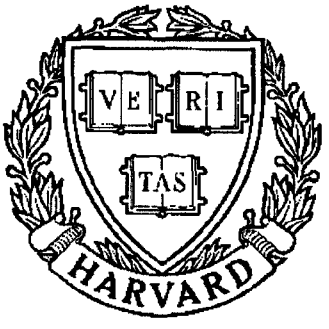
THESIS REPORT
Master's Degree

S Y S T E M S
R E S E A R C H
C E N T E R



Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), the University of Maryland, Harvard University, and Industry

A Neural Network Based Approach for Surveillance and Diagnosis of Statistical Parameters in the IC Manufacturing Process

by W. Zhang
Advisor: L. Milor

M.S. 92-15

Abstract

Title of Thesis: A Neural Network Based Approach for Surveillance and Diagnosis of Statistical Parameters in the IC Manufacturing Process

Name of degree candidate: Wei Zhang

Degree and year: Master of Science, 1992

Thesis directed by: Dr. Linda Milor
Department of Electrical Engineering

Despite advances in integrated circuits (IC) equipment and fabrication techniques, there still exist random fluctuations or statistical disturbances in any IC manufacturing facility, which can adversely affect the production yield. Actually devices and circuits are being designed with increasingly tighter parameter and performance margins. As a result, chip performance becomes even more sensitive to the statistical variations, and this may result in low production yield. Based a statistical process simulator, a methodology of tracking and diagnosing statistical variations of a real manufacturing process in a bid to implement real time statistical quality control of IC manufacturing process is presented in this thesis.

The main contributions of this thesis include the following. A neural network based approach for IC process diagnosis is proposed and has been realized. This approach needs a very short time in diagnosing significant variations of an IC process, hence is practical to be used in real-time monitoring and diagnosing of the process disturbances. Another contributive feature of this approach is that process diagnosis is a high dimension problem, and in our approach all variables are handled simultaneously, instead of eliminating of some variables that may have small but important contributions as in previous approaches. Other contributions include an algorithm to evaluate the fault observability and disturbance diagnosability. In addition, thresholding and coding methods are developed for pattern generation of the neural networks. A special sampling distribution is employed for simulation of samples, in conjunction with latin hypercube sampling techniques. Finally the approach is applied to a general example to show its efficiency with some experimental results.

A Neural Network Based Approach for Surveillance and Diagnosis of Statistical Parameters in the IC Manufacturing Process

by

Wei Zhang

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
1992

Advisory Committee:

Dr. Linda Milor, Chairman/Advisor
Dr. R. Newcomb
Dr. A. Tits

Acknowledgements

I would like to take this opportunity to express my deep appreciation to my advisor, Dr. Linda Milor. Her enlightening guidance, kind help, insightful comments, and encouragement are instrumental for the completion of this work. I want to thank Dr. J. Dayhoff and Mr. A. Teolis for the helpful discussions on neural networks. My appreciation also goes to the members of my advisory committee, Professor R. Newcomb and Professor A. Tits for their valuable help and comments. I would like to acknowledge the financial support from the Institute of Systems Research, University of Maryland at College Park. And finally I want to thank my wife Chien for her years of encouragement and supports.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

During the last decade, the feature sizes of VLSI devices have been scaled down significantly, and they are still becoming smaller. Despite advances in integrated circuits (IC) equipment and fabrication techniques, however, there still exist random fluctuations or statistical disturbances in any IC manufacturing facility, which can adversely affect the production yield. And unfortunately, these statistical variations in the critical device parameters, such as the MOSFET channel length and width, and threshold voltage have not been scaled down in proportion to the feature sizes of devices. Actually devices and circuits are being designed with increasingly tighter parameter and performance margins. As a result, chip performance becomes even more sensitive to the statistical variations, and this may result in low production yield.

To achieve an acceptable yield, this random nature of the IC manufacturing line should be taken into consideration during the design procedure. With the variety of CAD tools available nowadays, an accurate model can be established

and fine turned to simulate a real manufacturing process [1][2][3]. The process yield can thus be predicted through analyzing the critical process parameters extracted from the simulation process. The extracted process information also makes it possible to implement process control and maintain uniform process condition and high yield [4][5].

In the literature substantial effort has been put on the establishment of statistical models and the optimization of yield [6][7][8]. Based on some of these models and results, a methodology of tracking and diagnosing statistical deviations of a real manufacturing process in a bid to implement real time statistical quality control of IC manufacturing process is presented in this thesis.

The main contributions of this thesis are listed as follows.

1. A neural network based approach for IC process diagnosis is proposed and has been realized.

2. This approach is practical in real-time monitoring and diagnosis of IC process disturbances

3. Process diagnosis is a high dimensional problem and all variables are handled simultaneously, rather than subdividing the problem, which may lead to inaccuracies.

4. An algorithm is devised to evaluate fault observability and disturbance diagnosability.

5. Thresholding and coding methods are developed for pattern generation of the neural networks.

6. A special sampling distribution is employed for simulation of samples, in conjunction with latin hypercube sampling

7. The approach is applied to a general example to show its efficiency.

1.2 Monitoring and Diagnosing of IC Process Variations

The IC manufacturing process involves a sequence of basic processing steps that are performed on sets of wafers called lots. Each wafer may contain tens to hundreds of chips. Due to the batch character of IC manufacturing, process faults and deviations that may arise in the various steps of an IC process result in a large volume of defective products before they are detected and rectified.

The faults and variations that cause depreciation of process yield come from the deviated process conditions and the unavoidable fluctuations inherent in IC manufacturing process. Although most process conditions, such as temperature, gas flux and pressure can be measured and controlled accurately, the fluctuations, such as the diffusivity of boron and arsenic, variance of implantation profile and surface state density, etc. are generally not directly measurable. Therefore, monitoring the fluctuations, detecting the significant deviations of them from their nominal values and keeping them under control play an important role in stabilizing and improving yields of integrated circuit manufacturing.

An IC fabrication line is characterized by a number of fabrication recipes being used to manufacture different kinds of IC's. In each fabrication recipe there are a large number of process parameters that to different extents affect the output performance and in turn the yield of the process. When some of these

parameters depart from their designated values, or nominal values, what follows may be the generation of some functionally defective devices or even a big drop in the yield. Process diagnosis and control are employed, in this circumstance, to identify those shifted parameters and bring them back to their nominals. Due to the quantities of the process parameters, however, it is infeasible and uneconomical to monitor all of them. A much more effective way is naturally to monitor those parameters that have a significant impact on the process yield.

As aforementioned, most process fluctuations or disturbances, which are the main source of the parameters to be monitored, are generally not directly measurable. Hence we have to rely on in-line measurements and electrical measurements of a fabrication line to determine the distributions of the process disturbances. On the other hand, process disturbances provide more useful information if the process is out-of-control. Therefore, we have built control charts for process disturbances, and consequently it has been necessary to map the statistics of a set of observed measurements to statistics of process disturbances. The main thrust of the work is therefore first to find the optimal choice of parameters to monitor with control charts, the main tool used in statistical quality control, which will be introduced in succeeding chapters. Secondly, because a set of measurements may be related to multiple process disturbances, in order to diagnose a fault we have implemented an algorithm that will identify out-of-control process disturbances after a set of measurements have been made.

The relation between process disturbances and observable measurements is modeled by a process simulator. The process simulator will map process disturbances to observable measurements. So what we actually need is the inverse of this map. This problem has previously been studied by Spanos [3], who fits the

map from disturbances to measurements by using polynomial approximation, and then uses nonlinear programming to find the disturbance statistics. The main drawback of the approach lies in the high dimensionality of the problem, since it is very difficult to build accurate polynomial models for a high dimensional problem.

Assuming normal statistics have been determined, our problem is to diagnose major changes in the disturbances. In this paper we solve this problem by fitting a neural network to the map from statistics of measurements to statistics of process disturbances. The motivation behind the idea of using neural networks is based on the extensive applications of neural networks and common belief that neural networks can handle high dimensional problems better than polynomial approximation techniques. Some simplified device models have been used to demonstrate the feasibility and efficiency of this method. Results of the experiments will be provided in chapter 7.

1.3 IC Process Control and Yield Optimization

The manufacturing of today's micron and submicron features in integrated circuits requires a tight control of the fabrication process in order to realize the objective of maximizing process yield. In volume production, this means that both the size of these features and its distribution must be well controlled and fall within specifications.

Because of the complexity of the IC fabrication process, identifying those parameters to monitor is a non-trivial task. Some specifically designed experi-

ments, such as a factorial experimental design [9], or the latin hypercube method [10] could be applied to obtain a reasonable choice according to the circuit being manufactured and the specifications it needs to satisfy. Once a set of parameters has been selected, it is possible to choose an optimal set of control limits for the control charts established for these parameters. The choice of control limits should depend on the sensitivity of the yield to a given disturbance, the cost of producing faulty chips, and the cost of halting processing due to a false alarm. Research done in this area is summerized by Montgomery [11].

When a sample in a control chart indicates that the process has gone out of control, an investigation will be initiated for the assignable causes of the problem. This can either be malfunctioning equipment or the conditions need to be adjusted. To find out if it is possible to adjust the conditions to bring the process back into control, the sensitivity of the yield to the conditions at a given (out of control) operating point can be computed. If adjusting the conditions can not bring the process into control, then the source of the problem is likely to be faulty equipment. The use of process disturbance control charts should hence make it easier not only to identify an out of control process, but also find the appropriate actions to correct the process.

We have briefly discussed the background and objective associated with monitoring and diagnosis of the IC process variations. In next chapter, a detailed description about the IC manufacturing process and their statistical characteristics will be provided, and then some previous work relevant to this topic will be summarized.

Chapter 2

Statistical Characterization Problem in Integrated Circuit Manufacturing Process

2.1 Introduction to IC Manufacturing Process

The manufacturing process of an integrated circuit (IC) consists of a sequence of steps that are carried out in a specific order. These steps generally include mechanical, optical, thermal and chemical operations performed on a silicon substrate, and finally convert the circuit design into a functional silicon integrated circuit chip.

The objective of the IC manufacturing process is to produce IC chips satisfying specific requirements of a design at as small a cost as possible. To achieve this goal, production is traditionally done in batch mode, including processing up to several hundred ICs together on a wafer, and a few dozen wafers in a lot. Hence, several thousand of the same ICs are manufactured together, simply for the purpose of reducing the cost per chip. At the same time, effort has to be

directed towards minimizing the departure of geometrical and electrical features of the processed devices from those specified during design stage. Toward this end, a high degree of control over the parameters of each processing step is required. Equally rigid requirements apply to the physical and chemical properties of materials used for IC fabrication, and also to the cleanliness of the production environment.

The sequence and manner in which individual IC process steps are carried out vary from one IC process to another, and they are crucially important to the outcome of the manufacturing process. Although different techniques make use of different equipment and materials, hence creating different devices, the IC processing steps basically fall into the following six categories [12][13]:

Photolithography: This is a technique used in IC fabrication to transfer a desired pattern onto the surface of a silicon wafer. As such, photolithography is a key step in the entire circuit integration process. Specifically, it is a photochemical process during which the layout is transferred through visible or ultraviolet exposure from a photomask to a photoresist layout, which has been deposited on the wafer. The image is subsequently developed quite similar to the image on a photographic negative using specific chemicals that affect exposed and non-exposed areas differently. The geometry of the regions in which the photochemical reaction in the photoresist takes place corresponds to the pattern on the mask. The accuracy of the pattern transfer from the mask to the wafer is determined by the resolution of the photolithographic process. The higher the photolithographic resolution, the finer the geometrical features that can be patterned onto the wafer.

Oxidation: The oxidation of silicon is necessary throughout the modern

integrated circuit fabrication process. Silicon dioxide has several uses: to serve as a mask against implant or diffusion of a dopant into silicon; to provide surface passivation; to isolate one device from another; to act as a component in MOS structures and to provide electrical isolation of multilevel metallization systems. Several techniques for forming oxide layers have been developed, such as thermal oxidation, wet oxidation, the vapor-phase technique and plasma oxidation. They are employed in different circumstances to generate silicon dioxide layers serving the different purposes mentioned above.

Layer Deposition: The layers of both conducting substances and insulating materials constitute an important part of any semiconductor device. In contrast to the deposition of the silicon dioxide layer by thermal oxidation, the deposition process does not involve a chemical reaction with the substrate. In deposition, all components of layer being grown are independent of the composition of the substrate (deposition of a thin solid layer in this manner does not cause consumption of the silicon substrate as is the case in thermal oxidation of silicon). The configuration of the deposited thin layer reflects the topography of the substrate. This is an important consideration since in the case of high steps patterned on the substrate, coverage of the steps by the deposited material may not be conformal. The resulting non-uniformity of thickness of the deposited layer can cause reliability problems in the final device.

Etching: The process that immediately follows the photolithographic step is removal of the material from areas of wafer unprotected by photoresist. Various etching techniques are used for this purpose. Etching processes are characterized by their selectivity and their degree of anisotropy. Anisotropic etching occurs in one direction only, in contrast to the undesired isotropic etching, in which

material is removed at the same rate in all directions. The etching processes used in IC fabrication can take place either in a liquid (wet etching) or gas (dry etching) phase. They can be purely chemical, purely physical, or a combination of both.

Diffusion: Solid state diffusion is a process which allows atoms to move within a solid at elevated temperatures. Diffusion is a commonly used technique in IC fabrication to introduce dopants into the semiconductor substrate. Dopants affect the conductivity or change the type of conductivity (from n to p type or vice versa) of selected regions within the substrate. The objective is to achieve the desired impurity profiles (concentration of impurities as a function of the distance from the wafer surface) and eventually create junctions (surfaces that separate regions with dopings of alternate polarity on each side at the desired depth beneath the wafer surface.)

Implantation: Ion implantation is the alternative to the diffusion technique of dopant introduction used in IC manufacturing. From the processing point of view, the main difference between these two techniques lies in the significantly lower process temperature in ion implantation. This advantage combined with a much decreased lateral spreading of the doped region as well as overall better control over the dopant profile during ion implantation have led to the preferential use of this technique in high-density microcircuit fabrication.

Based on the above processing steps, various technologies have been developed to produce different kinds of IC components. Among them, NMOS (N-channel Metal-Oxide Semiconductor) and CMOS (Complementary MOS) are extensively used technologies currently.

NMOS technology is characterized by the negative doping of the transistor

channel. There are two main types of transistors produced by NMOS technology, namely depletion and enhancement transistors. The names reflect the effect an increasing effective field has on the concentration of the conducting impurities in the transistor channel. While in CMOS technology, the silicon substrate is selectively doped with negative or positive impurities. Three types of this technology are in use, namely p-well, n-well and twin-tub. Here the names reflect the technique employed to isolate regions with alternate doping. Since CMOS technology employs both NMOS and PMOS transistors to form logic elements, it has an advantage in a sense that the particular logic elements only draw significant current during the transitions, allowing power consumption to be minimized. This accounts for the reason why CMOS technology is recognized as a leading contender for existing and future VLSI systems.

2.2 Statistical Characteristics of the IC Process and Distributions of Process Disturbances

It is well known that random fluctuations, or disturbances exist in any IC fabrication process. Due to the existence of these disturbances, each manufactured wafer has a different and unique processing history. Furthermore all fabricated ICs on a wafer are different from one another because of random fluctuations in processing conditions across a wafer. Some of them may be affected by process disturbances in such a way that they do not meet expected specifications. In some cases, process instabilities may even cause only a small fraction of all

fabricated ICs to have an acceptable performance, hence significantly decreasing the manufacturing yield.

In this section, the principal disturbances in IC process and their effects on IC device performances will be classified and described. The distributions behind these disturbances will also be discussed.

2.2.1 Process Disturbances

The statistical variations of an IC process arise from the existence of a set of low level, non-measurable, non-controllable, independently statistically varying physical quantities, called process disturbances[14]. The exact definition of process disturbances will depend on the process model. Some examples of process disturbances include the diffusivity of a dopant, like boron and arsenic, to the substrate, variance of implantation profile spread of a dopant, various silicon dioxide growth rates, mask misalignments and surface state density, etc.[15] Sources of the random disturbances that occur in the IC fabrication process can be classified as two categories, namely global fluctuations and local fluctuations [3].

Global Fluctuations in IC Process

Among the sources of process disturbances, there sources of fluctuations affect chip performances in a random but globally homogeneous way. In other words, the effects are independent of the physical location of a particular chip or device on a chip during manufacturing. Although the local fluctuations are also generally characterized by affecting all devices on wafer/chip approximately equally, they are small compared to variances between chips, and in turn between wafers.

The sources of the global fluctuations can be further classified as:

• **Instabilities in the process conditions.** Although some of the process conditions can be controlled accurately, as mentioned in section 1.2, others such as the turbulent flow of gases used for diffusion and oxidation, furnace temperatures, etc. cannot be controlled as quite accurately. Because of these instabilities, each area of a wafer is exposed to slightly different environmental conditions, and hence no two manufactured chips can possibly have identical performance.

• **Material non-uniformities.** These are variations in the physical parameters of the chemical compounds and other materials used in manufacturing process. They are independent of the location of a chip on the wafer or the location of a wafer in the lot. Typical examples of material instabilities are fluctuations in the purity and physical characteristics of the chemical compounds, density and viscosity of photoresist, wafer and gas contamination, etc.

• **Translational mask misalignments.** These are errors in the position of a lithography mask with respect to the features already engraved on the surface of a wafer. They cause all edges defined by the mask to be shifted by the same amount with respect to the boundaries of the regions that already exist on the surface of the wafer. Therefore, the geometry of an IC could be significantly deformed from that of an ideal one, and hence wrong electrical connectivity could result.

• **Human factors and equipment failures.** These include imprecise equipment settings, imprecise equipment calibrations, etc.

Local Fluctuations in IC Process

In contrary to the global fluctuations, local fluctuations tend to affect ICs in some specific locations on a wafer. There are two main sources of spatial fluctuations.

• **Substrate inhomogeneities.** These are local disturbances in the properties of substrate wafers and are of three types: spot defects, dislocations and surface imperfections. Spot defects are local disorders in the structure of the lattice inside a semiconductor material. Dislocations are geometrical irregularities in the regular structure of the crystal lattice. These factors may cause serious or fatal changes in chip functionality.

• **Pattern Transfer.** While translational mask misalignments uniformly affect all chips on a wafer, rotational misalignments, pattern shrinkage or bloating and optical aberrations tend to dramatically affect those chips which are closer to the edges of the wafer.

2.2.2 Distributions of Process Disturbances

From the above discussion we can observe that process disturbances can cause either global or local deformations, and process induced deformations of any kind, local or global, geometrical or electrical, are all random.

Usually the variations caused by a global process disturbance that are observed within a single IC chip are small because all devices from a single IC chip are located very close to each other and therefore have very similar "process histories". Thus it can be expected that a specific parameter of all devices within a single chip to be similar and close to a certain mean value. Such is not the case, however, for the devices from different chips, since the mean process conditions for one chip can be quite different from that of another one, especially if they

are on different wafers.

The above discussion suggests that each global process disturbance should have a mean value characterizing average process conditions for some local area. The actual condition for a specific location within this area can be modeled by a variable randomly fluctuating about this local mean value with some local standard deviation. And these local parameters should also randomly change from one local area to another.

To account for local and global variations in device parameters, we can employ a multilevel structure for the random variables that characterize process disturbances [1]. With such a structure, disturbances are generated by a hierarchically defined random number generator (RNG) at levels that correspond to natural divisions, i.e. at the lot, wafer and chip levels.

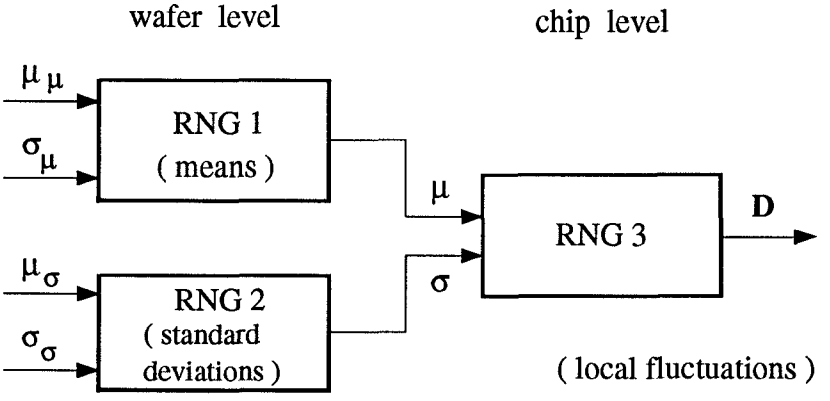


Figure 2.1: Two level structure of RNG's simulating local and global disturbances

Figure 2.1 shows a two-level structure composed of three RNG's. This structure is capable of generating disturbance data for a wafer, and this structure can be easily extended to the lot level.

As introduced in Section 2.2.1, the variations of an IC process are usually caused by a large number of independent physical quantities. Consequently, it is quite reasonable to assume that the random variables representing process disturbances are normally or log-normally distributed based on the Central Limit Theorem in probability [16]. This assumption has proved convenient and realistic in statistical simulation of IC the process. With this assumption, we can completely specify process disturbances by identifying their means and standard deviations.

2.3 Analytical Models of IC Process and Computer Simulation

In order to design and simulate an IC process for ideal yield, an accurate and efficient process model is a necessity. Traditional approaches for process simulation, such as that implemented in SUPREM [17], employ numerical models to characterize each fabrication step. These models, which are expressed in terms of partial differential equations (e.g. the diffusion equation), are solved using numerical techniques to produce the nominal profiles of impurities in silicon. The impurity profiles can then be used by device simulators such as SEDAN [18] and MINIMOS [19], in which semiconductor device models are described by a system of partial differential equations, which are solved numerically to produce I-V characteristics from which device parameters can be extracted. While such simulations can produce results that are accurate for a deterministic case, they are prohibitively expensive when used for statistical investigations.

To alleviate this difficulty, some statistical simulators were developed to serve

the purpose of statistical investigations. The statistical simulator FABRICS II is one of them [2]. Instead of using numerical models, FABRICS II employs analytical models which are solutions of the partial differential equations that describe each fabrication step under a set of restricted or simplifying conditions. Reasonable results have been achieved by these models. A detailed description of fabrication steps and analytical models implemented in FABRICS II can be found in [15]. However like other statistical simulators, there are some limitations with it. For example, it can not be used for the short-channel devices.

It should be emphasized that although FABRICS II has been employed to generate simulation data in our experiments, the algorithm presented in this thesis does not depend on any specific statistical simulator. It can be used with any simulator as long as the simulator can provide the required simulation data, or the manufacturing process itself.

2.4 IC Process Characterization Problem

The IC manufacturing processes vary from one to another, depending on the devices manufactured and the specific requirements for the devices. Hence, it is not difficult to understand that statistical characteristics are particular to a specific IC process. In order to simulate, monitor and diagnose an IC manufacturing process, we have to obtain sufficient knowledge about the process of interest. Specifically, we need to determine the quantitative effect of the controlled process conditions on the physical properties of various areas of the substrate. We also want to know the electrical characteristics of the manufactured devices after IC process is completed, and dependence of these characteristics on the process

conditions. Due to statistical nature of the IC manufacturing process, determining the statistical distributions and correlations of process parameters is also very important. The procedure of acquiring the above knowledge is usually referred to as IC process characterization.

In the statistical process simulator FABRICS II, an IC manufacturing process is viewed as a deterministic process with statistically varying inputs, i.e., the process disturbances. The deterministic process simulator actually consists of a number of analytical models that can accurately simulate the physics of a fabrication process. These models could be numerical, as with other process simulators. Thus given statistical distributions of the process disturbances, it is possible to simulate statistical variations within any process by means of Monte Carlo techniques [20]. This is accomplished in FABRICS II by using random number generators to generate a population of process disturbance values, which are then used as the inputs to the deterministic process.

A statistical simulator can produce a population of electrical and physical parameters pertaining to the finished ICs. Significant trends on process conditions due to the fluctuations of an actual process can be detected by statistically analyzing the simulated population of parameters.

In order to predict statistical attributes of output population, exact distributions governing the input population, namely process disturbances have to be known. In the literature [21], the task of obtaining probability distribution functions (pdf's) of the process disturbances is accomplished by extracting pdf's of the process disturbances so that joint probability distribution function (jpdf) of the simulated process outputs matches the jpdf of the measured process outputs.

2.5 Previous Approaches for IC Process Characterization

Because of the important role IC process characterization plays in modeling, analysis and simulation of the IC process, a lot of effort has been put into this problem. And several approaches have been employed to deal with different aspects of the problem. A brief summary will be given below about some representative approaches that are used extensively.

Worst case characterization [22] determines the extreme fluctuations of a given IC process in a sense that these fluctuations most adversely affect the performance. Once these fluctuations have been determined, simulations or the process can be skewed to worst case conditions. Thus the extreme performance of a population of manufactured ICs can be produced under the worst case. If these extreme simulated performances are within the specifications of design, we can expect that the bulk of performances of actually manufactured ICs to be within the specifications as well.

I-V matching is another technique used for IC process characterization [23]. This is actually a process of extracting some parametric values of the analytical device model, so that the model is in agreement with the measured device. If the device is typical of the process, then in a sense this is considered to be a characterization of the process.

Although these techniques have been extensively used, they have some deficiencies associated with them. One of the inherent deficiencies of these approaches is that the process statistics are modeled with correlated parameters, and an independent set of statistical variables is not determined.

A methodology for statistical IC process characterization was proposed to compensate for the deficiency of the above approaches [3].

In this approach the statistical process characterization problem is first formulated as a set of nonlinear equations of the form:

$$N(d_0 + d_e, p, y_0 + y_e) = 0 \tag{2.1}$$

where d_0 and y_0 are vectors of deterministic values and represent the nominal values of the disturbances and the outputs of an IC process, respectively. y_e and d_e are the statistical variations superimposed on the outputs of the process and the process disturbances. p represents the process conditions. Since $y = y_0 + y_e$ is related to $d = d_0 + d_e$ by nonlinear functions, the problem can be solved in two steps: first the nominal disturbance vector d_0 can be extracted given the nominal output parameter vector y_0, and then the statistical distribution of the disturbance variance vector d_e can be extracted given statistical distribution of the output parameter variance vector y_e.

Specifically, the nonlinear programming technique has been employed to solve this statistical extraction problem, which can be mathematically represented as follows.

$$\min_{d_0} \|N(d_0, p, y_0)\|_2 \tag{2.2}$$

and

$$\min_{\theta^d} \|G(\hat{\theta}^y, \theta^d)\|_2 \tag{2.3}$$

where $\hat{\theta}^y$ is a vector of moments that characterizes the distribution of y_e, and θ^d is a set of moments that characterizes the distribution of d_e. The map from

statistical distributions of process disturbances to that of output measurements is approximated by polynomial models. A detailed discussion on this approach can be found in [24].

Based on these models, a computer software PROMETHEUS [21] was developed to diagnose the nominal disturbances using optimization techniques. This approach proves to be efficient when the number of disturbances to be diagnosed is not large, for instance, smaller than 10. It may not be as efficient in the context of high dimensionality, because it is extremely difficult to build an accurate polynomial model for high dimensional cases. Nevertheless, once the nominal disturbances have been determined, they can be used in statistical process control.

2.6 The IC Process Diagnosis Problem and Previous Approaches to the IC Process Diagnosis

In the literature the IC process diagnosis is defined as the inference of the changes in the statistics of the process disturbances. As discussed earlier, significant shifts of process disturbances from their nominal statistics will result in process faults. Due to their low level physical nature, however, some of the process disturbances can not be directly measured. They have to be inferred by the measurements obtained on the circuits or test structures, which have been fabricated by the process. This section provides a brief discussion on two previous approaches for the IC process diagnosis problem.

The first approach is a pattern recognition based method which was proposed by Strojwas and Director [46]. The output measurements used in this approach are taken from chip performance testing. Since the performance specifications are employed as the thresholds, only pass or fail data are generated. The drawback of this method is that since the data are taken from chip performance testing which is carried out in the evaluation stage, the method is not very sensitive to the variations in the IC process. In order to diagnose a process fault, a large amount of data is required. Moreover the approach has limited observability, because only the faults already in the data base can be identified.

The second approach is polynomial approximation method proposed by Spanos [14]. The measurements used in this approach are taken from test chips, and therefore are more sensitive to the variations of the IC process. Polynomial models are employed to approximate the relation between the process disturbances and the output parameters in this approach. Then the diagnosis problem is converted to an optimization problem, which is solved iteratively using nonlinear programming techniques. The main drawback of this method is that with the high dimensionality of the problem, it is very difficult to build an accurate polynomial model at each operating point of the process, hence the method is very intensive in computation. Therefore it is impractical for use in real-time diagnosis of large variations of the process disturbances.

The approach proposed in this thesis is based on backpropagation neural networks. The mapping from the process output parameters to the process disturbances is accomplished by fitting a neural network to the IC process of interest. A distinguishing feature of this approach is that the diagnosing of process disturbances is very short once the neural network has been fine tuned to the IC

process. This feature makes it practical to be used in real-time monitoring and diagnosing of process faults. And since the neural networks are much easier to set up than the polynomial models, the difficulty due to the high dimensionality of the problem is alleviated using the neural network based approach. The details of this new approach will be the main subject of the remaining part of this thesis.

Chapter 3

Surveillance and Diagnosis of Statistical Parameters in IC Process

3.1 Introduction

The rapid advances in the miniaturization of today's microchips have resulted in ever-increasing circuit density and complexity. In the fabrication of such microchips, defects in materials, contamination in chemicals and deviations in process parameters can lead to dramatic yield reductions. The fabrication of integrated circuits requires a number of processing steps. If one or more of these steps are incorrect or exceeds certain design limits, the devices fabricated will either fail or not perform as intended. It has therefore become imperative to monitor the IC process continuously at each step in order to take immediate corrective actions in case of a deviation.

To assure that a maximum number of devices will be functional and meet the desired performance criteria, a sample of the wafers is taken for a variety of quality control inspections and also for measurements to determine the effects of any process variation upon the wafer properties. Such an operation is carried

out after every major step in an IC fabrication process. For example, after a photolithographic step, the dimensions and the alignment of patterns generated in the photoresist will be checked at perhaps five chip positions on 1 out of 10 wafers. This serves to verify mask performance [25].

The real-time monitoring of equipment and processes in IC fabrication makes it possible to identify and eliminate problems before the yield of fabrication has already been adversely affected.

In practice an integrated circuit production line is monitored by gathering and analyzing information obtained from varieties of sources [26]. For example, the uniformity of various process parameters can be measured from specially designed process monitor chips (PMC's) stepped into product wafers, and from a process validation wafer (PVW) consisting entirely of test patterns [27], the process control data can be obtained from a wafer capable of isolating the behaviors of particular process steps. Additional information may be gathered on the behavior of processing equipment, on various environmental factors, and on a variety of other aspects of IC fabrication. The quantity of data which can be collected is, therefore, virtually unlimited. The real challenge lies in effectively utilizing this information to achieve a high level of control and operation efficiency.

Various techniques have been employed to monitor and detect non-uniformity in the IC process based on the data extracted from it. Among these techniques, the frequency plots (histograms), the two-dimensional contour maps and the three dimensional perspective drawings are extensively used, which have been discussed in the literature [28][29]. These techniques have been proved effective in monitoring and exhibiting variations of some parameters such as sheet resistance

and film thickness in the IC process. To be able to detect or predict a significant trend of device performance in IC process, however, a large number of process parameters need to be monitored and evaluated simultaneously. The statistical process control technique has been proved to be more effective in this case to deal with the problem.

Due to the importance of IC process characterization, a substantial effort has been made to reveal and analyze the statistical nature of the IC process in the literature. As part of this effort, a new approach is proposed in this thesis, which incorporates the application of the statistical process control technique to achieve the goal of monitoring and detecting deviations of IC process parameters based on the measurable outputs of the process.

3.2 Statistical Process Control Technique and Its Application in IC Yield Optimization

In any production process there exists a certain amount of inherent or natural variation no matter how well it is designed and maintained. To manufacture products meeting the designed specifications, a production line has to be capable of operating with little variability around the nominal dimensions of the product's quality characteristics. In other words, the production process has to be kept in a stable condition. Statistical process control is a powerful collection of tools useful in achieving process stability and improving capability of the process through the reduction of variability.

The inherent variability in any production process arises from the cumulative effect of many small essentially unavoidable causes. As indicated in Section 2.2,

for example, variations in the IC process are due to the existence of a large number of low level physical quantities. When the inherent variability in a process is relatively small, we usually consider it to be an acceptable level of process performance, or the process is said to be under statistical control.

There are other kinds of variability which may occasionally be present in a process. Such variability is generally large compared to the inherent variability. The sources of such variability in the IC process may include imprecisely adjusted equipment, improperly controlled environmental conditions, defective raw materials and operator mistakes. These sources of variability are referred to as assignable causes in the field of statistical quality control. Occurring in a random mode, the assignable causes result in a shift in the process state and in turn cause a large proportion of the products not to meet the requirements. A process that is operating in the presence of assignable causes is said to be out of control.

A major objective of statistical process control is to quickly detect the occurrence of process shifts or assignable causes so that investigation of the process and corrective action may be undertaken before many nonconforming products are manufactured. Techniques such as the control chart can be used to monitor the process outputs and to detect when adjustments in the inputs are required to bring the process back to an under-control state.

The introducing of statistical process control to IC fabrication is motivated by the objective of optimizing process yield. The manufacturing yield of an IC process is defined as the ratio of the number of chips that successfully pass all of the selection steps in the process with respect to the total number of chips that enter the fabrication process at the very beginning [30]. It can be simply

expressed as:

$$Y_m = \frac{N_f}{N} \tag{3.1}$$

where N is the maximum number of chips that can be fabricated in an ideal IC manufacturing process, and N_f is the number of chips that have been classified as fault-free after the final selection step.

The yield of an IC process depends on many process parameters. In order to optimize yield, all those parameters whose changes have a substantial impact on yield have to be under strict control. The system of using statistical control to optimize process yield is illustrated in Figure 3.1.

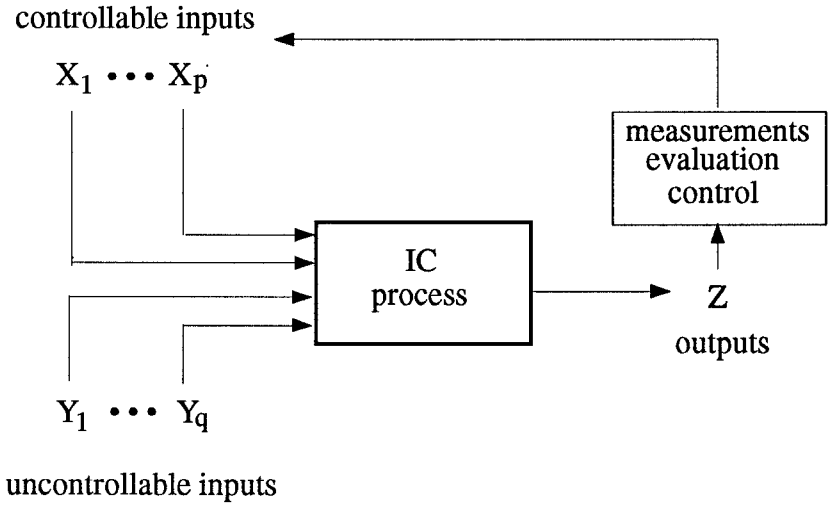


Figure 3.1: The system of yield optimization

The inputs $X_1, \cdots, X_p$ are controllable factors, such as process time, environmental temperature and gas flux etc., while the inputs $Y_1, \cdots, Y_q$ are uncontrollable factors, such as raw materials and human (operator) factors. The

manufacturing process transforms these inputs into a finished device which is expected to have specific electrical functions and meet some designated requirements at the same time.

Choosing input variables to perform statistical control is one of the critical steps in yield optimization. The key input variables that significantly influence the yield have to be identified in order to achieve high efficiency in control. Towards this end, first the analysis of sensitivity of yield to process parameters needs to be done to find the best set of input process parameters. Designed experiments proved to be extremely helpful in discovering the key variables. A designed experiment is an approach to systematically varying the controllable inputs and observing the effect these inputs have upon the output device parameters. In the literature [3.9], designed experiments are also referred to as a major off-line quality control tool, because they are often used during development activities and early stages of manufacturing, rather than as a routine on-line or in process control procedure.

Once the best set of inputs has been determined, they will be used as the target of monitoring, and appropriate control charts will be built associated with them. Whenever an out-of-control signal emerges from any of these control charts, an investigation will be initiated to search for the assignable causes behind it. Corrective measures will then be taken if an assignable cause really exists and is found. By continuous surveillance of the entire fabrication procedure, we can expect to keep the fabrication line in a stable condition and therefore achieve a high yield of the process.

3.3 Design and Application of Control Charts

As a main tool of statistical quality control, the control chart is an on-line process control technique widely used for the purpose of detecting the occurrence of process shifts or assignable causes [11]. The control chart may also provide information useful in improving the process. Note that the eventual goal of statistical process control is the elimination of variability in a process. It may not be possible to completely eliminate variability, but the control chart is an effective tool in reducing variability as much as possible.

There are several types of control charts built in different cases to detect the potential variability of a process for the best effectiveness. A typical control chart is shown in Figure 3.2, which is a graphical display of a statistical characteristic that has been measured or computed from a sample versus the sample number or time.

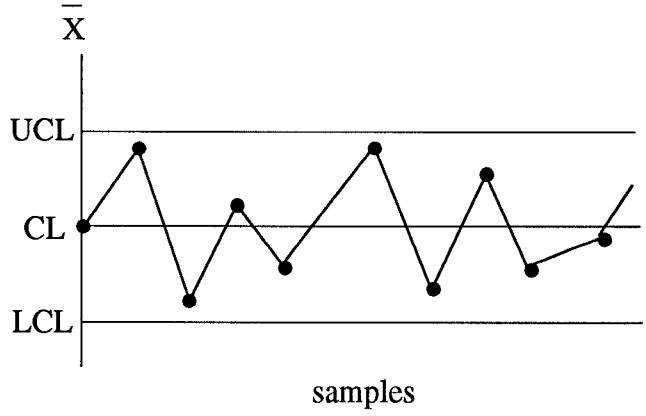


Figure 3.2: A typical variable control chart

The control chart shown in Figure 3.2 consists of a center line (CL) and two other horizontal lines that parallel the center line. The line above the center line

shown on the figure is called the upper control limit (UCL) and the line below called the lower control limit (LCL). These control limits are chosen so that if the process is under control, nearly all of the sample points will fall between them. As long as the sample points fall within the control limits, the process is considered to be under control, and no action is necessary. A point that falls outside of the control limits, on the other hand, is interpreted as evidence that the process is out of control; investigation and corrective actions are required in this case.

Figure 3.2 is actually a control chart for the IC device parameter channel length. Each point plotted on the chart is computed from ten consecutive samples from process simulation. Because this control chart utilizes the sample average $\bar{x}$ to monitor the mean of the channel length, it is usually called a $\bar{x}$ control chart. Note that all points in the figure fall within the control limits, so the chart indicates that the process is in statistical control.

A general model for a $\bar{x}$ control chart can be described as follows.

$$UCL = \mu_x + k\sigma_x$$
$$CL = \mu_x$$
$$LCL = \mu_x - k\sigma_x$$

where x is a sample statistic of a process parameter, μ_x stands for the mean of x and σ_x for the standard deviation of x. k is the "distance" of the control limits from the center line.

As stated at the beginning of this section, there are various control charts that are employed for the different purposes. But generally control charts can

be classified into two types, namely variable control charts and attribute control charts. Following is a brief description of the two types.

If the quality characteristic can be measured and expressed as a number on some continuous scale of measurement, it is usually called a variable. In such cases, it is convenient to describe the quality characteristic with a measure of central tendency and a measure of variability. Control charts for central tendency and variability are collectively called variable control charts. The $\bar{x}$ chart is the most widely used variable control chart for controlling central tendency, while charts based on either the sample range or the sample standard deviation are used to control process variability.

On the other hand, many quality characteristics are not measured on a continuous scale or even a quantitative scale. In these cases, we may judge each unit of product as either conforming or nonconforming on the basis of whether or not it possesses certain attributes, or we may count the number of nonconformities (defects) appearing on a unit of product. The control charts for such quality characteristics are called attribute control charts.

Since the central tendency is our main concern in IC process surveillance and control, $\bar{x}$ chart is the appropriate control chart which can serves our purpose. It is actually applied as the main chart in the approach proposed in this thesis, as will be seen in Chapter 6.

Note that due to the statistical nature of the observed parameters, one or more sample points may fall outside of the control limits occasionally even though the process is actually under statistic control. These out-of-control signals result in a "false alarm" that might cause unnecessary cost. By moving the control limits further from the center line, we can decrease the risk of a false alarm.

However, widening the control limits will also increase the risk of another type of error, namely the risk of a sample point falling between the control limits when the process is really out of control. Therefore, specifying the control limits is one of the critical decisions that must be made in designing a control chart.

In practice control limits are customarily determined as a multiple of the standard deviation of the statistics plotted on the chart. And the multiple is usually chosen as 3, although a more reasonable choice should be dictated by the economic considerations in a specific process. More complete discussion of this subject can be found in [11].

3.4 Surveillance and Monitoring of Process Disturbances Using Control Charts

From discussions in Section 2.2, we know that the reason for a yield drop of an IC process can be attributed to significant variations of the process disturbances in most cases. Obviously, if we can implement real-time monitoring and statistical control of the process disturbances, a great step will be made towards the objective of optimizing manufacturing yield.

Although statistical quality control techniques have been successfully applied to various stages of the IC manufacturing, the idea of establishing control charts for IC process disturbances in order to implement real-time direct monitoring of the process disturbances has not been attempted. The main difficulty lies in the fact that most of the process disturbances can not be observed and measured directly.

Even though a number of approaches have been developed to extract those

unmeasurable process parameters from the measurable outputs, their applications in real-time surveillance and control of IC process are strongly restricted by the intensive computations that they involve. A real-time based approach that is quick and simplistic in computation is needed for this purpose.

We have been aiming to establish control charts for the process parameters and disturbances that have a significant impact on yield. This idea has been made realistic by a successful algorithm relating the measurable outputs of an IC process to the process disturbances in a timely fashion. Actually, shifts in process disturbances are tracked based on the information provided by the on-line measurements of process outputs. This task is accomplished by tuning a multilayer neural network to the inverse map of the IC process of interest. In other words, the tuned neural network uses the on-line output measurements of the IC process as its inputs, and its outputs are the process parameters and disturbances we want to monitor. As such it is possible to get a real-time sampling of the process disturbances as long as a continuous sampling of the on-line output measurements are available, which turns out not to be a very difficult task.

The tuning or learning of the neural network employed is carried out after the design or development stage of an IC process, and uses a process simulator that have been tuned to the process of interest. The learning procedure initiates from the generation of simulation data for the process disturbances. Then designed experiments are employed to input a set of simulation data to the fine tuned process simulator, and the corresponding outputs of the simulator are recorded. The next step is to generate training patterns for the neural network. And this is done by using the pair of output and input sets of the process simulator as

the inputs and outputs of the neural network, respectively. Figure 3.3 shows the learning procedure.

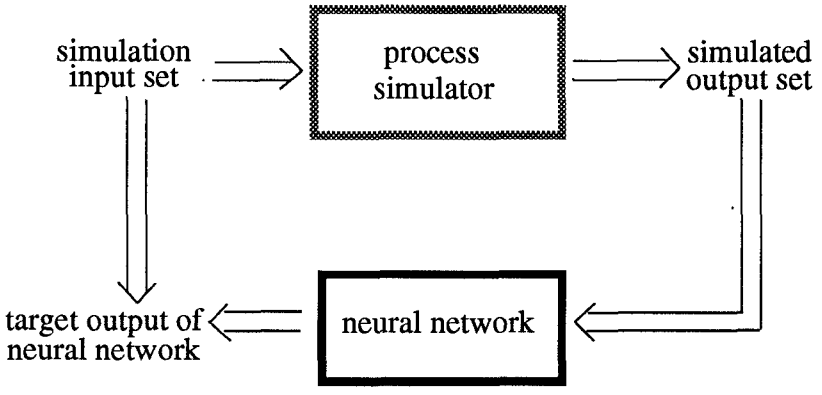


Figure 3.3: The system of implementing inverse of IC process

After the learning procedure the neural network is then ready to be used with the real IC fabrication line that has been simulated by the process simulator. The samples taken from the outputs of the real fabrication line thus can be used to obtain real-time samples of the process disturbances through the fine tuned neural network. With real-time sampling of the process disturbances, the control charts can be built to implement real time monitoring of these disturbances.

It is worth mentioning here that in order to minimize the false alarms and "fail to detect" errors that might result from the IC process itself and measuring process, control charts are also established for IC process output measurements. The detailed designing procedure of these control charts will be discussed in Chapter 6.

3.5 Practical Considerations

The main idea of implementing real-time monitoring of the process disturbances by using a neural network in conjunction with a process simulator has been introduced in the last section. The realization of the approach, however, has to be based on some practical considerations which will be discussed below.

As described in Section 2.2, every process disturbance can be represented as a random variable. Hence what we actually deal with is the statistics of these process disturbances. Furthermore, an assumption is made that the process disturbances to be monitored are independent from each other. This simplifies the problem by ignoring the correlations between the disturbances.

Another important factor that needs to be considered is the fact that the highly nonlinear relationship between the output measurements and the process disturbances makes the tuning of a neural network extremely difficult or even impossible in a global range, if the statistics of the process outputs and disturbances are used directly. In other words, the inverse of the relationship, i.e. from the statistics of outputs to the statistics of disturbances, is not valid over the global range, due to the fact that the functions relating disturbances to outputs are not one-to-one. A practical strategy to deal with this difficulty is to use coding.

Specifically, some properly spaced multiple thresholds in the control charts are first established for the statistics of both outputs and process disturbances. A specific number out of several preselected numbers is assigned to a variable which falls between any two thresholds. An ordered combination of the numbers assigned to all input variables then constitutes a digit string, called a code. The same thing is also obtained from the output variables. These codes will be

employed as the inputs and target outputs of the neural network to be tuned. It will be explained in Chapter 6 that the relationship between the input and output codes can be built on a one-to-one basis as long as no two process disturbances under surveillance affect exactly same set of output parameters.

After a neural network is fine tuned, it can then be used to monitor process disturbances in practice. To keep track of variations of the process disturbances, first of all, the on-line measurements of output parameters are taken from a real production line. These measurements will then be employed to obtain the statistics of the process outputs. The next step is to encode these statistics by the same procedure stated above. The generated code will be used as the input to the fine tuned neural network. And the output of the neural network will be the corresponding code for the statistics of disturbances in the real fabrication line. The decoding will create the samples of disturbance statistics, which will be plotted on their control charts.

The following two chapters will be devoted to the discussion of the theoretical and practical aspects of the neural network based approach.

Chapter 4

Mathematical Foundations of the Neural Network Based Methodology

4.1 Introduction to Neural Networks [31]

Neural networks are massively parallel systems that rely on dense arrangements of interconnections and simple processors. The systems actually consist of many nonlinear computational elements operating in parallel and arranged in patterns reminiscent of biological neural nets. Computational elements or nodes are connected via weights that are typically adapted during use to improve performance.

Neural network architectures are significantly different from traditional single processor computers. Traditional computing machines have a single CPU that performs all of its computations in sequence. In contrast, a neural network consists of a large number of processing units, called neurons which perform simple computations simultaneously. Each processing unit has four important components: input connections, through which the unit receives activation from other units; a summation function that combines the various input activations into a single activation; a threshold function that converts this summation of

input activations into an output activation, and output connections by which a unit's output activation arrives as an input activation at other units in the system. The inter-unit connections in neural networks are typically assigned numeric weights that modulate the activation passing through the connections. The power of the neural network lies in these interconnections.

As a useful computing tool, a neural network has strong capabilities to classify and recognize patterns, to perform pattern mappings, and to recover or complete patterns with missing segments. In this thesis the pattern mapping capability of neural networks will be employed. A further discussion on neural network's mapping capability is provided in the following section.

4.2 The Approximation Capability of Backpropagation Neural Network [32][33]

As one of the paradigms of neural networks, backpropagation is currently the most widely applied neural network architecture. This popularity primarily revolves around the ability of backpropagation networks to learn complicated multi-dimensional mappings.

The backpropagation neural network architecture is a hierarchical design consisting of fully interconnected layers of processing units which have been introduced in the last section. Backpropagation belongs to the class of mapping neural network architectures and therefore the information processing function it carries out is the approximation of a bounded mapping of function $f : \boldsymbol{A} \subset \boldsymbol{R}^n \longrightarrow \boldsymbol{R}^m$, from a compact subset A in n-dimensional Euclidean space to a bounded subset $f(A)$ of m-dimensional Euclidean space, by means of training on examples $(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k), \cdots$ of the mapping, where $y_k = f(x_k)$.

It will always be assumed that such examples of a mapping f are generated by selecting x_k vectors randomly from A in accordance with a fixed probability density function $\rho(x)$. A typical structure of backpropagation neural network is shown in Figure 4.1.

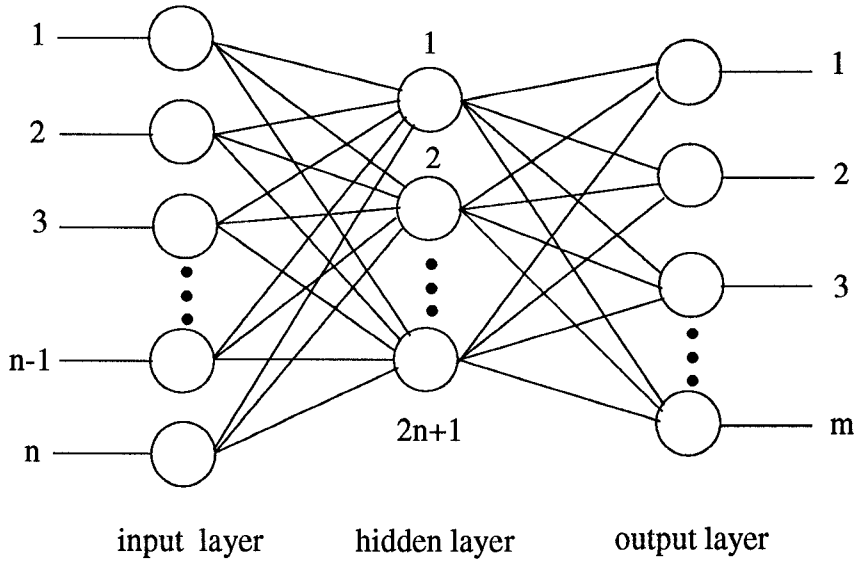


Figure 4.1: The typical structure of a backpropagation neural network

The scheduling of the backpropagation network's operation consists of two stages. The first stage (the forward pass) starts by inserting the vector x_k into the network's first layer, or the input layer. The processing elements of the first layer transmit all of the components of x_k to all of the units in the second layer of network. The same operations are performed between the second and third layers, and so on, until finally the m output units in the output layer emit the components of the vector $y_k\prime$ (the network's estimate of the desired output y_k).

After the estimate $y_k\prime$ is emitted, each of the output units is supplied with its component of the correct output vector y_k, starting the second stage (the

backward pass). The output units compute their squared errors, $\delta_k = \| y_k - y_{k'} \|_2$, and then transmit them back to their ancestor layer. The ancestor layer updates its weights and transmits its corresponding errors to the layer of a level higher. This process continues until the first hidden layer has been updated.

The two-stage cycle is continued until the network reaches a satisfactory level of performance (the error level is lower than the predesigned). And then the network will be able to implement or approximate the functional mapping $f : \boldsymbol{A} \subset \boldsymbol{R}^n \longrightarrow \boldsymbol{R}^m$.

The above discussion leads us to come up with a question — what kinds of functional forms can be approximated by the backpropagation neural networks? This problem had drawn a lot of concern and explorations in the past. A clear insight into the versatility of neural networks for use in function approximation came with the discovery of a new explanation to a classic mathematical result of Kolmogorov. The explanation states that for any continuous function $f : [0,1]^n \subset \boldsymbol{R}^n \longrightarrow \boldsymbol{R}^m$, there must exist a three-layer neural network (having an input layer with n processing elements, a hidden layer with $(2n+1)$ processing elements, and an output layer with m processing elements) that implements function f exactly. This result gave hope that neural networks would turn out to be able to approximate any function that arises in the real world.

Mathematically, this result can be more precisely expressed in the following theorem.

Theorem: Given any $\epsilon > 0$ and any L_2 function[1] $f : [0,1]^n \subset \boldsymbol{R}^n \longrightarrow \boldsymbol{R}^m$, there exists a three-layer backpropagation neural network that can approximate

[1]An L_2 function is a squared-integrable multi-variable function defined in a bounded domain.

function f to within ϵ mean squared error accuracy.

A complete proof of this theorem can be found in [32]. It is important to realize that although this theorem proves that a neural network with three layers is always enough in approximating any L_2 function, it is often essential to have more than three layers. This is due to the fact that for many problems, an approximation with three layers would require an impractically large number of hidden units, whereas an adequate solution can be obtained with a tractable network size by using more than three layers.

Although the above theorem guarantees the existence of a multilayer neural network with the correct weights to accurately implement an arbitrary L_2 function, how to determine the numbers of hidden layers and hidden units in the desired network is still an open question. A discussion on this topic can be found in [34].

4.3 Mathematical Formulation of the Process Surveillance Problem

From the discussion in Section 3.4, we know that the goal of process surveillance is achieved through obtaining real-time samplings of process disturbances and plotting them in the corresponding control charts. Due to the fact that it is extremely difficult or even impossible to directly sample all of the process disturbances of interest, we are challenged to acquire the desired information from the on-line measurements of the process outputs.

By using a statistical process simulator, the statistics of process outputs can be simulated provided that the statistics of the process disturbances are known.

As mentioned in previous chapters, what relates the statistics of the process disturbances to that of output parameters in the process simulator is a set of analytical models, and most of these models are nonlinear functions. Hence from the mathematical point of view, the process simulator actually implements a mapping from the input (process disturbances) space to the output (process output measurements) space. If this mapping is named as the direct mapping, then the mapping from the output space to the input space will naturally be the inverse. They are expressed in the following equations.

$$\text{Direct mapping:} \qquad \boldsymbol{Y} = N(\boldsymbol{X}) \tag{4.1}$$

$$\text{Inverse mapping:} \qquad \boldsymbol{X} = G(\boldsymbol{Y}) \tag{4.2}$$

where $\boldsymbol{Y}$ is a vector representing the statistics of output parameters, and $\boldsymbol{X}$ is a vector of statistics of process disturbances. N stands for a set of nonlinear functions and G is the inverse of N.

The statistical process simulator can implement the direct mapping, but what we actually need is the inverse mapping. Note that if we can implement the inverse mapping, we will be able to acquire the information we want about process disturbances when the output measurements become available. The universal approximation capability of a multilayer backpropagation neural network makes it one of the possible approaches to realize this inverse mapping.

The question of how a neural network can learn and implement a nonlinear and multi-variable mapping has been attracting a lot of attention and effort. In Section 4.2 it was shown that for a squared-integrable multi-variable function defined in a bounded domain, there always exists a three-layer backpropagation neural network that approximates the function to within any mean squared error

accuracy. The primary idea behind this result is the consideration of multi-variable Fourier series expansion.

This can be explained as follows. For a squared-integrable function $g(x)$ over a bounded domain, we can normalize its domain to be a unit cube $[0,1]^n \subset \boldsymbol{R}^n$, and then expand it into an n-dimensional Fourier series:

$$g(x) = \sum c_k exp(2\pi i \boldsymbol{k}^T x) \tag{4.3}$$

$$c_k = \int_{[0,1]^n} g(x) exp(-2\pi i \boldsymbol{k}^T x) dx \tag{4.4}$$

where $\boldsymbol{k}= (k_1, ..., k_n)^T$ is an n-dimensional vector in the frequency domain, and the summation in equation (4.3) is made from $-\infty$ to $+\infty$ for each component of $\boldsymbol{k}$. It is easy to understand that if the summation in (4.3) is accumulated over finite indices, for example, from $-N$ to $+N$, then an approximation is obtained, and the accuracy of approximation depends on how large the integer N is. In the literature, the integer N is interpreted as the equivalence to the number of learning iterations [35].

For an IC process, if the statistics are distributed over a bounded work domain for all process parameters and disturbances of interest, then we can always design a three-layer backpropagation neural network to learn and implement the inversion of what the process simulator does. However, since most of the conventional learning laws applied in backpropagation neural networks, such as the momentum version, the delta and generalized delta learning laws are of the point-learning type, how to design a neural network which can globally learn a function or mapping is still an open question. Some learning algorithms incorporated with random sampling and random search methods may help to cover

more learning domains, but they are not a global and fast approach to implement an entire mapping. Before any effective algorithm is created for global learning, we may design a backpropagation network to learn and implement the mapping in a specific range by some conventional learning algorithms.

Chapter 5

Realization of Neural Network Based Approach for IC Process Surveillance

5.1 Generation of Training Samples for the Neural Network

Among many different methods of selecting the values of input variables for training the neural network [36][37], there are three approaches that have considerable intuitive appeal. They are called random sampling, stratified sampling and Latin hypercube sampling (LHS) [10]. A brief description of the first two is given below, which is followed by a more detailed discussion on LHS.

Random Sampling. The input values $x_1, \cdots, x_N$ are selected from the sample space of X in simply a random manner. This method of sampling is perhaps the most obvious one.

Stratified Sampling. Using stratified sampling, all areas of the sample space of X are represented by input values. Let the sample space S of X be partitioned into I disjoint strata S_i, and let $p_i = Pr(X \subset S_i)$ represent the size

of S_i. Obtain a random sample $x_{ij}, j = 1, \cdots, n_i$ from S_i. Here the sum of n_i for all samples is N. Note that if $I = 1$, we have random sampling over the entire sample space.

Latin Hypercube Sampling. The same reasoning that leads to stratified sampling, ensuring that all portions of S are sampled, can lead further. If we wish to ensure also that each of the input variables X_k has all portions of its distribution represented by input values, we can divide the range of each X_k into N strata of equal marginal probability $1/N$, and sample once from each stratum. Let this sample be $x_{kj}, j = 1, \cdots, N$. These form the X_k components, $k = 1, \cdots, K$. The components of the various X_k's are matched at random. Note that the strata are not necessarily of equal length but each stratum contains the same probability $1/N$.

To help clarify how strata or intervals are determined, a simple example is focussed on below, where it is desired to obtain $N = 5$ input vectors in two variables [38]. Let us assume that X_1 has a normal distribution on the range from A to B, as shown in Figure 5.1. And for simplicity, another assumption is made that A is the 0.001 quantile and B is the 0.999 quantile, namely

$$Pr(X_1 \leq A) = 0.001 \tag{5.1}$$

$$Pr(X_1 \geq B) = 0.001 \tag{5.2}$$

This would imply that the mean of the normal distribution is given by

$$\mu = \frac{A + B}{2} \tag{5.3}$$

and since for a standardized normal variable Z

$$Pr(Z \leq -3.09) = 0.001 \tag{5.4}$$

it follows that the standard deviation of the normal distribution is given by

$$\sigma = \frac{B - \mu}{3.09} = \frac{B - A}{6.18} \tag{5.5}$$

To divide the range evenly in probability, the intervals for $N = 5$ would appear as follows:

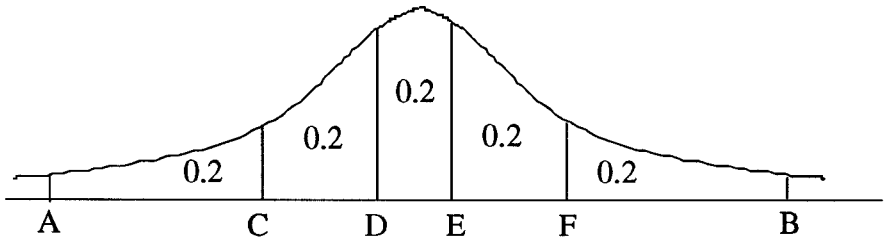


Figure 5.1: Evenly divided intervals in probability for X_1

where

$$Pr(A \leq X_1 \leq C) = Pr(F \leq X_1 \leq B) = 0.2 \tag{5.6}$$

$$Pr(C \leq X_1 \leq D) = Pr(D \leq X_1 \leq E) = Pr(E \leq X_1 \leq F) = 0.2 \tag{5.7}$$

Now assume that X_2 is a random variable which uniformly distributes on $[G, H]$. The corresponding intervals for X_2 would appear as shown in Figure 5.2.

The next step in obtaining the samples would be to pick specific values of X_1 and X_2 in each of their five respective intervals. This selection should be done in a random manner with the qualification that the selection should reflect the weight of the density across the interval. In the $[A, C]$ interval for X_2,

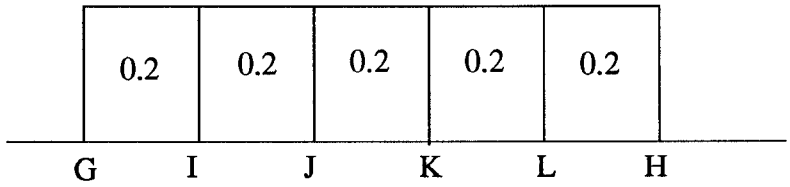


Figure 5.2: Evenly divided intervals for X_2

for example, values close to C will have a higher probability of selection than will those values close to A. Then the selected values of X_1 and X_2 are paired in a random manner to form the required five input vectors. This is done by associating a random permutation of the first N integers with each sample of input variables. For the present example, consider two random permutations of the integers 1, 2, 3, 4, 5 as follows:

permutation set 1: (5, 1, 3, 4, 2)

permutation set 2: (2, 4, 1, 3, 5)

Using the respective position within these permutation sets as interval numbers for X_1 (set 1) and X_2 (set 2), the following input vectors will be formed

Input Vector No.	**Interval No. for x_1**	**Interval No. for x_2**
1	5	2
2	1	4
3	3	1
4	4	3
5	2	5

Table 5.1: Formulation of input vectors using permutation sets

Figure 5.3 shows the two dimensional representation of the intervals selected

to form these input vectors. The asterisks in the diagram represent the specific pairs actually selected. It is obvious that if these pairs are projected onto the X_1 axis, the entire range of intervals of X_1 has been covered. This also holds true for X_2. And the same scenario can also be extended to higher dimensional cases. In fact a set of Latin hypercube sampling points in k-dimensional Euclidean space would cover the range of intervals of each of the k variables when projected onto the respective axes. This illustrates the nature of Latin hypercube sampling.

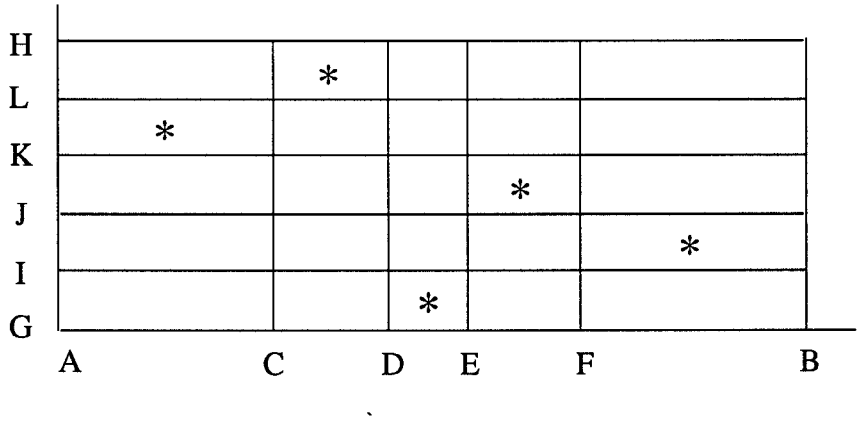


Figure 5.3: Two-dimensional representation of sample space

A distinct advantage of Latin hypercube sampling appears when the model output $Y(X)$ is dominated by only a few of the components of X, which is exactly the case in an IC process. Latin hypercube sampling ensures that each of those components is represented in a fully stratified manner, no matter which components might turn out to be important.

In order to effectively apply Latin hypercube sampling to our problem, an important factor, namely the distribution effect has to be taken into consideration in practice.

The procedure discussed above implies that different distributions used with

Latin hypercube sampling have the effect of concentrating input variable selection in different variable subranges. As a result, a variable may be assessed as important when sampled on a specified range with one distribution and deemed unimportant when sampled on the same range but with a different distribution. Therefore it is important to determine the extent of the distribution effect on model output. If there is a significant distribution effect, selection of appropriate ranges and distributions for the input variables is critical.

The distribution effect can significantly influence the mapping created by the neural network. In particular, if the distribution effect results in insufficient representation of variable subranges that have important influences on model output, the weights in the neural network can lead to erroneous sensitivity analysis conclusions.

On the other hand, we can also make use of the distribution effect to serve a specific purpose. In fact when performing the surveillance of the IC process disturbances, we can facilitate the task by establishing some limits, just like the control limits in the control charts. Since the samples taken near the limits dominate the determination of the thresholds, or the control limits for the output variables, it is fairly preferable to take more samples in the area nearby the control limits of the input variables than in the other areas.

A specific distribution like that shown in Figure 5.4 can be constructed to serve the purpose.

By using Latin hypercube sampling, the areas near UCL and LCL will be more heavily sampled than the area between them. From Figure 5.4, it is clear that the area in the midway between two control limits and the areas far beyond the control limits are lightly sampled. This is exactly what we want because the

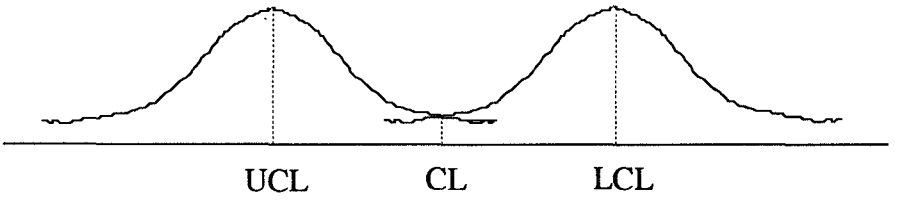


Figure 5.4: Specific distribution for unevenly sampling

samples from these areas have little influence on determining the thresholds of output variables.

In this section we have discussed the selection of training samples for the neural network. This approach will be employed in the real experiments to be presented in Chapter 6 where the real parameters of the IC process will be analyzed in detail.

5.2 Determination of Neural Network Structures and Training of Neural Networks

Once the input and output samples have been determined, the next task we are faced with is to construct a neural network with an appropriate structure to map the outputs to the inputs of the IC process. Determining the neural network structure is very important because of the role it plays in training of the neural network.

Although it has been known that a three-layer backpropagation neural network can theoretically perform all kinds of functional mappings, a solution is yet to be found as to the specification of the three or multiply-layer backpropagation architecture. As stated in Chapter 4, a three-layer network has an input, a

hidden and an output layer. The only difference between a three-layer network and a multi-layer (more than 3) one lies in the number of the hidden layers and hidden units. Usually the application dictates the number of input units and the number of output units in a rather obvious fashion. But very little, if any, suggestive information on the number of hidden units could be obtained out of our application. Specifying the number of hidden units is difficult and yet very important [32]. If there are too few hidden units the network will not learn the map successfully, no matter what algorithm is used for the training. Having too many hidden units however can degrade the learning rate of a neural network and therefore decrease the speed with which a learned mapping is performed [39].

It seems that no theory has currently been mature enough to determine the hidden layers and hidden units systematically, although a lot of effort has been put into this problem [34]. Thus it is not unusual to determine the number of hidden units in a network purely based on experiments. We can evaluate the effect of increasing hidden units by observing the amount and the speed with which the total squared error of a network decreases. When increasing hidden units can no longer serve to make the total squared error go down substantially, it may indicate that an optimal number of hidden units has been found. Experiments tend to convince us that $2n$ hidden units are necessary for an n-input network, which seems to justify the necessity of having $(2n+1)$ hidden units for a three-layer network in the theory presented in Section 4.2.

5.3 Training and Testing of Neural Networks

Updating the weights of a neural network in order to make it learn a specified mapping is called training a neural network. After the training is stopped, testing should be carried out to evaluate the performance of the trained network.

Backpropagation networks are trained by being presented with a series of pattern pairs – each pair consisting of an input pattern and a target output pattern. Each pattern is a vector of samples from real input and output variables. The target output pattern is the desired response to the input pattern and is used to determine the error values in the network when the weights are adjusted.

The target output pattern is sometimes designed to represent a classification for the input patterns. In this way, the network may be presented with a series of input patterns together with the classification for each input pattern. In our problem, the target output is a pattern created from the samples of the IC process input variables (including process parameters and disturbances), and it is exactly this same pattern that generates the respective input pattern of the neural network through IC process simulations. In this case, the network is trained to be a pattern-mapping system. Hence the task of training is actually learning to map the output patterns of an IC process to its corresponding input patterns.

The patterns in the training set are presented to the network repeatedly. Each training iteration consists of presenting each input/output pattern pair once. When all pattern pairs in the training set have been presented, the training iteration is completed, and the next training iteration is initiated. A typical backpropagation example might entail hundreds or thousands of training iterations.

One of the common problems in function approximation by neural networks is that in most situations infinite supplies of training and testing patterns are not available. If they are, then we can train the network using the largest possible set of data and then test its performance on the largest possible set of data. However this ideal case is rarely applicable. Even if the data were available, the practical considerations may keep us from doing this. Therefore all the training and testing operations have to be based on the fact that only a modest amount of data is available.

It is crucial that the training sets are sufficiently comprehensive so that they contain essentially every possible case that could be encountered in practice. On the other hand, it is strongly desired that fewer specific examples are included in the training sets. This is due to the fact that some neural networks, notably the backpropagation networks have the capability to learn a specific data set much better than they can learn a general problem. Since the goal of most mapping network systems is to perform well in an operational setting in which the environmental inputs have more variability than is evidenced in a training set, learning the specific examples of a training set too well is not desirable. What we expect is that the neural network can generalize from the training set examples to the entire problem environment.

In this context, the term "generalize" could almost be replaced with another word "interpolate". In other words, if a real world input vector lies between or close to training set examples, then we want the output of the network to be reasonably related to the outputs it would give for the training set examples. If the input is far away from all training examples, then the output of the network cannot be expected to be meaningful [40].

The above principle also applies to the generation of testing sets of a neural network. And if the network performs well on the testing sets generated in accordance with the above principle, then the ultimate problem will be considered solved.

Although the above discussion covers only a certain aspect of network training, it is of critical importance and may also serve to explain the reason for the usage of a complicated approach in generating training sets, such as the one discussed in Section 5.2.

5.4 Convergence and Error Analysis

During the neural network's training procedure, some rules have to be employed to evaluate the training performance and also guide the training towards a correct direction. When a network is trained successfully, it produces correct answers more and more often as the training session proceeds. It is important then to have a quantitative measure of training. The total squared error is usually calculated to reflect the degree to which training has taken place in the network. The total squared error E is defined to be

$$E = \sum_p E_p = \sum_p \sum_i (t_{pi} - o_{pi})^2 \tag{5.8}$$

where the index p ranges over the set of input patterns, i ranges over the set of output units, and E_p represents the squared error on pattern p. The variable t_{pi} is the desired output or target for ith output unit when the pth pattern has been presented, and o_{pi} is the actual output of the ith output unit when pattern p has been presented. This measure reflects how close the network is to getting the

correct answers. The object of training is to find a set of weights that minimize this function.

It is quite useful to consider how the total squared error E varies as a function of any give weight in the system. In the case of the simple single-layered linear system, a smooth error function such as the one shown in Figure 5.4 could often illustrate the dependence between overall error and changes in a single weight in the network [40].

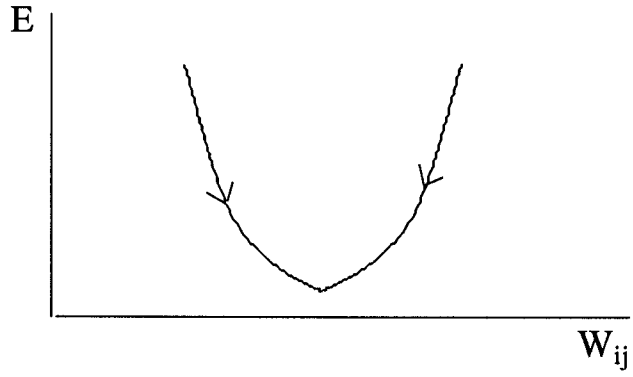


Figure 5.5: Function of error vs. weights

The well-known least mean square (LMS) procedure finds the values of all of the weights that minimize this function using a method called gradient descent. That is, after each pattern has been presented, the error on that pattern is computed and each weight is moved down the error gradient toward its minimum value for that pattern. Since we cannot map out the entire error function on each pattern presentation, we must find a simple procedure to determine, for each weight, how much to increase or decrease each weight. The idea of gradient descent is to make a change in the weight proportional to the negative of the derivative of the error, as measured on the current pattern, with respect to each weight. Thus the training rule becomes

$$\Delta W_{ij} = -k\frac{\partial E_p}{\partial W_{ij}} \tag{5.9}$$

where k is the constant of proportionality. Substituting in the pattern squared error $E_p = (t_{pi} - o_{pi})^2$ defined in (5.8) and carrying out the derivative of the error measure in equation (5.8) we get

$$\Delta W_{ij} = \epsilon \delta_{pi} i_{pj} \tag{5.10}$$

where $\epsilon = 2k$ and $\delta_{pi} = t_{pi} - o_{pi}$ is the difference between the target for unit i on pattern p and the actual output produced by the network. i_{pi} is the activation or output value of the originating unit j that connects to output unit i. This weight adjustment equation is the well-known generalized δ rule.

If we change each weight according to this rule, then each weight is moved toward its own minimum and we consider the system as moving downhill in the weight space until it reaches its minimum error value. When all of the weights have reached their minimum points, the system will reach zero error and the weights will no longer be modified. This process is referred to as convergence.

However convergence is not always easy to achieve because the process may take an exceedingly long time and sometimes the network gets stuck in a local minimum and stops learning altogether. In this case we consider the system to be unable to get the problem exactly right; the best it can achieve is to find a set of weights that produces as small an error as possible.

The most frequent reason that keeps a network from converging is local minima. When a network gets stuck in a local minimum, it can not use the gradient descent method to escape from it and proceed for a global minimum. A study has indicated that the number of hidden units really has something to do with

the appearance of local minima [40]. In networks with many hidden units, local minima seem quite rare, while with fewer hidden units, local minima can be more common. However an unlimited increment of hidden units cannot be beneficial, because this will significantly complicate the system and make the training process extremely long. Therefore a tradeoff has to be made in determining the hidden units of a neural network.

A number of approaches have been devised to solve the local minimum problem. The well-studied methods include simulated annealing [41], statistical thermodynamics [42] and the random optimization method [43]. Although these approaches theoretically prove to be efficient in dealing with the local minimum problem, these is still more work to be done before putting them into practice. A much simpler method is employed in this thesis to get the network out of a local minimum [31]. This method is based on the idea that adding small random values to the weights allows the network to escape from a local minimum encountered by moving the position of the network from a local minimum to a random point some distance away. If the new position is sufficiently removed from the valley of the local minimum, then convergence may proceed in a new direction without getting stuck in the same local minimum again. The amount of the random value or noise required depends on the local "landscape", which is typically unknown to the investigator. Thus, there is some degree of luck involved in getting a network out of a local minimum. Although this method is short of assurance and efficiency, its simplicity and feasibility contributes a lot to its wide spread application in backpropagation neural network training.

Several important aspects on the realization of the neural network approach have been addressed in this chapter. In the next chapter, a detailed discussion

on the real experiments and some experimental results will be provided to show the effectiveness of this method.

Chapter 6

Design of Experiments and Result Analysis

6.1 Introduction

In the last five chapters, an introduction and some relevant background information about IC process monitoring were presented. And a methodology was developed to implement real time monitoring and surveillance of disturbances in the IC process using statistical process control techniques. As an important part of the methodology, backpropagation neural networks were introduced, and a discussion on a backpropagation network's approximation capability of functional mapping was provided.

The purpose of the present chapter is to provide some examples about the design of experiments, with a focus on the sample size selection, control limit determination, thresholding and the generation of training and testing patterns. Finally some results of the experiments will be provided, which is followed by an error analysis.

6.2 Design of Experiments

In this section more detailed descriptions about several important aspects in design of our experiments will be presented in sequence.

6.2.1 Selection of Parameters to be Monitored

As mentioned earlier, in order to increase the efficiency of IC process surveillance and simplify the task, only those "key" process parameters and disturbances that have a significant effect on process output are selected for monitoring. For all the examples presented in this chapter, we will use a simple p-mos transistor model as the fabrication sample. The employed disturbances and process output parameters are listed in Table 6.1 and Table 6.2.

The selection of these process disturbances is based on the results of sensitivity analysis. The experiments of sensitivity analysis are carried out in a simplistic way as mentioned earlier. Among all the process parameters and disturbances (nearly 40 excluding the parameters of narrow channel and short channel effects), one parameter is selected each time to be shifted from its nominal value by a certain amount, say 3σ, while keeping all other parameters at their nominal values. If any of the output parameters being observed shows a significant shift from its nominal values, or goes beyond the predetermined control limits, then this unusual shift will be attributed to the change of the process disturbance we shifted. And this process disturbance will be taken as one of the process disturbances that have significant effects on process output, and thus falls into the set of process disturbances to be monitored. If no output parameter shows a significant shift, on the other hand, the disturbance that has currently been

Order	Disturbance	Nominal Mean	Nominal Standard Deviation
1	ΔW_p (nitride mask)	2.605e-04	7.815e-06
2	ΔL_p (poly mask)	6.440e-05	1.932e-06
3	Segcoefboron	2.000e-03	6.000e-05
4	Diffoxboron	1.000e-01	3.000e-03
5	Mlatarsen	3.000e-01	9.000e-03
6	Diffphos	2.641e+01	7.923e-02
7	Parabolicwet	6.250e-02	1.875e-03
8	Speconres	1.400e+02	4.200e+00
9	Q_{ss}	7.489e+10	2.247e+09
10	Coefucrit	3.335e+03	1.000e+02

Table 6.1: Selected disturbances and their nominal values

Order	Name	Nominal Mean	Order	Name	Nominal Mean
1	$L[m]$	2.354e-06	**2**	$W[m]$	8.439e-05
3	VT0 [m]	0.000e-00	**4**	KP [A/V^2]	2.164e-05
5	GAMMA [$V^{1/2}$]	5.009e-01	**6**	PHI [v]	6.940e-01
7	PB [V]	9.201e-01	**8**	CGSO [F/m]	2.868e-10
9	CGDO [F/m]	2.868e-10	**10**	CGBO [F/m]	4.472e-10
11	RSH [$\Omega/\square$]	6.437e+01	**12**	CJ [F/m^2]	2.997e-04
13	MJ	5.000e-01	**14**	CJSW [F/m]	2.997e-06
15	MJSW	5.000e-01	**16**	JS [A/m^2]	4.476e-03
17	TOX [m]	2.809e-08	**18**	NSUB [/cm^3]	1.023e+16
19	NSS [/cm^2]	7.479e+10	**20**	NFS [/cm^2]	8.019e+10
21	XJ [m]	2.945e-07	**22**	LD [m]	2.394e-07
23	U0 [cm^2/Vs]	1.806e+02	**24**	UCRIT [V/cm]	0.000e+00

Table 6.2: Selected measurable output parameters

shifted will be considered as a insignificant disturbance and no surveillance is necessary for it.

To precisely define the impact of a process disturbance upon the outputs, we must resort to statistical analysis. The analysis below is based on the assumption that the random variable under study is normally distributed. Since all process disturbances we are studying obey normal or approximately normal distributions, the theory can be readily applied to our problem.

First let us take a look at the difference a non-shifted and a shifted disturbance on its correlated outputs. This is actually to test the hypothesis that the mean μ of an output when its correlated disturbance has been shifted equals its nominal value μ_0, i.e.

$$H_0 : \mu = \mu_0 \tag{6.1}$$

$$H_1 : \mu \neq \mu_0 \tag{6.2}$$

where μ_0 is the mean of an output when its correlated disturbance has been kept at its nominal.

As σ^2 is unknown, it is estimated by sample variance S^2. The test statistic is

$$t_0 = \frac{\bar{x} - \mu_0}{S/\sqrt{n}} \tag{6.3}$$

The null hypothesis $H_0 : \mu = \mu_0$ will be rejected, or the hypothesis $H_1 : \mu \neq \mu_0$ will hold true if $|t_0| > t_{\alpha/2,n-1}$, where $t_{\alpha/2,n-1}$ denotes the upper $\alpha/2$ percentage point of the t distribution with $n-1$ degree of freedom. A typical example is given below.

Sample	Sheet Resistance (RSH) [Ω/□]
1	64.121
2	64.093
3	64.125
4	64.153
5	64.127
6	64.116
7	64.126
8	64.075
9	64.097
10	64.095
$\bar{x}$	64.1128
S	2.2622e-2

Table 6.3: Samples of RSH when the correlated disturbance No. 4 shifted to 3σ

Using equation (6.1) we get:

$$t_0 = \frac{\bar{x} - \mu_0}{S/\sqrt{n}} = \frac{64.1128 - 64.3706}{0.0226/\sqrt{10}} = -36.0387 \tag{6.4}$$

$$|t_0| > t_{0.0005,9} \tag{6.5}$$

where $\alpha = 0.001$, so the hypothesis $H_1 : \mu \neq \mu_0$ holds true with $100(1-\alpha)\% = 99.9\%$ confidence.

We can further compute the magnitude of this difference using following equations.

$$\bar{x_1} - \bar{x_0} - t_{\alpha/2,n_1+n_0-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_0}} \leq \mu_1 - \mu_0 \leq \bar{x_1} - \bar{x_0} + t_{\alpha/2,n_1+n_0-2} S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_0}} \tag{6.6}$$

$$S_p{}^2 = \frac{(n_1 - 1){S_1}^2 + (n_0 - 1){S_0}^2}{n_1 + n_0 - 2} \tag{6.7}$$

Table 6.4 displays two sets of samples of the sheet resistance that are obtained when its correlated disturbance boron diffusivity is kept at nominal and is shifted to $\mu - 3\sigma$, respectively.

Using equation (6.6) (6.7) we can get the difference of the means of RSH at these two cases and the confidence, $\alpha = 0.001$. The results are shown below.

$$\bar{x_1} - \bar{x_0} - t_{0.0005,18} S_p \sqrt{\frac{1}{10} + \frac{1}{10}} \leq \mu_1 - \mu_0 \leq \bar{x_1} - \bar{x_0} + t_{0.0005,18} S_p \sqrt{\frac{1}{10} + \frac{1}{10}} \tag{6.8}$$

$$S_p = \sqrt{\frac{S_0^2 + S_1^2}{2}} = 3.0263 \times 10^{-3} \tag{6.9}$$

Sample	RSH (nominal)	RSH (shifted)
1	64.417	64.631
2	64.390	64.670
3	64.362	64.660
4	64.376	64.584
5	64.413	64.654
6	64.346	64.636
7	64.360	64.645
8	64.439	64.612
9	64.402	64.658
10	64.399	64.696
$\bar{x}$	64.3904	64.6446
S^2	8.5538e-6	9.7627e-6

Table 6.4: Samples of RSH when the correlated disturbance No. 4 at nominal and shifted to -3σ

$$64.6446 - 64.3904 - 3.0263x10^{-3}\sqrt{\frac{1}{5}} \leq \mu_1 - \mu_0 \leq$$

$$64.6446 - 64.3904 + 3.0263x10^{-3}\sqrt{\frac{1}{5}} \tag{6.10}$$

$$0.2489 \leq \mu_1 - \mu_0 \leq 0.2595 \tag{6.11}$$

The difference between the value of $\bar{x}$ for the shifted disturbance, T_4, and the centerline CL_4 is $T_4 - CL_4 = 0.2091$, so a shifted disturbance will produce a sample point that falls outside the threshold with more than $100(1-\alpha)\% = 99.9\%$ confidence.

Another critically important issue in our algorithm is fault observability. In other words, if we want the mapping between the disturbances and outputs to be one-to-one, we have to make sure that the disturbances are distinguishable and observable. In order to eliminate those undistinguishable disturbances from the set of the disturbances to be monitored, a specific algorithm must be used. The basic steps of this algorithm are explained as following.

Step 1 : Let the standardized normal disturbances be $d_1, d_2, \cdots, d_n$ and the standardized normal outputs be $y_1, y_2, \cdots, y_m$. Shift the ith disturbance d_i by 3σ and find the shift in the jth output, y_j. And compute the significance level, s_{ij}, that y_j is different than its nominal. If $s_{ij} > T$, a prescribed threshold, then set:

$$f_{ij} = \begin{cases} 1 & \text{if } y_j \text{ is increased} \\ -1 & \text{if } y_j \text{ is decreased} \\ 0 & \text{if } s_{ij} < T \end{cases}$$

where f_{ij} is an entry of a $n \times m$ matrix F, called the fault matrix.

Step 2 : For each disturbance, $d_1, d_2, \cdots, d_n$, compute $\sum_{j=1}^{m} |f_{ij}|$. If

$$\sum_{j=1}^{m} |f_{ij}| = 0$$

then the ith disturbance, d_i is unobservable, given the measurement set. Eliminate d_i from the set of disturbances to be monitored or add additional measurements.

Step 3 : Find out if there are two rows that are the same in the fault matrix. First compute:

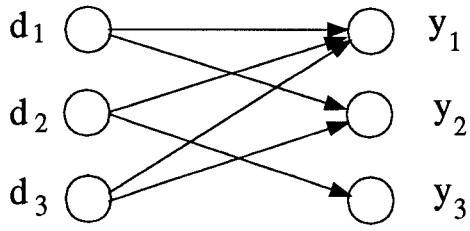
$$r_{ij} = \sum_{k=1}^{m} |f_{ik} - f_{jk}|$$

If $r_{ij} = 0$, then rows i and j in the fault matrix are identical. This means that disturbances d_i and d_j are not guaranteed to be uniquely diagnosable, and are said to be in the same ambiguity group.

A simple example is given in Figure 6.1 to show how this algorithm works. In this example disturbance d_1 has significant impact on output y_1 and y_2, so does d_3. While disturbance d_2 affects y_1 and y_3.

Suppose d_1 and d_3 affect y_1 and y_2 in a similar way, then the fault matrix as shown in Figure 6.1 can be obtained. By comparing the rows in this matrix, we find that row 1 and row 3 are identical. Therefore the disturbances d_1 and d_3 are in the same ambiguity group according to the algorithm. We conclude that the disturbances d_1 and d_3 are not distinguishable.

We can use this algorithm to find out if a set of measurements are sufficient to detect significant shifts in the set of disturbances that need to be diagnosed.



$$\begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \end{array} \begin{pmatrix} 1 & -1 & 0 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

fault matrix

Figure 6.1: An example showing the procedure of determining the disturbances in the same ambiguity group

6.2.2 Generation of Thresholds

When performing sensitivity analysis, we also keep track of the output parameters that are significantly influenced by a certain disturbance, and the mean values of their shifts. These mean values will be used as the multi-thresholds of an output parameter. Therefore if an output parameter is affected by two disturbances, then there will be two thresholds associated with it.

This procedure is figuratively described in Figure 6.2. Figure 6.2(a) shows the corresponding relations between the thresholds and the control limits of the disturbances, and Figure 6.2(b) shows two different corresponding situations of the disturbances and the output.

In practice each threshold is calculated by first shifting a specific disturbance within a certain range around a control limit, for example 3σ, so the range can be 3σ-0.5σ to 3σ+0.5σ. Each shifted value of the disturbance, along with other disturbances (nominal values) are used as the input to a statistical simulator, and generally 10 to 20 sets of simulations are performed to get mean values for some output parameters. After the whole range of a shift has been covered, the mean values obtained for each output parameter will be employed to compute a threshold of this parameter with respect to the specific disturbance that has been shifted. It should be emphasized that only one disturbance will be shifted in a major way at a time while keeping all others at nominals so that additive effects are minimized.

The objective of establishing the thresholds is to identify the disturbance that is currently affecting the output parameter of interest among all potential disturbances that could affect this parameter. This approach works based on a well-observed fact that no two disturbances tend to affect an output parameter

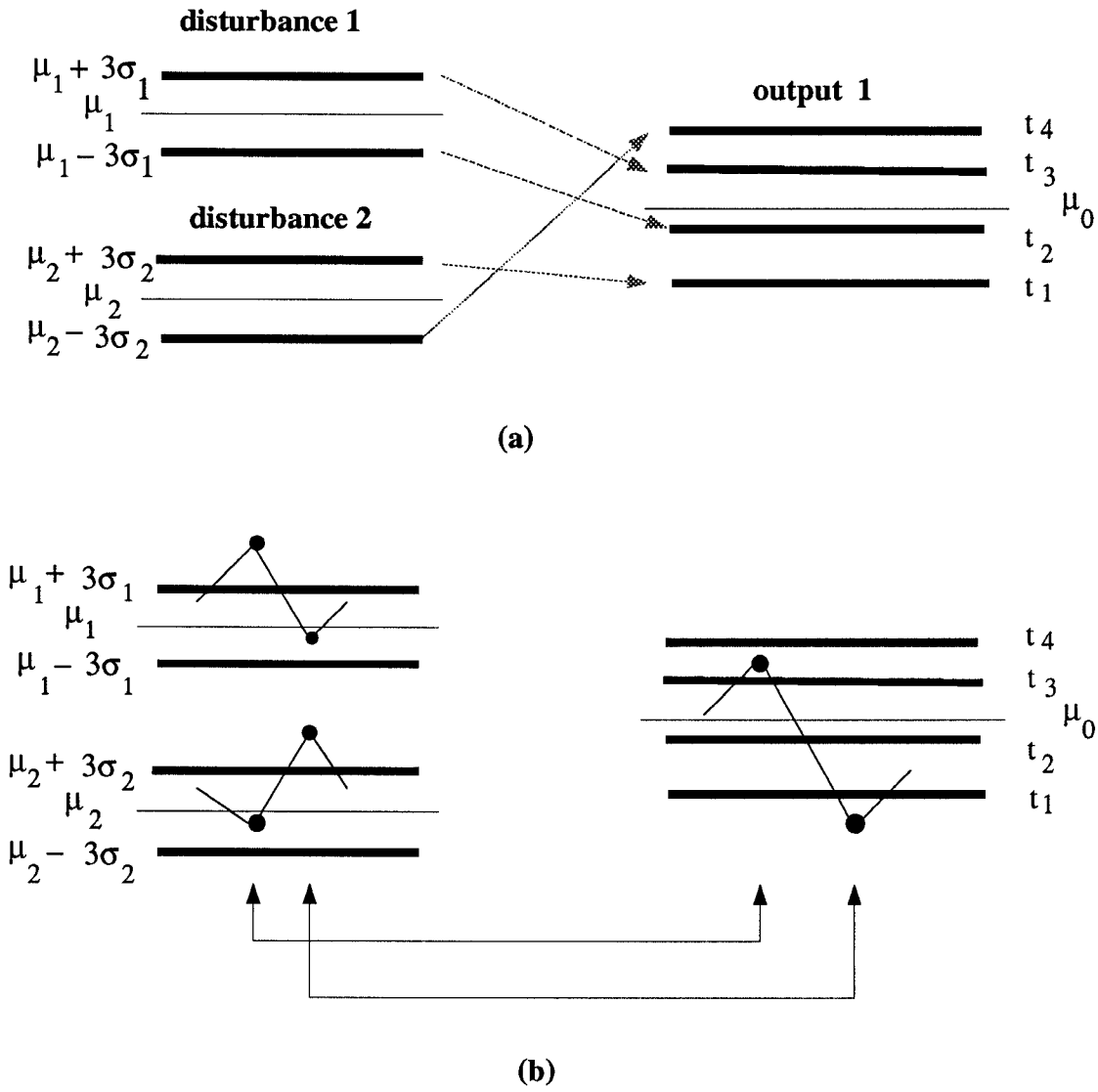


Figure 6.2: Procedure of creating thresholds for an output parameter affected by two disturbances

in a same manner. In other words, a specific disturbance will affect a unique set of output parameters by a certain magnitude based on the relations of physical properties between them. The thresholds are set up to classify the magnitude by which an output parameter has been influenced. By observing the set of output parameters that have been significantly affected and the magnitude by which they are shifted, we can identify a disturbance whose shift can be attributed to the shifts in the output parameters.

To facilitate the learning of the neural networks, the value of an output parameter, or a point falling within a specific pair of thresholds is assigned a specific value, called a code. Thus a code is an indication of within which two thresholds a point falls. A set of these codes for each output parameter creates a unique pattern. A similar task is done for each disturbance. In this thesis three different codes have been used for each disturbance, which corresponds to the situation when a disturbance lies within the control limits, beyond the upper control limit or beyond the lower limit, respectively. An example is given in Figure 6.3. In Figure 6.3, disturbance 1 and 2 are encoded as a_2 and a_0 because of the ranges they fall in. Due to the significant shifts of disturbance 1 and 2, output 1 falls between the thresholds t_m and t_n, thus a code C_{mn} is assigned to output 1. Similarly C_{ij} is assigned to output 2. By doing so, a unique pair of input-output patterns $[a_2 a_0]$, $[C_{mn} C_{ij}]$ is created.

When the manufacturing processes for producing a special device has been determined, it is reasonable to assume that for a specific disturbance in these processes the output parameters are always affected in basically the same way. And generally only one disturbance goes out-of-control at a time. Under this assumption, we can prove that a certain set of codes for outputs corresponds

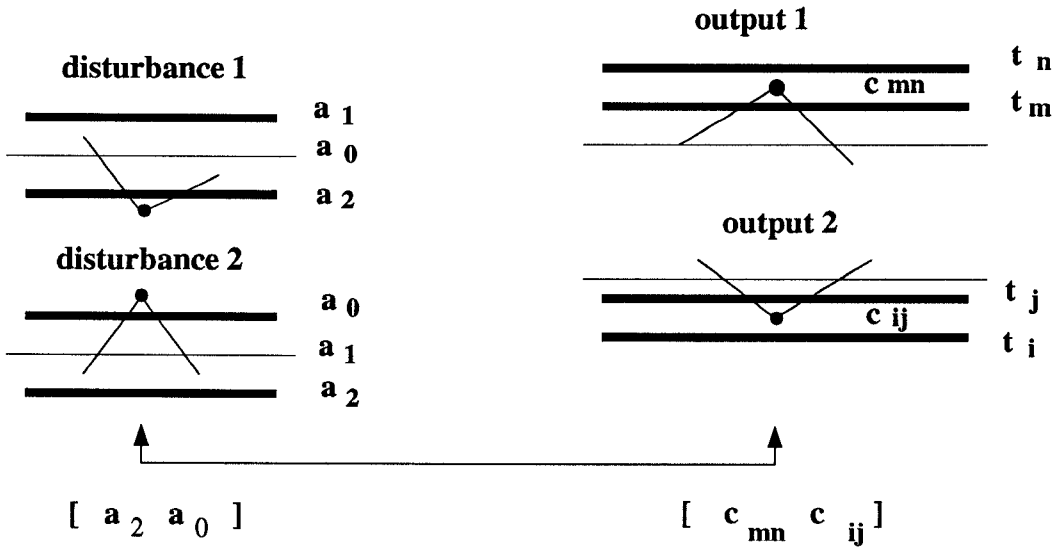


Figure 6.3: An example of the generation of codes for disturbances and outputs

with a high probability to a unique set of codes for the disturbances if no two disturbances in the same ambiguity group are selected . From mathematical point of view, there is a high probability that the coding of process variables (disturbances and output parameters) transforms the mapping between disturbances and output parameters to an one-to-one map. This serves as the basis on which the neural networks can be employed to implement the inverse mapping between them.

6.2.3 Determination of Control Limits for the Control Charts

As discussed in Section 3.3, the determination of control limits is an important step in using the control chart technique. If the control limits are selected to be close to the central line, the probability of a "false alarm" occurrence will become larger. When the control limits are selected far away from the central

line, on the other hand, the occurrence of the other error, i.e. "fail to detect" error, will be more frequent.

First let us talk about the control limits for process disturbances. Since most of the process disturbances can be modeled as random variables obeying a normal distribution, the "σ limit" convention can be applied to them. Although the 3σ limits are widely used control limits and also extensively employed in this thesis, other control limits can be used without difficulty in the methodology presented in this thesis. The control limits for the process disturbances can be determined either from statistical estimation or from practical experiments. For example, generally we set the control limits of all process disturbances at their 3σ level. This means that only when some disturbances depart as far as 3σ from their nominal values in a non-random pattern, can attention be paid to them. A search for an assignable cause would be initiated in this case and some measures will be taken to bring them back within control limits if an assignable cause has been found.

In some fabrication processes which are designed to produce the devices that need to meet special (usually more strict) requirements, however, the parameters for the devices manufactured may not be acceptable well before the disturbances go beyond the predetermined 3σ limits. In this case, the width between the control limits for the disturbances should be narrower than 6σ, from -3σ to $+3\sigma$. The appropriate choices could be 2σ, 1σ or determined by practical experience. Once the control limits for the process disturbances have been determined, the control limits for the output parameters can be determined accordingly.

To set up the control limits for output parameters, first the center lines, or mean values for each parameter have to be determined. This can be accomplished

through a large number of simulations with all process disturbances kept at their nominal values. Actually these mean values were obtained through 1000 to 2000 simulations in the experiments. We can also make use of the results in these simulations to compute the standard deviations, or σ values for each parameter.

If we use these σ values to set up control limits for the output parameters, then it will be difficult to distinguish which disturbances are potential assignable causes for the occurrence of out-of-control signals in the control charts. And it would be extremely difficult or even impossible for the neural networks to learn the mapping between the output parameters and the disturbances. The thresholding approach introduced in Section 6.2.2 has been devised to alleviate this difficulty and enhance the "resolution" of the mapping.

6.2.4 Sample Size Analysis

As a final topic of experimental design, the selection of sample size is briefly analyzed in this section.

By sample size, we mean that the numbers of simulations, or experiments that need to run to get a sample point in the control charts. The sample size is actually a mean of the result in several simulations. In statistical quality control, the chart we are using in this thesis is called the $\bar{x}$ chart. The $\bar{x}$ chart monitors the average quality level in the process. Therefore samples should be selected in such a way that maximizes the chances for shifts in the process average to occur between samples, and thus to show up as out-of-control points on the $\bar{x}$ chart.

Although it is not possible to give an exact solution to the problems of sample size selection, determination of control limits and frequency of sampling, some guidelines coming from experience prove to be very helpful in the design of

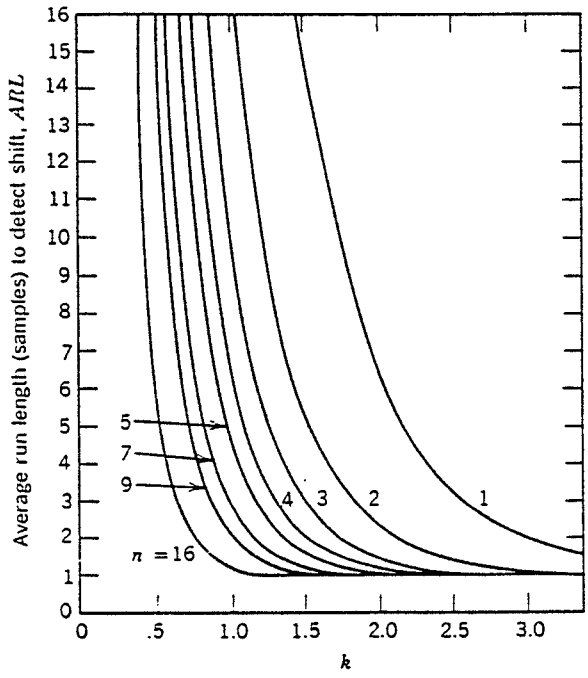


Figure 6.4: Average run length (sample) for the $\bar{x}$ chart with 3σ limits, where the process mean shifts by $k\sigma$. (Adapted from *Introduction to Statistical Quality Control.* by D.C.Montgomery, John Wiley & Sons, pp. 230, 1991)

control chart.

One way to approach the decisions regarding sample size and sampling frequency is through the average run length (ARL) of the control chart. Essentially, the ARL is the average number of points that must be plotted before a point indicates an out-of-control condition. The ARL for the $\bar{x}$ chart can be expressed as:

$$ARL = \frac{1}{p} \tag{6.12}$$

where p is the probability that any point exceeds the control limits. Figure 6.4

displays these ARL curves for sample size of $n = 1, 2, 3, 4, 5, 7, 9$, and 16 for the $\bar{x}$ control chart, where ARL is in terms of the expected number of samples taken in order to detect the shift [11]. It can be seen from Figure 6.4 that if we wish to detect a shift of 1.5σ using a sample size of $n = 3$, then the average number of samples required will be 3, and we could reduce the ARL to approximately 1 if we increase the sample size to $n = 16$.

Basically if the $\bar{x}$ chart is being used primarily to detect moderate to large shifts, for example on the order of 2σ or larger, then relatively small samples of size 4, 5 or 6 are reasonably effective. On the other hand, if we are trying to detect small shifts, then larger sample sizes of possible 15 to 25 are needed. This guideline has been employed in our experiments. A detailed analysis on the sample size selection can be found in [11]

6.3 Experimental Results

6.3.1 Monitoring IC Process Parameters Using Control Charts

Statistical control charts play an important role in our approach of monitoring the disturbances of the IC process. In this thesis control charts have been extensively used in sensitivity analysis and in the determination of correlations between the disturbances and the output parameters. Some of the results from the experiments are displayed below to show the correlation effects between certain disturbances and outputs of a IC process.

This is perhaps one of the simplest correlations that has been found in our experiments. It is actually a one-to-one correlation between ΔW_p (dist.No.2) and

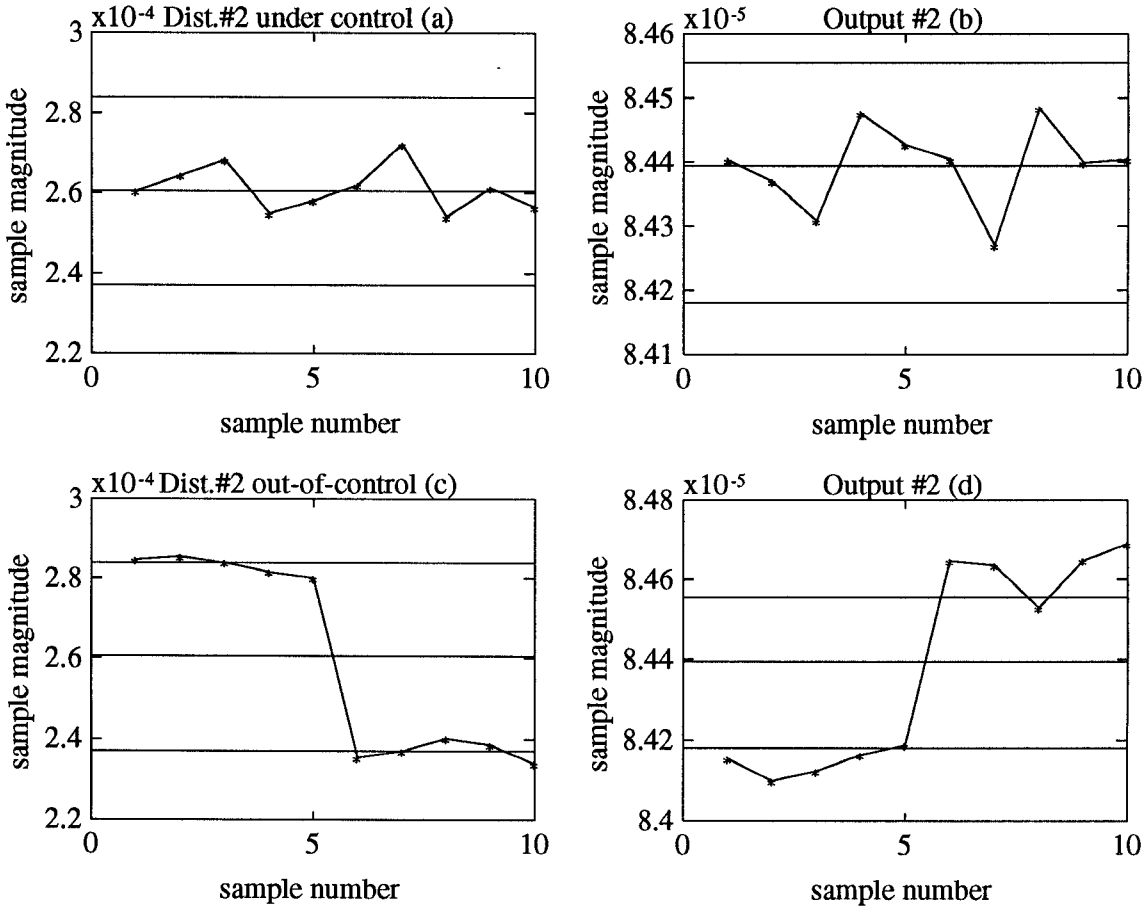


Figure 6.5: Control charts for correlated disturbance and output parameter

channel width W (output parameter No.2). Figure 6.5(a) shows the control chart for ΔW_p, which varies around its nominal value. The corresponding behavior of channel width W (output No.2), which is the only observed correlated parameter with ΔW_p, has been drawn in Figure 6.5(b). It can seen that it doesn't show any significant shift as expected. When ΔW_p is shifted beyond its control limits, as shown in Figure 6.5(c), the channel width W has a significant shift from its nominal value as indicated in Figure 6.5(d).

For comparison, the control charts for another output parameter, channel length L (output No.1) in these two cases are shown in Figure 6.6. We can see that channel length L does not show a significant shift in both cases.

It is worth mentioning that what we use as input and output variables are actually the statistics of the process disturbances and output parameters. Therefore given the same set of inputs, different sets of outputs may result at different sampling points. This accounts for the reason that the channel length still varies, although none of its correlated disturbances has shifted in both cases.

Note that the control limits in the output control charts have been determined with hundreds of simulations using correlated disturbances shifting around their critical values (μ+/-3σ in the above experiments).

6.3.2 Distribution of the Sampling Density in Pattern Generation

As discussed in Section 5.2, if more samples of disturbances in the training patterns are taken near their control limits, the learning of neural networks will be more efficient. Based on this reasoning a special distribution has been created to govern the sampling density. A typical distribution is the superposition of two

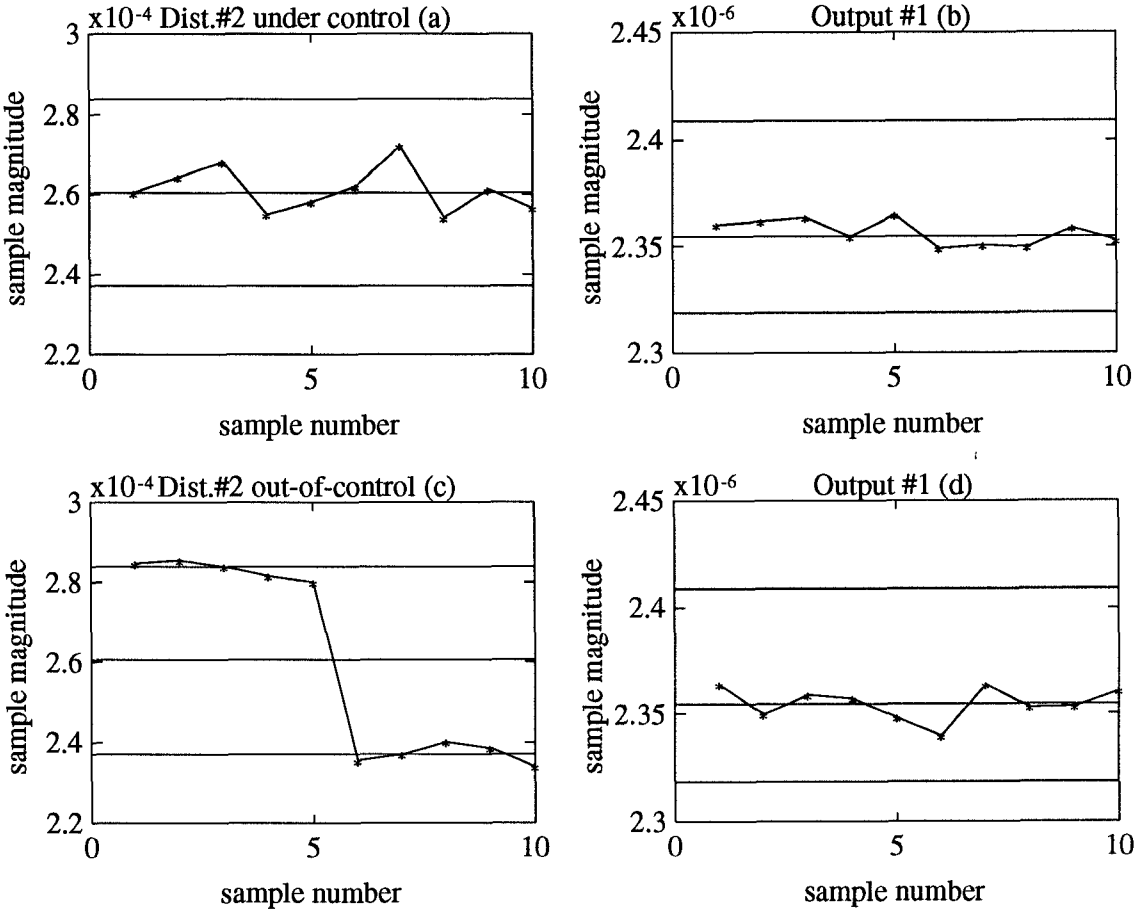


Figure 6.6: Control charts for uncorrelated disturbance and output parameter

Gaussian distribution functions. An example is shown Figure 6.7.

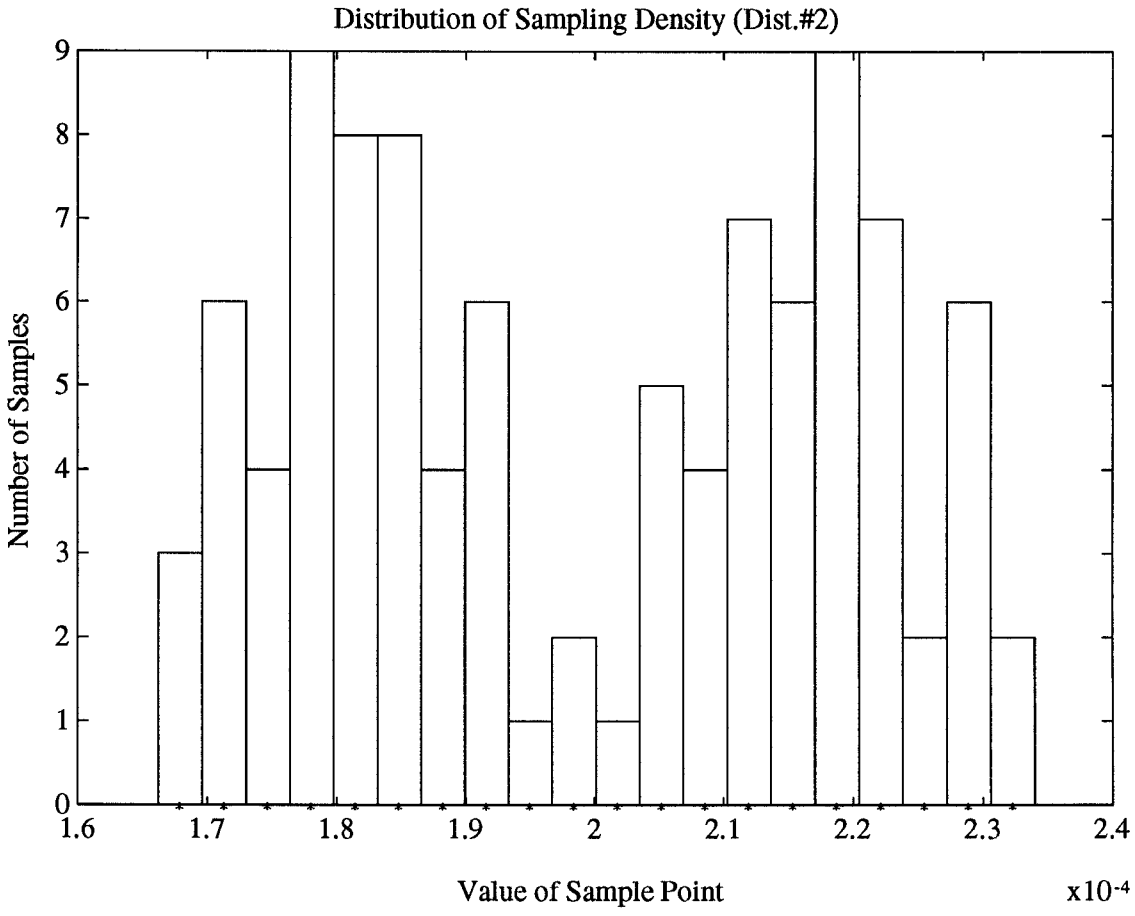


Figure 6.7: A typical distribution function of sampling density

6.3.3 Testing Results for the Neural Network

After a successful training of the neural network has been completed, the weights of this neural network are saved and are tested by the testing patterns. The testing patterns are generated in a quite similar way as the training patterns. In the following experiments 20 sets of testing patterns are collected, and each set

consists of 10 patterns.

The testing results are expressed in terms of number of matches between the output of neural network and the target values. An example is given in Figure 6.8 to illustrate the testing procedure.

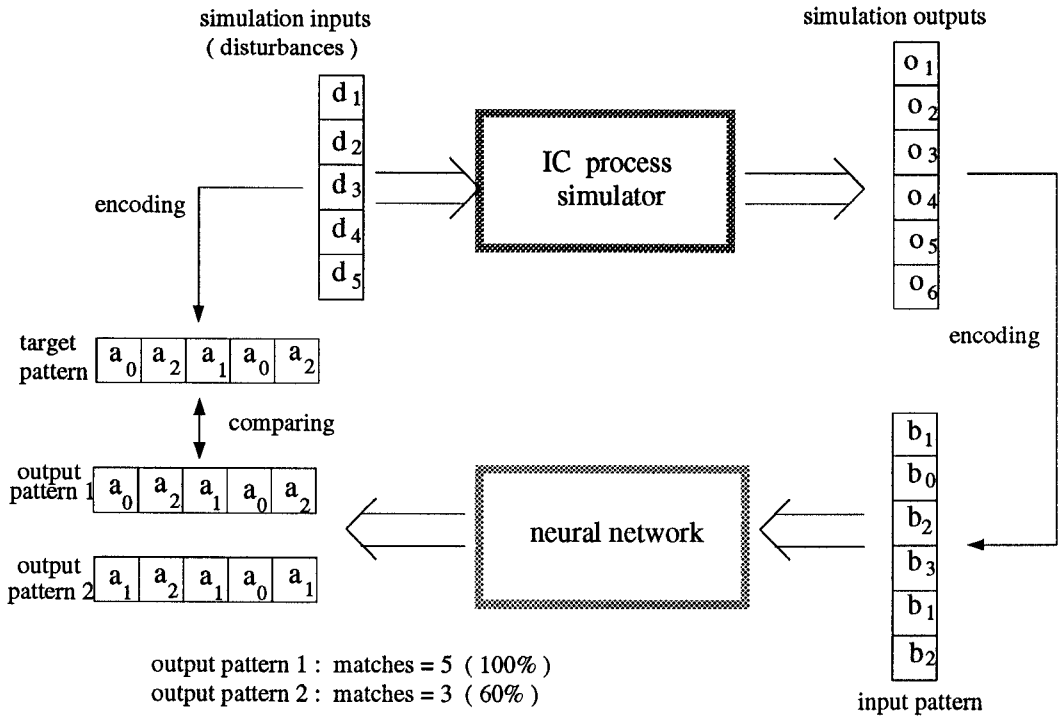


Figure 6.8: The testing procedure

A match occurs only when the output of a specific unit in the neural network's output layer is exactly same as its target value. This means that the situation of this specific disturbance, namely between control limits or beyond upper or lower control limits, has been correctly detected.

Figure 6.9 shows the average match rate of each disturbance, and the rates are obtained from 20 sets of testing patterns.

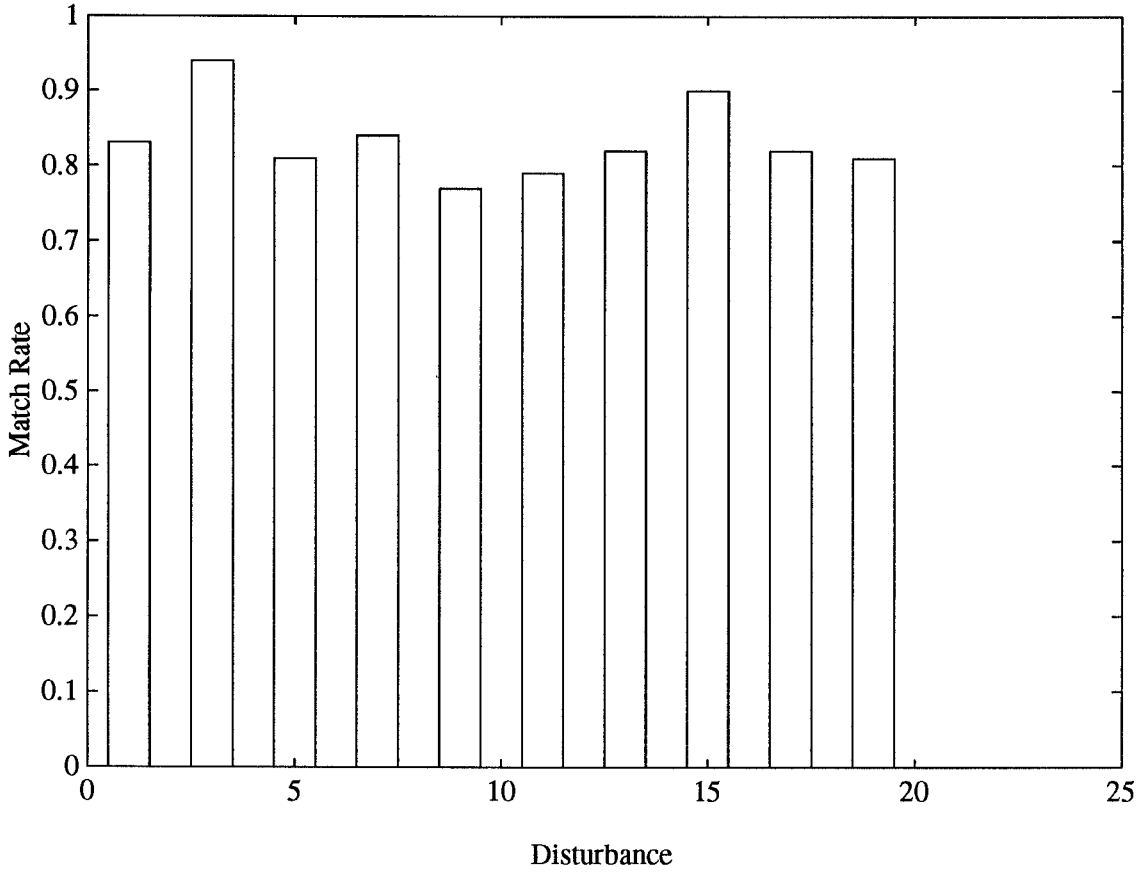


Figure 6.9: The average match rate for each disturbance

We can see that the average match rate for each disturbance differs slightly. Experience tends to indicate that the higher degree of correlation a disturbance has, in other words, the more output parameters a disturbance significantly affects, the lower average match rate it can achieve. The reason can be traced to the errors involved in the simulations, the thresholding and the generation of training and testing patterns.

In order to inspect the general match rate for all experiments, the average match rate for each set of testing patterns has been plotted in Figure 6.10.

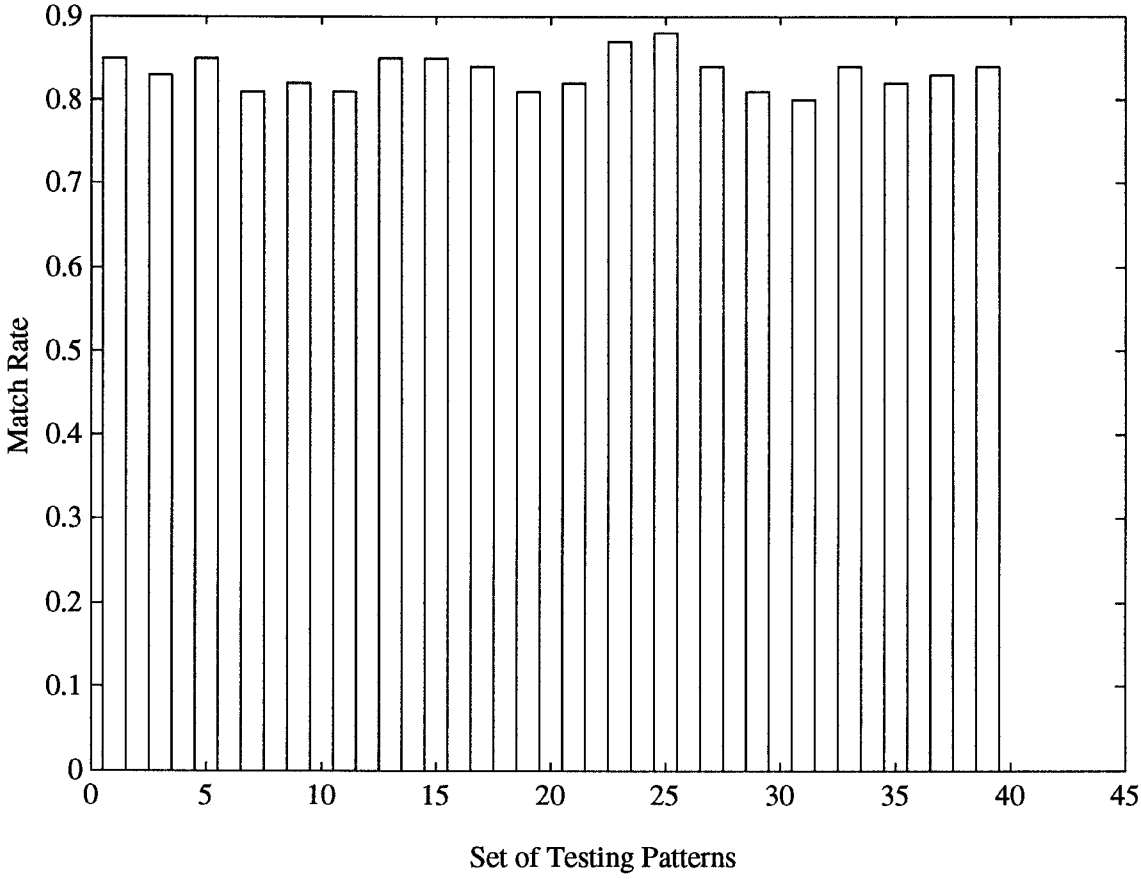


Figure 6.10: The average match rate for each set of testing patterns

Figure 6.10 indicates that the average match rates for all sets of testing patterns are basically identical. This is normal because the testing patterns have been generated randomly. It can be easily computed from the above results that the total average match rate, or the rate of correctly detecting the shifts of the process disturbances using the algorithm presented in this thesis is approximately 83%.

6.4 Error Analysis of the Experiments

Although we always expect experimental results to be exactly what we want them to be, it is often not the case in practice. The reason lies in the fact that there always exist some experimental errors associated with each step in the experiments. It is impossible to remove all of these errors in most cases. The best we can do is to keep them as small as possible. This section focuses on an analysis of the experimental errors using the algorithm presented in this thesis and the possible reasons behind them.

The simulation accomplished by a simulator like FABRICS is a stochastic process, in which the process conditional parameters vary in a random fashion. In other words, even if exactly the same set of inputs are used, the simulations will yield different results in different experiments. This is why the mean of the output results from a set of 10 to 20 simulations is used as an output sample, since it doesn't make a lot of sense to use the result from only one simulation as a sample point. Moreover, in order to minimize the errors inherent in any stochastic process, each threshold of an output parameter is created using a set of 20 points sampled around the control limit of a specific disturbance. The

threshold created in this way is more accurate because it is the mean of a set of output samples rather than just one point.

In the following context the term "good pattern" will be used to refer to a pattern that results from a correct mapping, and the "bad pattern" is referred to a pattern that is generated with the random interference in simulations, and hence corresponds to an incorrect mapping. An example is given in Figure 6.11 to explain how "bad patterns" are created.

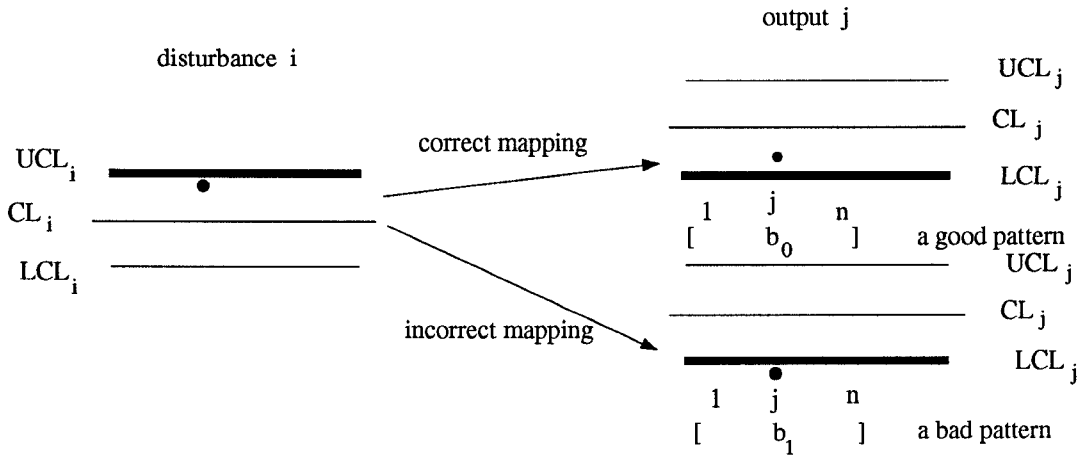


Figure 6.11: Creation of "good pattern " and "bad pattern"

Figure 6.12 displays an approximated probability distribution of obtaining correct results, or good patterns for each input-output pair.

The probability of obtaining a set of good patterns is simply the superposition of the probabilities of all those output parameters under observation. The curve indicates that the probability to generate a good pattern is rather large when samples are collected in the area far away from the control limits. On the other hand, the probability is smaller when sampling is performed near the control limits. This situation is made worse when the distribution of the sampling

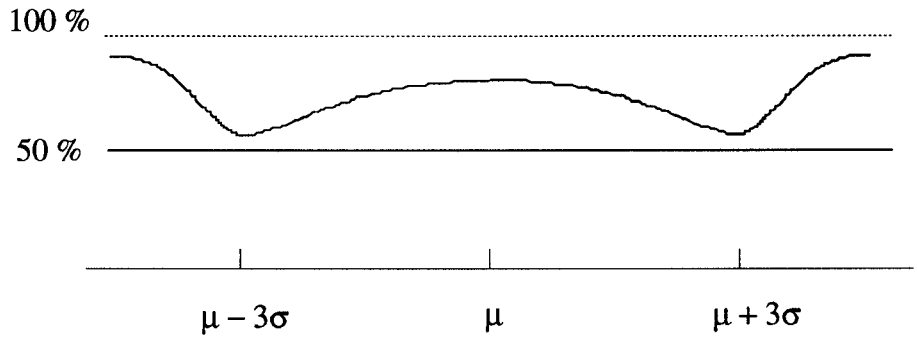


Figure 6.12: An approximated probability distribution of yielding "good" patterns

density shown in Figure 5.4 has to be used because of the reasons discussed in Section 5.2.

To diminish the undesired effect caused by the sampling density distribution, the two means in the distribution are designed not to lie at the control limits. For example, if the control limits are μ+/-3σ, the means in the distribution may be selected as μ+/-2.5σ, as shown in Figure 6.13.

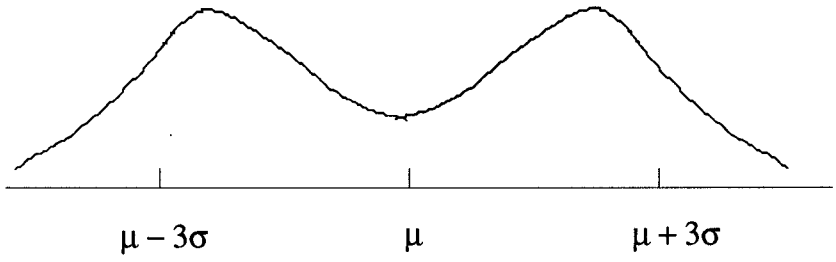


Figure 6.13: Probability distribution of sampling density

Besides the factors discussed above that can adversely affect the pattern generation, another factor, the "false alarm" error or the α error that exists in every statistical process can also contribute to the generation of the "bad patterns".

As introduced above, the α error is associated with the fact that even if no significant variation happens to the process disturbances, every so often there will be some samples on control charts, which show out-of-control signals. The probability that an α error occurs ranges from 0.0028 to 0.0033 in our experiments. This probability is small compared with the probability discussed earlier. But since there is no way to distinguish these "false alarms" from those "true alarms", the output will be encoded in the same way as in the case where no "false alarm" occurs.

The direct impact of these factors upon the experiments is the generation of some wrong training patterns that can mislead the training of the neural network. These patterns are not helpful when the neural network is learning the mapping, and often make it more complicated.

It goes without saying that the more "bad patterns" in the training sets, the more poorly a neural network can learn the mapping. It was observed in this case that the neural network either does not converge or converges at an extremely slow pace. Even if the neural network does converge, or the final total squared error of the training is smaller than the predetermined value, it may not learn the mapping accurately, or as accurately as usual, which is reflected by a much lower rate of matching.

One plausible method for improving the training of a neural network is to devise an algorithm to pick out those "bad patterns" before the patterns are used as training sets. The difficulty that comes with this kind of algorithm lies in the intensive computations involved, because of the large number of training patterns and the complicated correlations existing between the process disturbances and output parameters.

Chapter 7

Future Work

7.1 Summary

Up to this point, we have discussed the background, development, implementation and testing results of our approach for the surveillance and diagnosis of the statistical parameters in an IC manufacturing process. Although we have used a specific device model and a specific set of process parameters in the testing experiments, the algorithm can be readily applied to other complicated device models and more process parameters. In this last chapter, a brief summary and discussion on the drawbacks of our algorithm and what needs to be done in improving it will be presented.

Statistical process control and yield optimization is an important topic in the integrated circuit fabrication process. Some effort has been put into this issue, as introduced in [3][4][5], etc. However most of the approaches are limited to the design stage of an IC fabrication process. After a well designed IC fabrication line is put into production with all the designated parameters being specified, it may turn out not to work as well as expected. The performance depreciation

of the IC fabrication line with time can't be avoided by just designing process parameters well during design stage. Equipment wears out and needs to be replaced at times. The best thing that can be done in this case is to impose strict real-time process control during the production stage. The main idea behind the approach proposed in this thesis was motivated by an attempt of implementing real-time process control of the IC fabrication line based on the on-line measurements to diagnose process failures.

From this starting point, IC process characterization has been used in conjunction with the statistical process control techniques in this thesis. The critically important parts in the approach proposed include IC process simulations, control chart techniques and backpropagation neural networks which are employed to implement the inverse mapping needed in the approach. Although the errors resulting from the statistical simulator used and the control charts could cause inaccuracy in the algorithm, the greatest difficulty lies in the implementation of the inverse mapping.

7.2 Future Research Directions

As discussed in the previous chapters, the direct mapping by a process simulator is a complicated nonlinear model with high dimensionality. This makes the inverse mapping from the process output to the process disturbances extremely difficult to implement accurately. Although we can locally approximate the nonlinear model by a linear one, it is hard to make the linear model valid over a wide range of process disturbances. We are particularly interested in large variations and major changes in the process. Coding the input and output

variables that has been used in this thesis proves to be an effective method in linearizing the nonlinear map for large-scale variations. The main drawback of the coding method lies in the errors inherent to the way in which the codes are generated. Actually what we can obtained from the output of a neural network is a set of values that may not always be close enough to the codes preset. In other words, it is difficult to select accurate thresholds in coding the outputs of a neural network. More effort needs to be done to minimize the error associated with the coding method.

Another critical problem is the training of neural networks. As discussed in Section 5.5, the most frequent reason that prevents a neural network from convergence is the existence of local minima. A number of algorithms have been proposed to deal with the local minimum problem in the literature [41][42][43]. Applying them to our approach depends on a complete understanding of the functions involved in our process model. A deep investigation into the process models will definitely benefit both the establishment of linear mappings and successful training of the neural networks. Moreover, in order to take full advantage of neural network techniques, more reasonable training pattern generation techniques have to be devised and applied. And finally if we could train the neural network while using them to monitor an IC process, that will save us a lot of time and effort in the training session, and will make this approach more practical and real-time oriented.

Bibliography

1. W. Maly, A.J. Strojwas, "Statistical simulation of the IC manufacturing process," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems,* Vol. CAD-1, No. 3, pp. 120-130, July 1982.

2. S. Nassif, A.J. Strojwas, and S.W. Director,"FABRICS II: A statistically based IC fabrication process simulator," *IEEE Trans. Computer-Aided Design,* Vol. CAD-3, No.1, pp.40-46, Jan. 1984.

3. C.J.B. Spanos, "Parameter extraction for statistical IC process characterization," *IEEE Trans. Computer-Aided Design,* Vol. CAD-5, pp.66-78, Jan. 1986.

4. P.K. Mozumder, C.R. Shyamsundar, and A.J. Strojwas, "Statistical control of VLSI fabrication processes: A framework," *IEEE Trans. Semiconductor Manufacturing,* Vol.1, No.2, pp.62-70, May 1988.

5. P.K. Mozumder, C.R. Shyamsundar, and A.J. Strojwas, "Statistical control of VLSI fabrication processes: A software system," *IEEE Trans. Semiconductor Manufacturing,* Vol.1, No.2, pp.72-82, May 1988.

6. T.K. Yu, S.M. Kang, et al, "Statistical performance modeling and parametric yield estimation of MOS VLSI," *IEEE Trans. Computer-Aided Design,* Vol. CAD-6, pp.1013-1022, Nov. 1987.

7. Herr, J.J. Barnes, "Statistical circuit simulation modeling of CMOS VLSI," *IEEE Trans. Computer-Aided Design,* Vol. CAD-5, No.1, pp.15-22, Jan. 1986.

8. L. Milor, and A. Sangiovanni-Vincentelli, "Computing parametric yield accurately and efficiently," *Proc. ICCAD,* Santa Clara, CA, pp.116-119, 1990.

9. G. Box, W. Hunter, and J. Hunter, *Statistics for Experimenters,* Wiley-Interscience, 1978.

10. M.D. Mckay, R.J. Beckman, and W.J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics,* vol.21, No.2, pp.239-245, May 1979.

11. D.C. Montgomery, *Introduction to Statistical Quality Control,* John Wiley & Sons, New York, 1991.

12. W. Maly, *Atlas of IC Technologies,* The Benjamin/Cummings Publishing Company, Inc., 1987.

13. S.M. Sze, *VLSI Technology,* McGraw-Hill Book Company, 1988.

14. C.J.B. Spanos, "Hippocrates: A methodology for IC process diagnosis," *Proc. ICCAD,* pp.513-516, Santa Clara, CA, 1986.

15. CMU Research Center for CAD, *" FABRICS II: user's manual,"* Department of Electrical Engineering, Carnegie-Mellon University, June 1987.

16. A. Papoulis, *Probability, Random Variables and Stochastic Processes,* McGraw -Hill, 1965.

17. D.A. Autoniadis, et al, "SUPREM II: A program for IC process modeling and simulation," *Tech,. Rep. 5019-2,* Stanford Electronic Lab, June 1978.

18. D.C. D'Avanzo, M. Vanzi and R.W. Dutton, "One-dimensional semiconductor device analysis (SEDAN)," *Tech. Rep. G-201-5,* Stanford Electronics Lab, Oct. 1979.

19. S. Selverherr, A. Schutz, and J.W. Potzl, "MININOS: A Two-dimensional MOS transistor analyzer," *IEEE Trans. Electron Devices,* Vol.ED-27, No.8, Aug. 1980.

20. J.M. Hammersley and D.C. Handscomb, *Monte Carlo Methods,* Fletcher and Son Ltd, Norwich, 1964.

21. C.J.B. Spanos and S.W. Director, "PROMETHEUS: A program for VLSI process parameter extraction," *Proc. ICCAD,* Santa Clara, CA, 1983.

22. M. Bolt, et al, "Realistic statistical worst-case simulations of VLSI circuits," *IEEE Trans. Semiconductor Manufacturing,* Vol.4, No.3, Aug. 1991.

23. O. Melstrand, et al, " A data base driven automated system for MOS device characterization, parameter optimization and modeling," *IEEE Trans. Computer-Aided Design,* Vol. CAD-3, No. 1, pp.47-51, Jan. 1984.

24. C.J.B. Spanos, "Statistical parameter extraction for IC process characterization," Dissertation of Ph.D, Carnegie-Mellon University, May 1985.

25. K.M. Gardiner and S.R. Jalley,"Manufacturing high-density memory chips," *Solid State Technology,* Vol.24, pp.117-122, Oct. 1981.

26. T.J. Russell, et al, "A microelectronic test pattern for measuring uniformity of an integrated circuit fabrication technology," *Solid State Technology,* Vol.22, pp.71-74, Feb. 1979.

27. D.S. Perloff, T.F. Hasan, and E.R. Blome, "Real-time monitoring of semiconductor process uniformity," *Solid State Technology,* Vol.23, pp.81-86, Feb. 1980.

28. G. Wolfe, "Scaling the mountains of data-data reduction and graphic display," *Circuits Manufacturing,* Vol.16, No.4, pp.48, 1976.

29. D.S. Perloff, F.E. Wahl, and J.D. Reimer, "Contour maps reveal nonuniformity in semiconductor processing," *Solid State Technology,* Vol.20, pp.31-36, Feb. 1977.

30. S.W. Director, W. Maly, and A.J. Strojwas, *VLSI Design for Manufacturing: Yield Enhancement,* Klumwer Academic Publishers, Boston, 1990.

31. J. Dayhoff, *Neural Network Architectures,* Van Nostrand Reinhold, New York, 1990.

32. R. Hecht-Nielsen, "Theory of the backpropagation neural network," *IJCNN,* pp.I-593-I-605, Washington, D.C., June 1989.

33. K. Hornik, "Approximation capabilities of multilayer feedforward networks," Neural Networks, Vol.4, pp.251-257., 1991.

34. S.Y. Kung and J.N. Hwang, "An algebraic projection analysis for optimal hidden units size and learning rates in backpropagation learning," *Proc. IEEE Inter. Conf. Neural Networks,* Piscataway, NJ, pp.I-363-370, 1988.

35. Y.L. Gu, "On nonlinear system invertibility and learning approaches by neural networks," *Tech. Rep.,* FP13, Oakland University, Rochester, Michigan, 1989.

36. R.L. Iman, et al, "An approach to sensitivity analysis of computer models - part I," *J. of Quality Technology,* Vol. 13, No. 3, pp.174-183, July 1981.

37. R.L. Iman, et al, "An approach to sensitivity analysis of computer models - part II," *J. of Quality Technology,* Vol. 13, No. 4, pp.232-240, Oct. 1981.

38. R.L. Iman, et al,"Latin hypercube sampling," *Tech. Rep, SAND79-1473,* Sandia Lab, Albuquerque, NM, 1980.

39. D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing,* Vols. 1 & 2, MIT Press, Cambridge, MA

40. J.L. McClelland and D.E. Rumelhart, *Explorations in Parallel Distributed Processing,* MIT Press, Cambridge, MA , 1988.

41. M.A. Styblinski and T.S. Tang, "Experiments in nonconvex optimization: stochastic approximation with function smoothing and simulated annealing," *Neural Networks*, Vol.3, pp.467-483, 1990.

42. D. Ingman and Y. Merlis, "Local minimum escape using thermodynamic properties of neural networks," *Neural Networks,* Vol.4, pp.395-404, 1991.

43. N. Baba, " A new approach for finding the global minimum of error function of neural networks," *Neural Networks,* Vol.2, pp.367-373, 1989.

44. K. Doganis and D.L. Scharfetter, "General optimization and extraction of IC device model parameters," *IEEE Trans. Electron Devices,* Vol. ED-30, pp.1219-1228, 1983.

45. P. Conway, et al, " Extraction of MOSFET parameters using the simplex direct search optimization method," *IEEE Trans. Computer-Aided Design,* Vol. CAD-4, pp.694-698, 1985.

46. A.J. Strojwas and S.W. Director, " A pattern recognition based method for IC failure analysis," *IEEE Trans. Computer-Aided Design,* Vol. CAD-4, pp.76-92, Jan. 1985