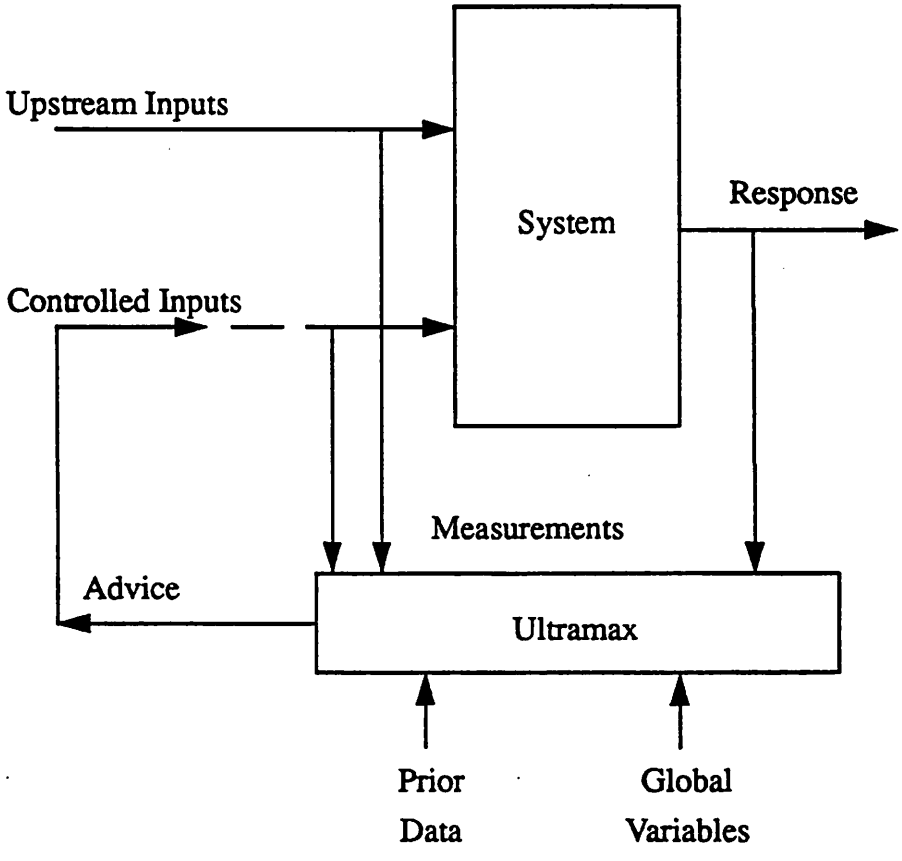


Figure 11. Using Ultramax as a Controller

Before monitoring the system, any prior data that may be available is used to construct an initial model of the system. The global variables specify parameters of the cost function such as target values and specification limits. Feed-forward control is achieved by first examining the upstream inputs and adjusting the advice for the controllable inputs accordingly. After each run, the outputs are measured and the models are updated. Even if

the controllable inputs are set to values different than what was advised, the run data is still used to update the equipment models. Some details about the algorithm can be obtained from the Ultramax User's Guide [1].

## 4.3 Simulated Performance and Conclusions

Simulations of the Eaton photoresist spin and bake station have been run to compare the performance of Ultramax to EVOP. The specifications, starting point and the amount of noise that were chosen to evaluate EVOP in Chapter 3 are also used here.

Table 2 shows the results. Similar to EVOP, as the aggressiveness of the search

**TABLE 2.**

| Type of Search | Runs to reach optimum | Average cost after optimum is reached | Thickness Cpk after optimum is reached | Reflectance Cpk after optimum is reached |
|---|---|---|---|---|
| Conservative | 15 | 0.348 | 0.723 | 0.951 |
| Moderate | 12 | 0.373 | 0.718 | 0.933 |
| Aggressive | 9 | 0.682 | 0.461 | 0.871 |

increases, the number of runs required to reach the optimum decreases, but the steady state cost increases and the process capabilities decrease. Compared to EVOP, Ultramax requires very few runs to find the optimum.

In general, the number of runs required to reach the optimum is a function of the linearity of the response surface. Since EVOP calculates linear effects, a linear reponse surface is ideal. With Ultramax, fewer terms in the model need to be derived for a linear response surface. Thus, as the linearity of a system increases, the efficiency of EVOP and Ultramax will increase.

The simulations performed above assume that the specifications for thickness and reflectance are known. Typically, this information is subjective and based primarily on

operator experience. Even if the given specifications are accurate at a particular point in time, they need to be altered when shifts in equipment or changes in the final output specifications occur. A methodology for determining accurate specifications for all steps in a processing sequence is presented in the next chapter.

# Chapter 5 Dynamic Specifications for a Multi-Step Process

## 5.1 Introduction

Modern semiconductor manufacturing consists of numerous individual steps. In order to maximize the yield of the final outputs, it is important for the intermediate outputs of all equipment in the sequence to be within acceptable regions. Determining the optimal target range for each equipment is complicated by noise in the system and affected by shifts in the equipment response and changes in the final output specifications. Further, it is important to find a consistent set of specifications among all processing steps since specifications at a given location in the sequence will directly influence the specifications of all prior steps. As shown in Figure 12, the outputs of the intermediate steps are targeted for acceptability regions which will ensure that the following steps can meet their specifications. A further discussion of this topic can be found in [6].

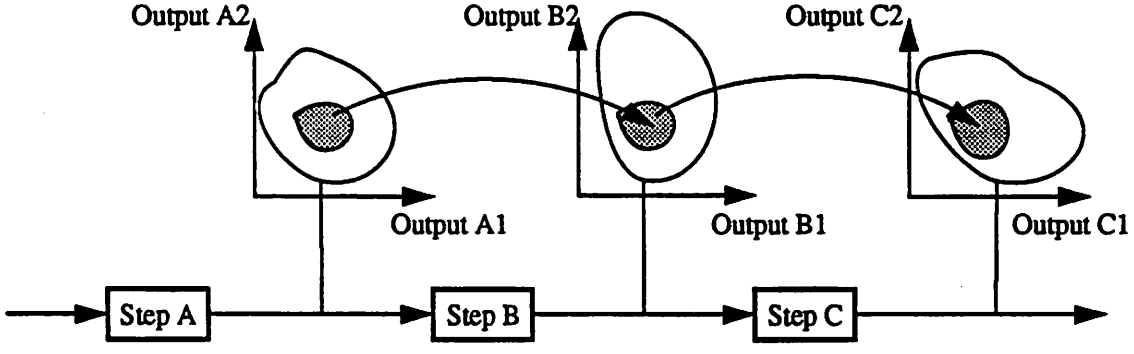


Figure 12. Alignment of Intermediate Outputs

In the figure, the shaded areas represent the acceptable region of the outputs of each step, whereas the outer region is the entire output space reachable by the step. This outer region can be obtained by varying a step's inputs over their ranges and determining the output. The arrows between the acceptability regions represent the intent of the intermediate specifications. If the specifications are met at a given location in the

processing sequence, then a careful selection of inputs for the next step might ensure that the following specifications will also be met. However, noise and shifts in the equipment response may make this selection difficult or impossible.

Feed-forward control can help to correct process variability in previous equipment, but only to a limited extent. In other words, the application of feed-forward control widens the acceptability region of the previous step. If however, the upstream inputs deviate substantially from their ideal values, no amount of feed-forward control can compensate for previous mistakes. The goal of dynamic specifications is to align the intermediate outputs to a range where feed-forward control can still be used to bring the final outputs to their targets. Specifications are propagated upstream through the processing sequence so that compensation for a change in equipment response or final output specifications can be made as early as possible.

Some approaches [6] use fixed acceptability regions that are derived at system start-up. However, as equipment response changes over time or if the final output specifications change, these fixed acceptability regions are no longer valid. They must be updated to reflect the current status of the entire manufacturing line in order for the process capability to be maximized. Next we describe a novel approach for the implementation of dynamic specifications.

## 5.2 An Approach for the Dynamic Modification of Process Specifications

Figure 13 shows the integration of dynamic specifications into a control system. In the figure, products flow from left to right. A local controller with feed-forward and feedback capabilities is responsible for determining the values of the controllable inputs at each step. By utilizing a current model of the step based on recent data, the controller is able to perform feedback control. Feed-forward control is accomplished by first examining the important parameters of the incoming wafer and then adjusting the recipe to compensate

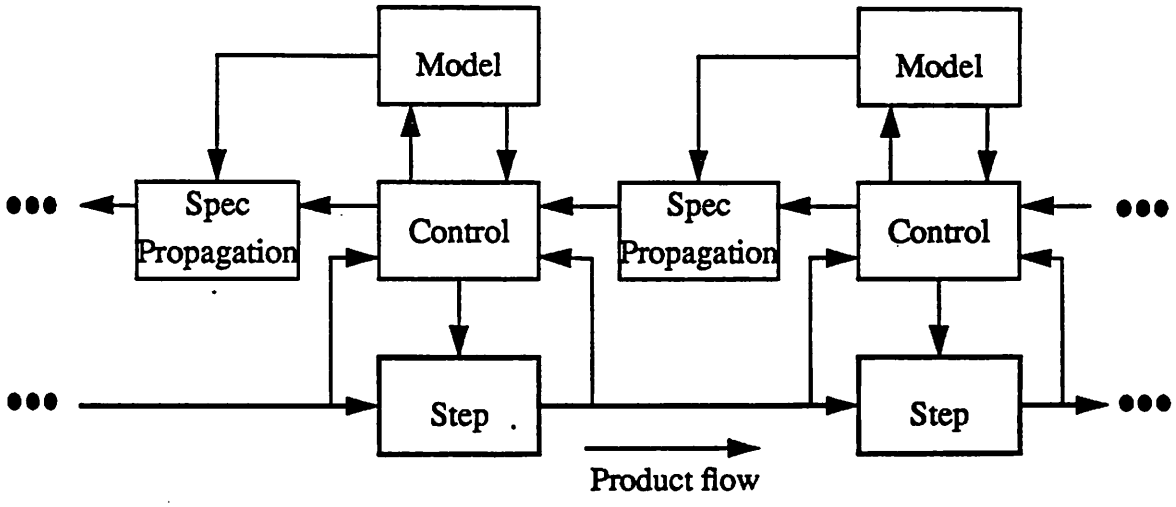


Figure 13. The Dynamic Specification System Within a Supervisory Controller

for any variation. As each wafer goes by, the output is examined and the model is updated. Once a change is detected, specifications for all upstream steps are derived by starting at the final outputs and propagating the specifications upstream through all steps in the processing sequence. In order to transform the specifications across a piece of equipment, its current model is used. At system start-up, the models are generated from historical data. After the specifications have been updated, a cost function is derived and will be used by the controller for future runs.

The intermediate specifications must be re-evaluated whenever a change in the system is detected, as indicated by a control alarm [3]. Note that only the specifications of steps preceding a change need to be updated. Alternatively, specifications can be re-evaluated for every equipment after each run, even if no changes have been detected. If no changes to the system have occurred, then the intermediate specifications may change only slightly due to noise. For simplicity, the latter approach is currently being used in this work.

## 5.3 Propagating Specifications

The propagation of specifications across a step has been divided into four stages: Monte Carlo simulation, selection of acceptable points, principal component analysis of the acceptable points and cost function derivation. The key issues regarding each of the stages is described next.

### 5.3.1 Monte Carlo Simulation

The first stage in transforming specifications from the outputs to the upstream inputs of a step is to perform a Monte Carlo simulation using its current model. The goal is to determine the acceptable region of the upstream variables. Random combinations of upstream inputs and controllable inputs are generated uniformly across the input space.

The range of the upstream input space is determined by operator experience. It should be made large enough to virtually guarantee that any wafers having parameters outside of the range are unsatisfactory. If the range were made large enough to only enclose the current operating region, then it might miss parts of the acceptability region when performance shifts occur. The only disadvantage to making the upstream range unnecessarily large is that more points must be simulated to ensure adequate resolution during the subsequent step of mapping the acceptability region.

A uniform distribution is used to generate an even coverage of the input space, since the goal is to determine the acceptability of all locations within the space. Note that the Monte Carlo simulation is not intended to reflect the typical distribution of the parameters of the incoming wafers. If this were the goal, a normal distribution would be more appropriate. The accuracy of a Monte Carlo simulation is independent of the number of dimensions and is therefore preferred to an exhaustive grid search especially for equipment with many inputs [7]. Empirical equipment models result in a tremendous speed-up improvement over using detailed physically-based simulators.

For each combination of inputs generated, the cost associated with the resulting outputs is determined. During the simulation, the values of the controllable inputs are chosen randomly and independently of the upstream inputs. Further, no noise is added to the outputs generated by the model, even if accurate estimates of the noise are known.

### 5.3.2 Selection of Acceptable Input Points

Once the input combinations have been generated and their cost determined, the points are divided into two categories: those that are expected to meet the specifications and those that are not. This division is performed based on evaluating the cost. Using the maximum cost function introduced in Chapter 2, if the cost is less than one, then the outputs have met their specifications and therefore the inputs that generated the outputs are considered acceptable. All combinations of upstream inputs whose cost is less than one are used to define the acceptability region for the previous step.

Replication noise will cause the observed cost to be different from the expected cost; consequently, the identification of an input combination as being acceptable does not ensure that the specifications will actually be met when the process is run with these inputs.

It is conceivable that for the simulated range of inputs, no points generate a cost numerically less than one. A number of explanations are possible. First, a piece of equipment could be malfunctioning, causing the inferred models to change dramatically. In this case, an alarm should have been generated causing the equipment to be serviced. Second, the given output specifications may not be attainable. Finally, the defined input space may not be wide enough to handle the typical variations in equipment response. The solution for the last two cases is to search over a wider input range until acceptable input points are found. If no points with a cost less than one can ever be found, then the specifications are not attainable and an alarm is generated.

### 5.3.3 Principal Component Analysis

Principal component analysis (PCA) is a technique used in various applications to reduce the dimensionality of a problem. Typically, a highly correlated multi-dimensional data set is decomposed into fewer independent variables which are orthogonal in the multi-dimensional space. PCA begins by finding the direction which explains the most variation in the data set. Then, the direction explaining the most variation in the space orthogonal to the first direction is determined. This process repeats until all of the variation is explained by $n$ mutually orthogonal directions in the $n$-dimensional space. Any of the principal component directions that explain negligible variance are eliminated. The original data set is then transformed into variables which are the distances along the remaining principal component directions. The directions with the highest variability are deemed to be the most important.

PCA has been applied to the propagation of specifications, but the intent is much different than that described above. Instead of attempting to reduce dimensionality, it is used to form orthogonal linear specifications that approximate a collection of acceptable upstream input points. Figure 14 shows the information derived from the principal component analysis. Given the acceptable inputs points, PCA determines the target value for all upstream inputs and the direction of the principal components. The center lines are easily derived and then the spread of the distances from the acceptable points to each center line is determined. Note that the typical number of acceptable points used in the principal component analysis is much larger than shown in the figure.

A direction with a large spread indicates that there are acceptable points which are distant from the center line perpendicular to the direction. This means that the process is relatively insensitive to changes along that direction and therefore the specifications can be wide. Conversely, a direction with a relatively small spread indicates that acceptable points only exist near to the center line perpendicular to the direction. Consequently, the

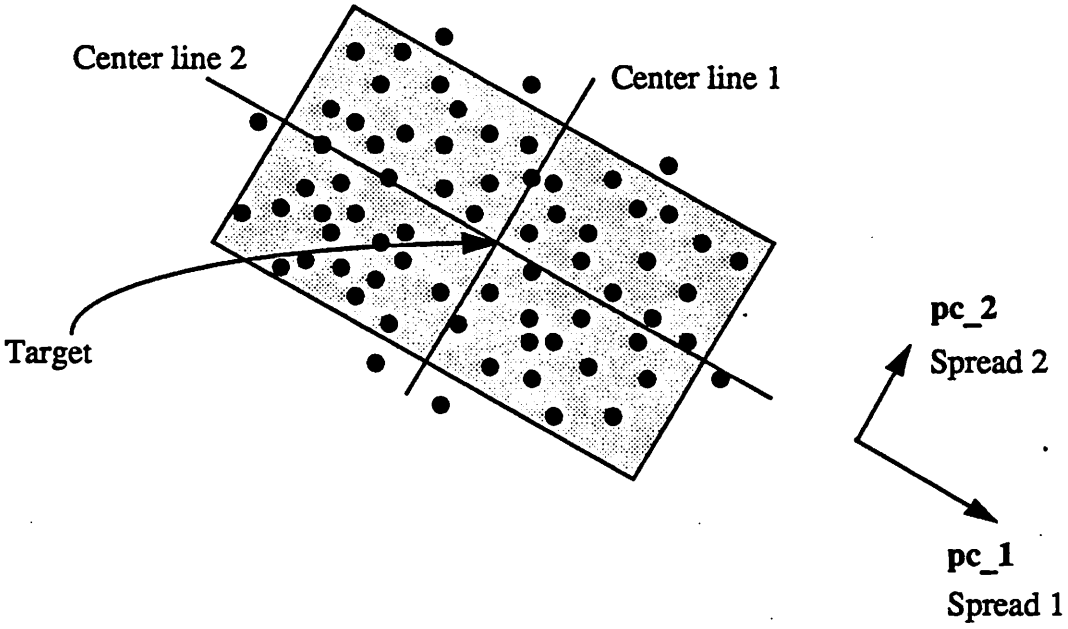


Figure 14. Information Derived from PCA of Acceptable Points

process capability will degrade rapidly as the upstream inputs move away from the target in that direction. Thus, narrow specifications must be set.

The variance of the distances from the acceptable points to the center line, $\sigma^2_{acc}$, is used as an intermediate step to determine the spread along each direction. Assuming that the distance from the acceptable points to the center line is a random number that is uniformly distributed, the width of the specifications can easily be determined from the variance of this distance. The variance of a uniform distribution of width $w$ is $w^2/12$. The distance from the center line to the specifications for that direction is then:

$$w/2 = \sqrt{3\sigma^2_{acc}} \qquad (17)$$

If the acceptability region is rectangular, then the assumption about the distances being uniformly distributed will be valid. Simulations have shown that even when the acceptability region is not rectangular, accurate specifications are still generated.

It is important to normalize the upstream inputs before performing the principal component analysis so that each input dimension carries equal weight. Using the following formula, the inputs are scaled to be within the range -1 to 1. Without normalization, the analysis will be skewed by the units chosen for the inputs. The variance evident in those inputs that vary over a large range will dominate whereas the variance evident in those inputs that vary over a small range will be insignificant.

$$x_i' = \frac{x_i - x_i^{center}}{(x_i^{high} - x_i^{low})/2} \tag{18}$$

### 5.3.4 Cost Function Derivation

The cost associated with each pair of parallel specifications is calculated independently. Let $\mathbf{d}^T = (d_1, d_2, d_3..., d_n)$ be a principal component direction. Let $\mathbf{t}^T = (t_1, t_2, t_3..., t_n)$ be the target value in the acceptability region. Let $\mathbf{x}^T = (x_1, x_2, x_3..., x_n)$ represent a vector that belongs to the upstream input space. Then, the equation of the center line that passes through $\mathbf{t}^T$ and is orthogonal to $\mathbf{d}^T$, is

$$\mathbf{d} \bullet (\mathbf{x} - \mathbf{t}) = 0 \tag{19}$$

The distance from this center line to an arbitrary point $\mathbf{u}^T = (u_1, u_2, u_3..., u_n)$ in the upstream input space is $\mathbf{d} \bullet (\mathbf{u} - \mathbf{t})$, assuming that $|\mathbf{d}| = 1$. The square of this distance is divided by the square of the distance from the center line to the specifications, $w/2$. The resulting polynomial is of the same form as the quadratic cost function introduced in Chapter 2, equation 5. For each of the principal component directions, the $k_{ij}$ coefficients are computed and combined to yield the coefficients for the total cost function. For the maximum cost function, equation 4, the $k_{ij}$ coefficients for each direction must be kept separate so that the cost in each direction can be computed. For the quadratic cost

function, $k_{ij}$ coefficients corresponding to the same terms of the cost function can be added together.

## 5.4 Alternative Method to Propagate Specifications

An entirely different approach to propagating specifications would be to attack the problem analytically. The outputs of each step can be represented as a function of the inputs. With given output specifications, the input specifications can be determined through a simple variable substitution. Repeating the variable substitutions will enable us to define the specifications for all steps. However, with many steps, the resulting expressions for the specifications can become unnecessarily complex.

## 5.5 Summary

Figure 15 overviews the stages required to propagate specifications. First, a Monte Carlo simulation is run using randomly generated points in the input range of a step. For each input point, the outputs are calculated using the model of the step and the costs are determined by the output specifications. Second, for each output point which meets the specifications, the corresponding input point is chosen to be acceptable. Third, the specifications of the input points are derived by performing a principal component analysis. Finally, a cost function is generated from the specifications obtained in stage 3.

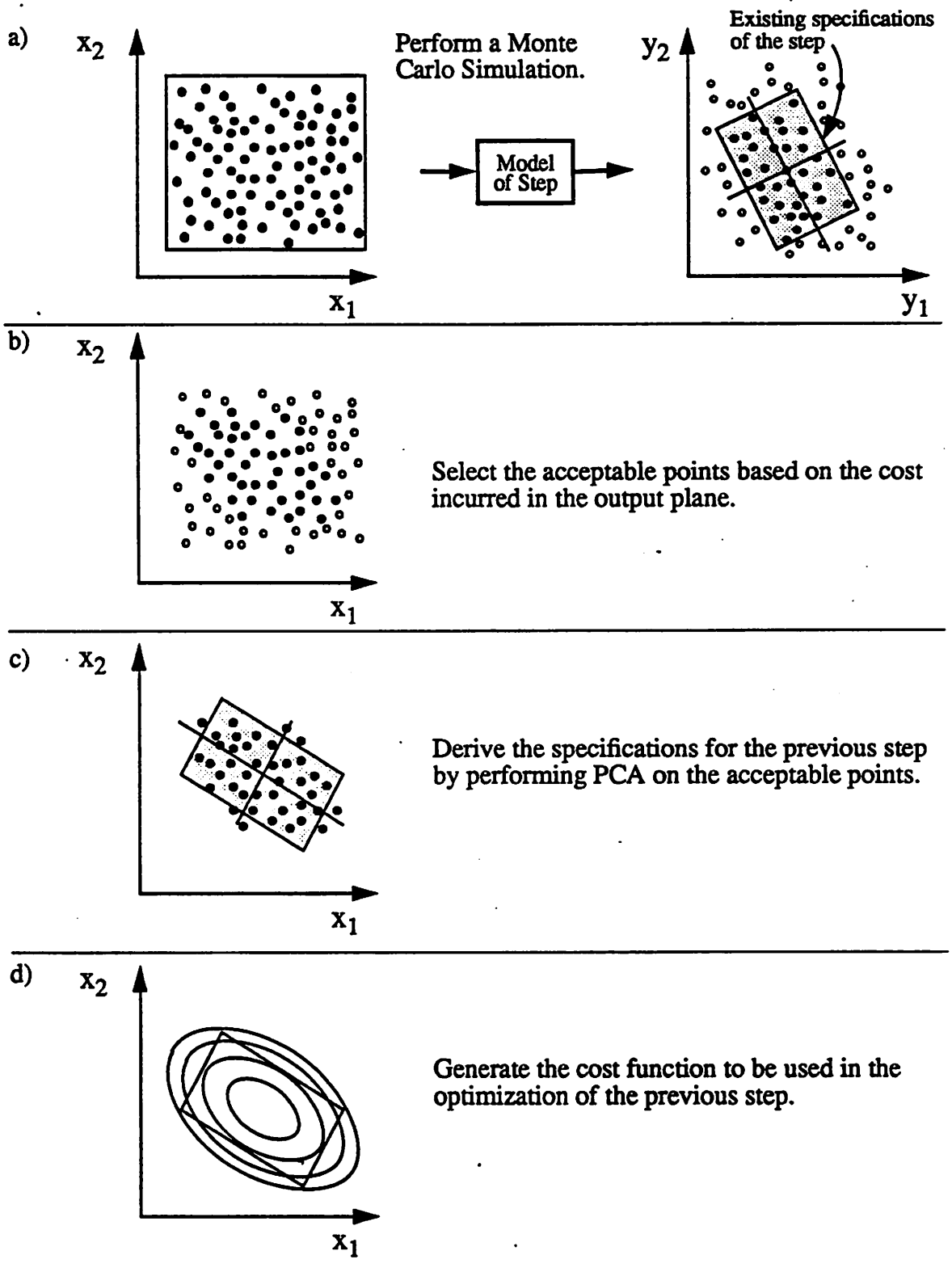


Figure 15. The Four Stages of Propagating Specifications

# Chapter 6 System Simulation and Conclusions

## 6.1 Implementation

The software for propagating dynamic specifications was implemented in C++ and was merged with equipment models developed by the Berkeley Computer Aided Manufacturing group. Portions of the code have been dedicated to the simulation of the photolithographic workcell, a three step process. The sequence consists of the photoresist spin coat and bake station introduced in Chapter 3, as well as a stepper and a developer. The inputs and outputs used for control purposes for each step are shown below. The goal is to achieve a critical dimension (CD) on the developed photoresist layer within a certain range.

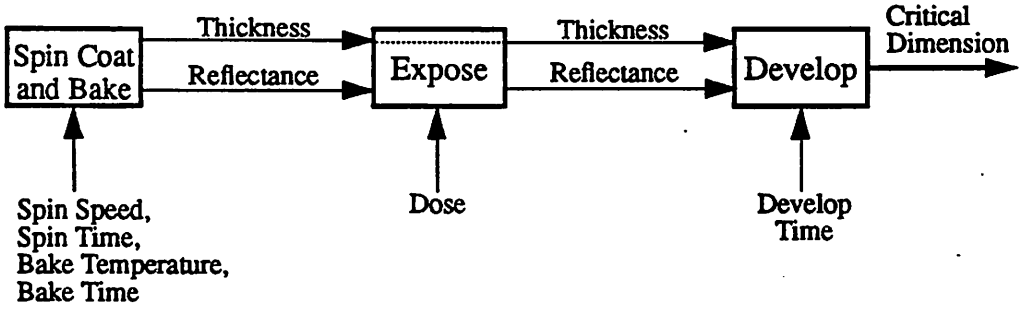


Figure 16. The Photolithographic Workcell

Ultramax has been used as a controller for each of the three steps, although any controller that can perform feedback and feed-forward control would be appropriate. The principal component analysis has been implemented using the Berkeley Interactive Statistical System [8]. Details of the model generation for the steps in the photolithographic workcell can be obtained from [9].

## 6.2 Determination of Intermediate Specifications

The target value for the CD was 2.66 μm, as determined in [3] and the specifications were chosen to be +/- 0.02 μm around that value. Given this information, the acceptability region for the thickness and reflectance outputs of the stepper were determined. The Monte Carlo simulation searched over the rectangular region defined by the boundaries, 11000 Å< Thickness < 13000 Å, 70% < Reflectance < 90%, while the develop time was held constant at 60 seconds. The model used for the critical dimension, as determined in [9], is:

$$\begin{aligned} CD &= 8.86 - 0.076Rin - 0.00075Tin + 0.00000375 \cdot Tin \cdot Devtime \\ &+ 8.95 \cdot 10^{-6}(Rin \cdot Tin) - (4.27 \cdot 10^{-8})(Tin \cdot Rin \cdot Devtime) \end{aligned} \quad (20)$$

The acceptable points from the simulation and the derived specifications are shown in figure 17. This figure demonstrates the acceptable input thickness and reflectance to the developer assuming that feed-forward control is not used and instead the develop time is held fixed at 60 seconds. Note that this figure shows the normalized values of thickness and reflectance. The resulting quadratic cost equation using the normalized values of thickness and reflectance is shown below.

$$Cost = 1.669(T + 0.07352)^2 + 1.440(T + 0.07352)(R - 0.09692) + 1.682(R - 0.09692)^2 \quad (21)$$

## 6.3 Allowing for Feed-forward Control During Propagation of Specifications

If a feed-forward controller is in place, then the controllable inputs of a step should be varied while performing the Monte Carlo simulation. Given values of the upstream inputs may or may not be considered acceptable, depending on the values of the controllable inputs generated for the simulation. Figure 18 shows the effect of varying the controllable inputs in the Monte Carlo simulation. For all three plots, the search space is 10500 to

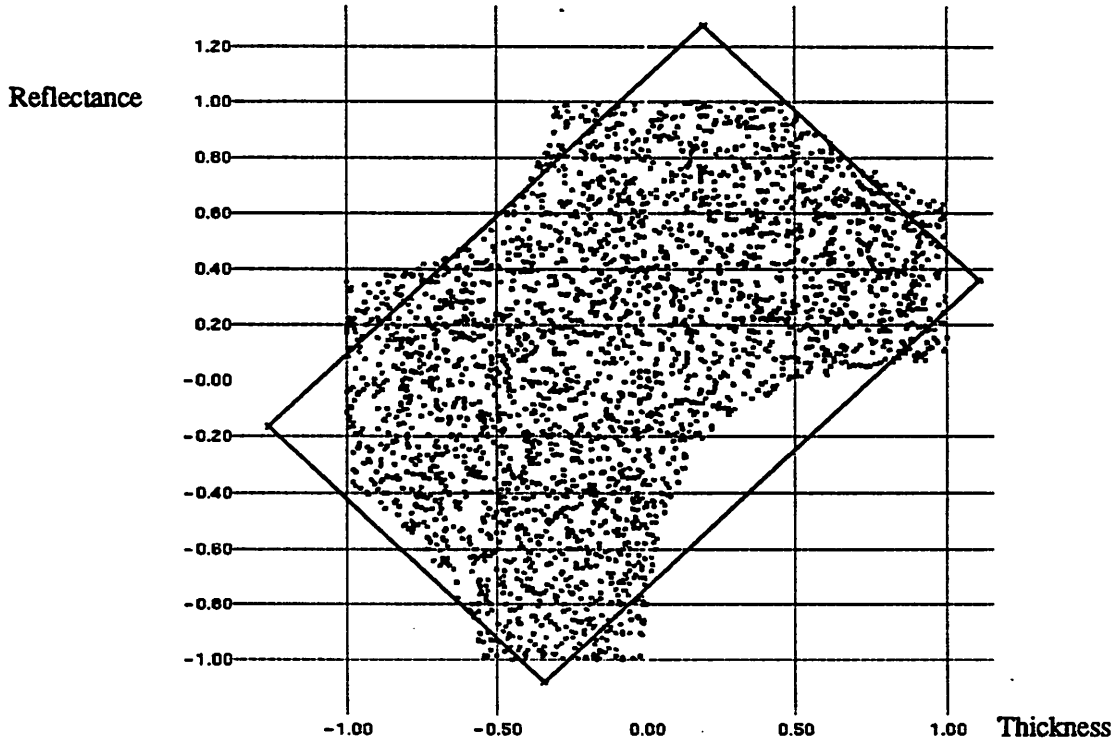


Figure 17. Acceptability Region for the Outputs of the Stepper

13500 Å for the thickness and 65% to 95% for the reflectance. The specifications are 2.66 +/- 0.02 μm. The plots in Figure 18 show that the acceptable points extend out to the edge of the search space, indicating that the search space perhaps could have been widened to include more acceptable points. As the develop time is allowed to vary over a wider range, the acceptability region expands and the density of the points varies within the acceptability region. For combinations of thickness and reflectance that have a high density of acceptable points, many values of develop time over the covered range in the Monte Carlo simulation result in a satisfactory output. If all simulated points within a certain region are acceptable, then any develop time is adequate. Conversely, for combinations of thickness and reflectance that have a low density, few choices of develop time result in a satisfactory output, and so the develop time must be chosen carefully when the actual process is run. Note that the proposed specifications tend toward the dense regions of the acceptable points.

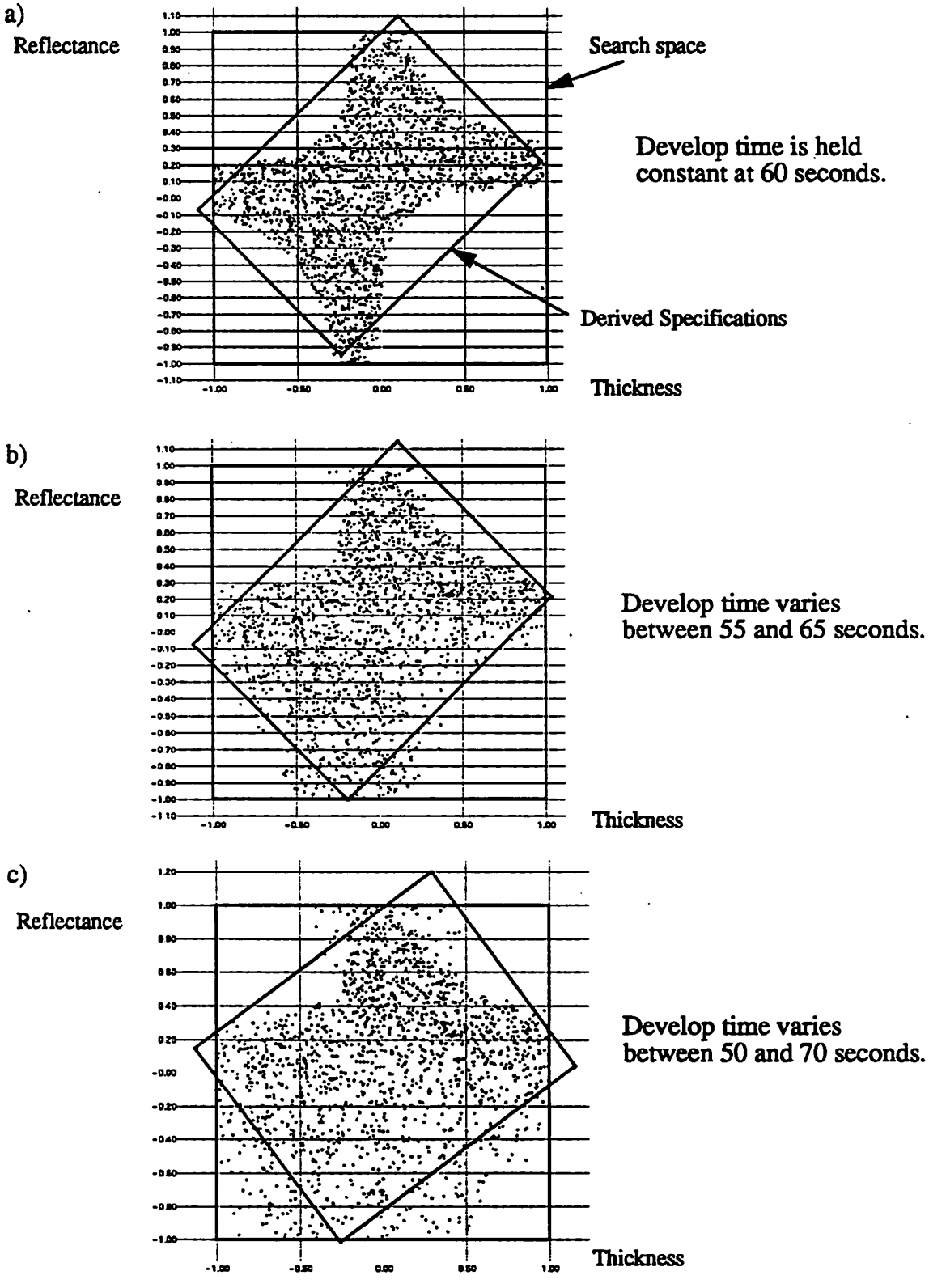


Figure 18. The Effect of Feed-forward Adjustments of the Controllable Input

The relative densities of acceptable points can be explained by examining the response surface plots. Figure 19 shows the response surface of the CD as a function of thickness and reflectance. The three develop times chosen, 50 seconds, 60 seconds and 70 seconds, span the range that was used in the Monte Carlo simulation above. Using critical dimension specifications of 2.64 μm to 2.68 μm, the surfaces reveal that for small reflectance values, the acceptability region is highly dependent upon the develop time. As a result, the density of the points in the lower half of figure 18c is low. Conversely, the well-defined acceptability region in the top portion of figure 18c is consistent with the fact that the top portion of response surfaces are similar for any develop time.

## 6.4 The Effect of Equipment Response or Final Specification Changes

As the equipment response or final output specifications change, the intermediate acceptability regions need to be updated. Figure 20 shows the result of changing the specifications from 2.66 +/- 0.02 μm to 2.64 +/- 0.02 μm. All other experimental

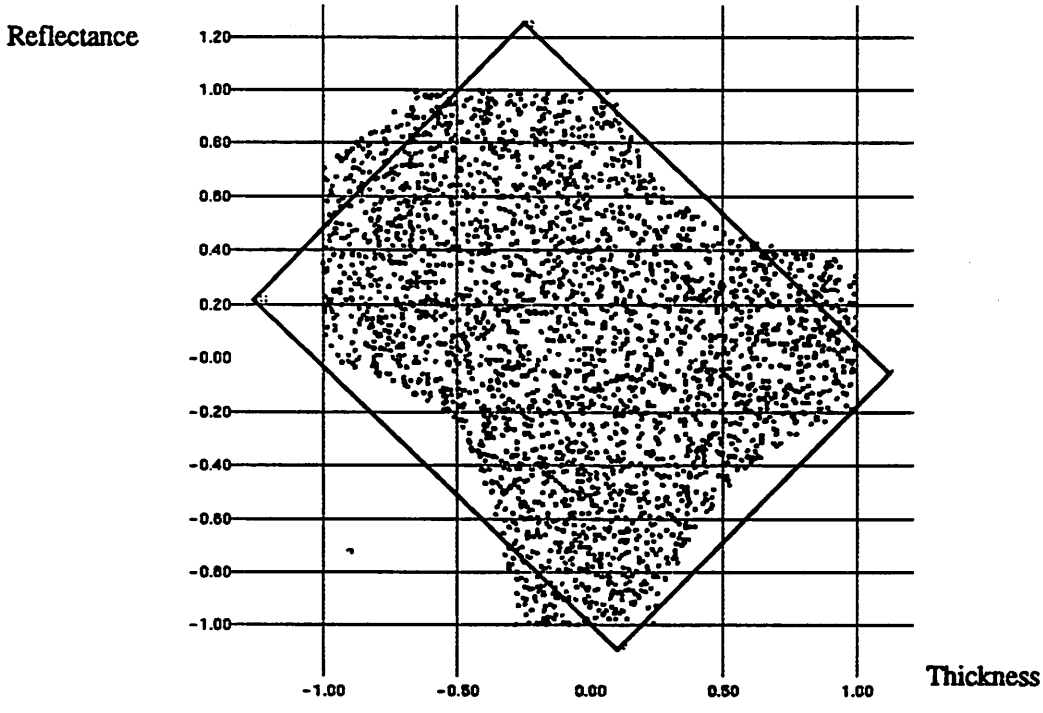


Figure 20. New Acceptability Region After a Change in Specifications

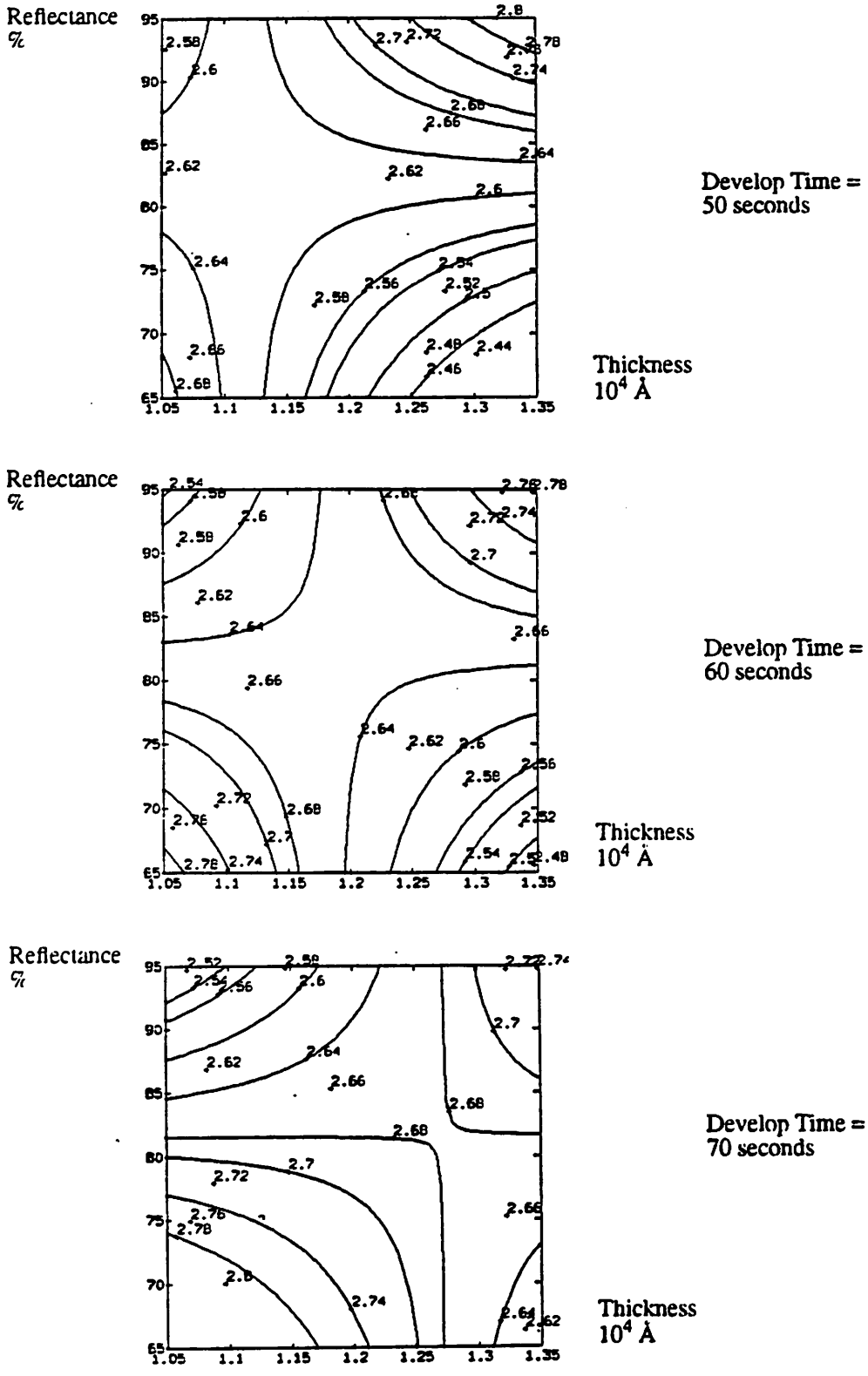


Figure 19. Response Surface Plots For Various Development Times

conditions are the same as those used to generate Figure 17. Comparing these two figures reveals that the acceptability regions overlap, as expected, since the specifications overlap. The difference is that the combinations of thickness and reflectance which generate a CD between 2.66 and 2.68 μm have been removed and combinations which generate a CD between 2.62 and 2.64 μm have been added. As shown in the figures, changing the output specifications results in the intermediate specifications being compressed in one direction and expanded in the other. This can be explained by examining the response surface plot.

Changing the model of the developer will also change the acceptability region for the incoming thickness and reflectance. It can easily be shown that increasing the constant term of the CD model by $x$ will have exactly the same effect as lowering the output specifications by $x$. The effect of varying other terms in this polynomial model is best determined by simulation.

## 6.5 Highly Non-Linear Acceptability Regions

It is possible that non-linearities in the equipment model may result in an acceptability region that is not adequately approximated by orthogonal specifications. Figure 21 shows an example where the acceptability region is not only non-linear, but it is discontinuous. This example was generated by using output specifications of 2.66 +/- 0.01 μm. The derived orthogonal input specifications are clearly not adequate since they enclose a significant portion of the input space that does not generate acceptable outputs. Blindly applying the derived specifications to the outputs of the previous step will lead to substantial errors.

It is felt that discontinuous acceptability regions are uncommon; nonetheless, they cannot be ignored. Two separate issues need to be addressed. The first is how to propagate the specifications farther upstream if adequate specifications for the current step have not been generated. The solution to this problem would be to run the Monte Carlo simulation

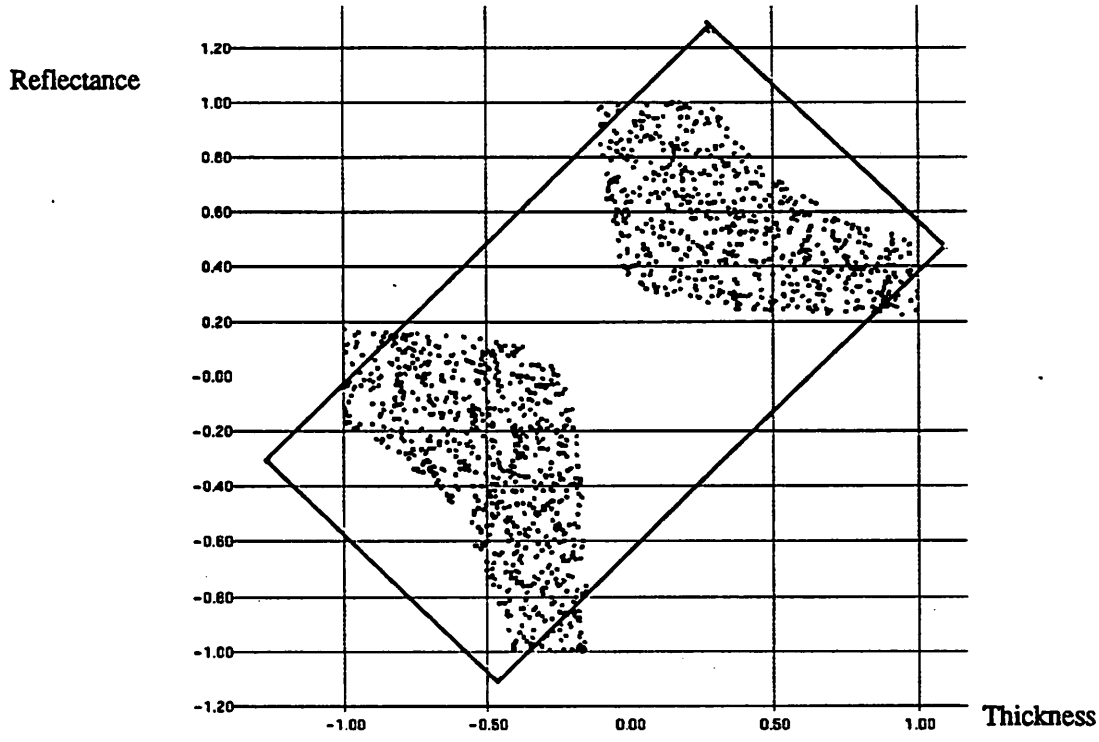


Figure 21. Discontinuous Acceptability Region

across more than one step, terminating with a step whose specifications are approximately orthogonal. Then, the cost to use for the original upstream input points in the simulation would be the cost generated after the step whose specifications are orthogonal. For example, if the acceptability region for the upstream inputs of the developer are as shown in figure 21, the acceptability region for the upstream inputs of the stepper can be determined by generating inputs to the stepper and using the simulated cost after the developer. By doing this, the discontinuous acceptability region at the output of the stepper is bypassed. The only disadvantage to this approach is that the Monte Carlo simulation time will increase because more model evaluations need to be executed.

The second problem encountered when discontinuous acceptability regions arise at the output of a step is how to determine the cost function to be optimized by the controller for the step. Setting the target to be the centroid of the acceptable points will no longer suffice since such a target may not even lie within an acceptable region. In addition, the width of

the specifications cannot be determined by the spread of all acceptable points. It is important to identify when clusters of acceptable points exist and then to choose the 'best' cluster. Practical considerations may help to determine which cluster is the most appropriate. More research is necessary to automate this selection. However, once a cluster has been chosen, the cost function can be derived as before.

## 6.6 Conclusions and Future Work

A novel specification propagation methodology has been demonstrated in this chapter. Given the output specifications for a step, the input acceptability region and input specifications can be derived. It has been shown that feed-forward controllers can help to enlarge the acceptability region of the inputs. Further, varying the output specifications can dramatically change the intermediate specifications.

As mentioned in the previous section, the methodology needs to be made more robust to highly non-linear acceptability regions. The recognition of discontinuous clusters of acceptable points is also important.

The next step in this research would be to incorporate the dynamic specifications into a control system and evaluate its performance. The controller could be an implementation of EVOP or Ultramax, as described in Chapters 3 or 4, or any other controller that uses targets and specification ranges. In addition, the propagation of specifications could be incorporated into the BCAM framework where simulations can be performed using equipment models or the actual equipment in the microfabrication laboratory.

# References


[1] Ultramax Corporation, "Ultramax User's Guide", 1990.

[2] Bart Bombay, Costas Spanos, "Application of Adaptive Equipment Models to a Photolithographic Process", SPIE, September, 1992.

[3] Sovarong Leang, Costas Spanos, "Application of Feed-Forward and Feedback Control to a Photolithography Sequence", to appear in ASMC, October, 1992.

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

[5] George E. P. Box, William G. Hunter, J. Stuart Hunter, "Statistics for Experimenters", John Wiley & Sons, 1978

[6] S. Director, W. Maly and A. Strojwas, "VLSI Design for Manufacturing: Yield Enhancement", Kluwer Academic Publishers, 1989.

[7] Robert Spence, Randeep Singh Soin, "Tolerance Design of Electronic Circuits", Addison-Wesley Publishing Company, 1988.

[8] D. Mark Abrahams, Fran Rizzardi, "BLSS: The Berkeley Interactive Statistical System", W. W. Norton and Company, 1988.

[9] Sovarong Leang, MS Thesis, UCB/ERL M92/70, University of California, Berkeley, 1992.