PROBABILISTIC MODELING FOR FAULT CLASSIFICATION OF PLASMA EQUIPMENT

by

Anna Maria Ison

Memorandum No. UCB/ERL M00/2

7 January 2000

COVER

PROBABILISTIC MODELING FOR FAULT CLASSIFICATION OF PLASMA EQUIPMENT

by

Anna Maria Ison

Memorandum No. UCB/ERL M00/2

7 January 2000

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Probabilistic Modeling for Fault Classification of Plasma Equipment

by

Anna Maria Ison

B.S. (Massachusetts Institute of Technology) 1991
M.S. (University of California, Berkeley) 1994

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy
in
Engineering - Electrical Engineering and Computer Sciences
in the
GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Costas J. Spanos, Chair
Professor Kameshwar Poolla
Professor David A. Hodges

Fall 1999

Abstract

Probabilistic Modeling for Fault Classification of Plasma Equipment

by

Anna Maria Ison

Doctor of Philosophy in Engineering - Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Costas J. Spanos, Chair

The continual push of performance limits for upcoming technology generations has resulted in a flurry of activity to improve manufacturing practices in the semiconductor industry. While the advent of high density, low pressure plasma etch systems has enabled chip makers to meet current performance demands without sacrificing throughput, these benefits are accompanied by increased complexity requiring good process characterization, monitoring and control.

A comprehensive model of the plasma etch equipment based on sensor data is constructed, which identifies modes of behavior corresponding to normal operation and specific failures. Plasma etching is viewed as a complex process exhibiting hybrid behavior - that is, the process contains both continuous and discrete dynamics. The continuous machine state, characterized by real-time tool signals under normal operating conditions, changes abruptly as a result of machine failures. However, the failures themselves are best classified into discrete groups corresponding to a particular type of faulty behavior. Thus, at a higher level, the state of the process can be described as nominal (i.e. no machine failures), or faulty, where the faulty state is further subdivided into categories corresponding to different causes or failure modes. At a lower level, the continuous dynamics evolve depending on the discrete state of the process. The description of the process is further complicated since, due to the nature of single wafer processing, these continuous dynamics are evolving over different time scales (a) on a second by second basis, within the processing time of a wafer (b) from wafer to wafer, and (c) from lot to lot.

Time-series and linear modeling techniques are used to characterize the continuous behavior of the machine at three time-scales. The decomposition into different time-scales also facilitates the development of a robust procedure for fault detection using statistical process control techniques. To enhance the fault detection mechanism, models are developed which capture long term trends in the signals, visible on a lot to lot basis, which are mainly caused by changing machine dynamics due to machine aging. Methods for feature selection, extraction and classification are investigated to determine the limitations of current sensor data, and whether such data can effectively be used to identify discrete failure modes. Mixture models are built which provide likelihood estimates for assignment to a fault category based on sensor variables. These are combined in a graphical model encoding the relationships among the variables of interest.

Professor C.J. Spanos

Committee Chairman

The dissertation of Anna Maria Ison is approved:

_____

Chair Date

_____

Date

_____

Date

University of California, Berkeley

Fall 1999

Probabilistic Modeling for Fault Classification of Plasma Equipment

by

Anna Maria Ison

In loving memory of my Grandparents

Table of Contents

List of Figures

List of Tables

Acknowledgments

My father once told me that truly great men stand on the shoulders of those who came before them. This sentiment captures the essence of how I feel at the conclusion of this journey. In a sense, these few paragraphs are the most difficult to write - there are so many who contribute so much to one's growth in graduate school. Yet, I must be content to merely list them all, knowing that I will likely leave someone out, and certainly cannot do justice to how each one has influenced and sustained my aspirations.

Without the solid guidance, support, and friendship of my advisor, Professor Costas Spanos, I would not be where I am today. The high degree of success achieved by all his graduates in a wide range of fields is incontrovertible evidence of his value as an advisor, mentor and friend. So, I thank him whole-heartedly for allowing me to be part of his legacy. I also want to thank U.C. Berkeley Professors Kameshwar Poolla, David A. Hodges, David Brillinger, Shankar Sastry, Elijah Polak, Andy Packard, Pravin Varaiya, Alberto Sangiovanni-Vincentelli, Alice Agogino, Amy Shuen, Kris Pister and MIT Professor Duane Boning for their support, encouragement, enthusiasm and wisdom.

My gratitude also goes to several industrial mentors, without whom, this research would not be possible: Gabe Barna, Stephanie Butler, Mike Clayton, Tim Dalton, Rick Dill, Jimmy Hosch, Ken Krieg, Jack Mott, Randy Mundt, John Plummer, Ginny Poe, M. Surendra, and Dan White.

I would like to thank the folks in my research group, spanning more than one generation, some of whom have become long-lasting friends: Junwei Bao, Eric Boskin, Runzi Chang, Roawen Chen, Mareike Claassen, Sean Cunningham, Tim Duncan, Darin Fisher,

Mason Freed, Mark Hatzilambrou, Herb Huang, Nickhil Jakatdar, Sovarong Leang, Jaewook Lee, Sherry Lee, Man Li, Jeff Lin, Greg Luurtsema, Tony Miranda, David Mudie, John Musacchio, Xinhui Niu, Sundeep Rangan, Johan Saleh, Manolis Terrovitis, Nikhil Vaidya, Crid Yu, Haolin Zhang, Dongwu Zhao.

My graduate school experience was further enriched by my friendships with past and present graduate students, Andy Abo, Kostas Adam, Ben Bonham, Bill Clark, John Davis II, Akash Deshpande, Heath Hoffmann, Thor Juneau, Jeff Kao, Mark Lemkin, Gene Marsh, Sekhar Narayanaswami, Keith Onodera, Alan Peevers, Tom Pistor, Matt Podolsky, Trey Roessig, Chris Rudell, Bob Socha, J.P. Tennant, Greg Walsh, Mark Walker, Greg Walter and Nathan Yee. Thanks also to Richard Edell, Farokh Eskafi, Aleks Gollu, John Lygeros and Karl Petty, for invaluable support and help over the years.

Thanks to the special women with whom I have had the pleasure of sharing extraordinary thoughts: Olga Blum, Lisa Buckman, Mireille Broucke, Greta Gize, Lisa Guerra, Beth Koh, Tetiana Lo, Lili Nova-Roessig, Robin Sacco, Jennifer Scalf, Polly Shrewsbury, Ekta Singh, Edita Tejnil, Dawn Tilbury, Laura Walker and Angela Wang.

A special thanks to those who added balance and vitality to my life, contributing to my physical, emotional, mental and spiritual health: to the bioengineers, who doubled as great volleyball players- Brenda Baker, Andy Liu, Mark and Sue Noworolski, Bruce Parnas, Kerstin Pfann, Wade Sunada, and Mary Wagner. Thanks also to Ben Ludwig, Micheal Shilman, Niraj Shah and the Cory Hall volleyball crowd. Thanks to my wonderful support group of women - the indoor power soccer team, and the NCWHL (women's hockey league) and especially, to "Chicks with Sticks" and coach Marc Beck. Thanks to Andre LaCroix and the Oakland "Stingers". Thanks also to the members of my softball team in the EECS softball league. Finally, my thanks go to Sensei Seijei Tanaka and fellow San Francisco Taiko Dojo members for a one of a kind experience.

To my best life-long buddies, for their constant faith, belief and support, I could not have endured without them: Stephen Nicholls, Katia Keshishian, David Oliver, Casey Santos, Yildiz and Matthew Ferri, and Richard Schenker. And thanks also to my newest buddy, Stephen Hinkson, for making the end taste both savory and sweet.

To my sister and brother, Lisa and Nory, who will always be part of me and what I accomplish - I will always be thankful for their unique brand of support.

Looking at this substantial list of people, who have made my graduate experience as full as I could imagine, I know two who will always have my appreciation, gratitude and love. For providing a solid foundation, source of trust, strength and faith; and for making me such that I could come out and make a difference through the lives I've touched - thanks Mom and Dad. You are the best of the best.

This research was funded through my fellowship sponsors, the Semiconductor Research Corporation and Motorola; and through the state of California Micro program, and participating companies (Advanced Micro Devices, Applied Materials, Atmel Corporation, Lam Research, National Semiconductor, Silicon Valley Group, Texas Instruments).

1 Introduction

1.1. Motivation

The competition to acquire and retain the forefront position in the semiconductor industry is driven by the dual goals of advancing technology while simultaneously reducing the cost per function. Achieving tighter specifications on smaller feature sizes would not be possible without developments in process technology, inevitably resulting in more complex processes and higher investment in fabrication tools. With such a heavy investment at stake, there is clearly a need to improve current manufacturing practices - monitoring and control of processes, data analysis, management and decision making to optimize overall equipment effectiveness.

1.1.1. Growth in the Semiconductor Industry

Over 75% of the world's semiconductor consumption is attributed to the production of silicon complementary metal-oxide semiconductor (CMOS) integrated circuits [1]. The wide gamut of electronic products that has resulted is due to a large extent to the staggering but steady growth of an industry, reported at an annual rate of 15% for the past 35 years [1]. Maintaining the historic productivity growth of 25-30% reduction in cost per function for IC technology characteristics despite escalating factory costs of 20% per year has become a growing concern among both manufacturers and equipment suppliers. In addition, new technical challenges posed by process complexity and the increasing number of process steps required to meet more stringent performance specifications are threatening the industry's ability to maintain the 25-30% manufacturing cost learning curve. Traditionally, the factors driving industry growth have included feature size, wafer diameter, yield, and factory productivity. With fewer gains arising from reductions in feature size, and larger wafer diameters, improvements in factory productivity, equipment and operations

are even more critical. Consequently, new emphasis has been placed on factory integration to fully exploit equipment and operational productivity.

1.1.2. Investment in Fabrication Facilities

To stay on the desired cost/performance track, the packaged unit cost per function must decrease 24% per year for microprocessors, and 29% per year for DRAMs [2]. Effectively, this means that wafer cost/cm^2 must be optimized with each generation. Factory capital costs have a major effect on the cost/cm^2, increasing at a compounded annual growth rate (CAGR) of 20%, which translates to capital cost/cm^2 increases of 15% per year [2]. Coupled with the rising cost of complex tools, growing process complexity, and wafer size increases, tool capital costs are projected to reach 90% of the total factory investment. This is no small amount considering that the cost of new semiconductor fabrication factories is viewed to approach $3 billion by the year 2000, and $10 billion by the year 2005 .

1.1.3. Process Complexity

The demands posed by IC design requirements, increasing wafer size and value, and process physics have lead to greater complexity; the number of process steps to complete an IC is projected to more than double by the year 2012 [3]. Clearly, as the amount of data collected from the process increases, the efficacy and ability of fab engineers to make decisions and extract relevant information is a key aspect to managing this complexity. To reach these goals, it is estimated that the productivity of computerized decision support tools must improve over six-fold in the next 15 years.

1.1.4. Process Control to Increase Productivity

Analyses conducted by SEMATECH on the productivity for 250nm and 180nm feature sizes and 200mm and 300mm wafer diameters reveals that reductions in feature size and increases in wafer diameter will not be sufficient to maintain the desired rate of 25-30% reduction in cost/function for IC technology characteristics [1]. To improve overall equipment effectiveness (OEE), advancements are needed in process and equipment control techniques to detect and correct faults, reduce monitor wafer usage, and optimize tool throughput.

1.2. Thesis Objective

The objective of this thesis is to address some of the challenges threatening the productivity growth rate of the industry by enhancing overall equipment effectiveness. This is accomplished through the development of better monitoring, process characterization, decision making and management of complexity for a bottleneck family of tools - plasma etch equipment. To implement these goals, various models are employed to capture machine behavior while statistical process control techniques enable effective monitoring of the evolution of the system. Using probability and statistics as a foundation provides tools for feature selection, extraction and classification of data to infer the process state. Finally, a unifying framework is presented to manage the complex behavior of the process for real-time and run to run fault tolerant supervisory control.

1.2.1. Monitoring and Process Characterization for Fault Detection

Due to the nature of single wafer processing, the machine dynamics are evolving over three different time scales-- (a) on a second by second basis, within the processing time of a wafer, (b) from wafer to wafer, and (c) from lot to lot. In this work, long term trends over several lots in marathon runs are investigated using appropriate modeling techniques and data structures to deal with the vastly different time scales. Once normal operation has been modeled and characterized, trends can be filtered out of the signals and statistical process control (SPC) techniques can be applied to the resulting residuals to detect abnormal behavior. The decomposition into different time scales also facilitates the development of a robust procedure for fault detection using SPC.

1.2.2. Data Analysis and Decision Making

The detection of an out-of-control condition merely indicates the possible presence of a fault. In order to confirm the hypothesis that a fault has occurred and to identify an assignable cause, a methodology to classify faults into discrete categories is developed. Identification and classification of normal and faulty system states utilize data from various sources. The different data types correspond to simulating small internal fluctuations around a nominal operating point through designed experiments (DOE's), and to actual failure modes or faulty states, with data collected from machines diagnosed with real man-

ufacturing problems during qualification runs. Fault classification is comprised of two distinct steps: (1) feature selection and extraction and (2) building a structure to integrate the various data types and classify them into useful categories. The final diagnostic model provides estimates of likelihood that the machine has made a transition to a faulty state. This structure is meant to function as a decision support tool to enhance the engineer's ability to make crucial decisions made possible through timely identification of the machine state.

1.2.3. A Comprehensive Model

To manage the complexity of the process, a comprehensive model is developed which characterizes the machine state, taking into account the different time scales and failure modes. This model combines and utilizes other models for both continuous and discrete behavior. Time series and linear models are used to characterize the continuous behavior of the machine. The fault detection mechanism finds abnormalities in the continuous state, and thus detects transitions from nominal to faulty states. This framework is compatible with current real-time and run-to-run control schemes, allowing for the development of fault tolerant supervisory control.

1.3. Thesis Organization

Background information and a description of the plasma process, the etch tool, and signals used for process characterization and machine monitoring are presented in Chapter 2. A description of the experiments and sensor data follows in Chapter 3. Chapter 4 discusses modeling of long term trends and the development of robust fault detection, which accounts for machine aging. Chapter 5 is devoted to a discussion of modeling techniques and the construction of a framework to integrate these various methods for the purposes of decision making. The next chapter presents the implementation and analysis results using sensor data from plasma etch equipment. Finally, Chapter 7 summarizes this work with a discussion of conclusions and future directions.

2 Etch Process and Equipment Description

2.1. Introduction

To fully appreciate the value added by better monitoring and process characterization, it is necessary here to develop some background regarding the process itself, as well as an understanding of the tool that is a critical part of this manufacturing cycle.

2.2. Overview of the Plasma Etch Process

Much of the productivity gains realized in IC fabrication can be attributed to major improvements and advances in equipment and process technologies in the etch sector. The etching process is used in the patterning of thin films to form significant features in chips. These features include gates and interconnect lines, and contact holes, later filled with metal to contact the source and drain, and to connect levels of metal to one another (vias). With the industry moving toward greater circuit integration and multilevel metallization, the successful formation of these features is even more critical, and balancing trade-offs in etch goals poses a bigger challenge. To keep pace with industry trends, designers have moved to tighter geometries, more film layers per circuit and vertical circuit structures. This in turn has resulted in a multiplicative increase in the number of film layers and etch steps per layer.

2.2.1. Goals of Dry Etching

Over the past decade, one of the most significant improvements in the etch sector has been the development of a single-wafer, dry-etch process (parallel plate reactor) replacing the wet-etch, batch processing techniques that had once dominated the industry. Through application of a voltage across two parallel plates, and with the proper mixture of gases, a

plasma of energetic electrons, photons, ions and chemically reactive species can be generated and used to etch materials. Typically, a successful etch is considered in terms of achieving several parameters. These include high uniformity, selectivity, and etch rate, while maintaining control of profile and CD, damage, sidewall passivation, residue and particles [4]. Unfortunately, there is an inherent trade-off in that each of these parameters can often only be optimized at the expense of at least one of the others.

High uniformity of the etch is desirable across the die and the wafer. This must be accomplished in the presence of both densely packed and isolated features contained in most die. Non-uniformity due to this condition is known as microloading. Critical dimension uniformity is crucial to maintain consistent performance in devices. Improvements in process monitoring and control are necessary to achieve uniformity from wafer to wafer, and from machine to machine.

Selectivity is defined as the ratio of the etch rate of one material to another, usually that of the desired etched material to a masking layer, typically photoresist, or of some underlying material. Selectivity to the masking layer is important because of its effect on CD and profile control. Furthermore, smaller feature sizes require thinner photoresist to be adequately resolved, and consequently higher selectivity is necessary for smaller geometries. Selectivity is often worse in high aspect ratio features, where etching at the bottom of a contact slows or sometimes even stops. Edges and flat areas may be given different selectivity specifications since they etch at different rates.

Throughput of the system, and hence productivity, is determined by the etch rate. However, although a high etch rate is desired, it may be accomplished at the expense of selectivity and damage control.

Profile and CD control refers to forming anisotropic profiles. Achieving close to vertical etched features (often at least 88-89 degree profiles for leading edge applications) means that the packing density on a chip can be maximized, and thus profile control is crucial to a successful etch [5].

Damage often occurs when there is non-uniformity in the plasma, inducing currents on the wafer surface that can result in electrical damage. Ion bombardment can also mechan-

ically damage a film's crystalline structure. Clearly, controlling this damage is important, especially during gate stack formation.

Sidewall passivation occurs when carbon from photoresist combines with etching gases or etch by-products to form a polymer that can coat the sidewalls and the bottoms of features. Although this residue can be useful, and in fact, is sometimes even required in order to etch anisotropic profiles, it needs to be removed after the etch. Failure to remove residue can lead to contamination and problems during resist stripping steps. This polymer film can also deposit on reactor walls, changing with pressure and time. The system may require more frequent cleaning, and more importantly, the residue can have an effect on plasma flux and hence, etch uniformity. Factors most affecting controlling residue include temperature, bottom RF power, backside cooling and process pressure.

Finally, particle control is extremely important and technology dependent. For instance, a requirement specifying fewer than 0.05 particles per square cm, with particle size determined to be smaller than 0.35 μm is not unreasonable.

2.2.2. Components of Etch

Optimizing to achieve a balance of the various parameters of dry etching - uniformity, selectivity, etch rate, profile and CD control, damage and residue control - involves an understanding of two different mechanisms that occur in the etching process.

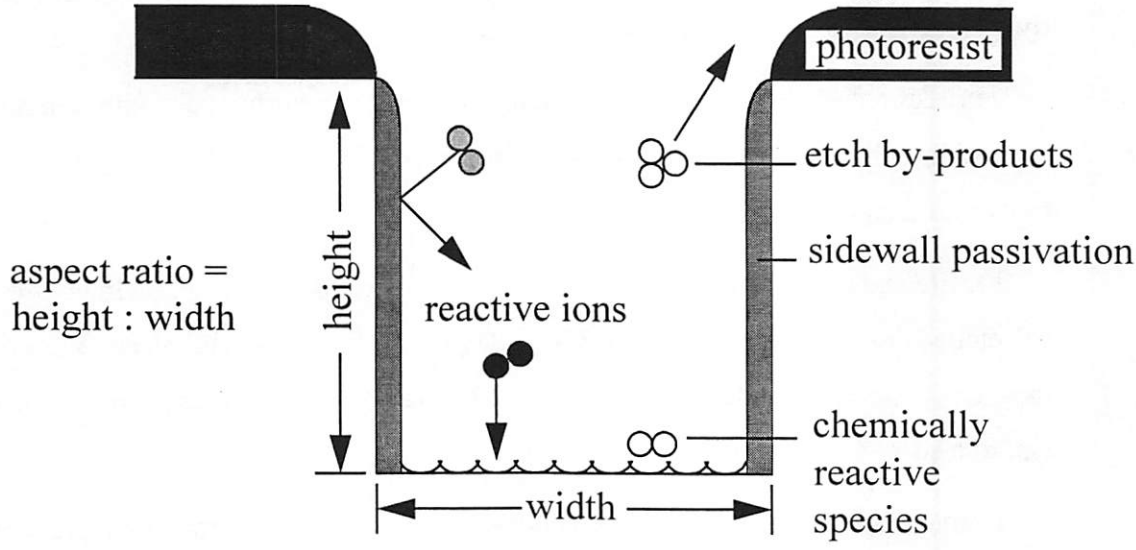


Figure 2-1. Etch mechanisms: chemical reaction and ion bombardment.

The first mechanism can be considered purely a chemical one in which a plasma generates reactive ions that react with the wafer surface and form volatile by-products. A careful selection of the gases pumped into the plasma can bolster selectivity. However, in a purely chemical etch, etching will occur equally in all directions, with no preferential direction or bias. The result can lead to a highly isotropic etch. Thus, high selectivity is achieved at the cost of profile control.

The other etching mechanism, commonly known as sputtering, is purely mechanical, a result of the bombardment of ions on the wafer surface. An electrical bias is used to propel the ions with a force strong enough to physically remove material. The main advantage of this is a highly directional etch that can be used for anisotropic profiles. However, because the mechanism is mechanical, it has poor selectivity for one material over another, and thus a different trade-off exists.

2.2.3. Challenges for Dry Etch

The real challenge lies in achieving all etch goals through a careful balance and control of the two mechanisms comprising the etch process. These parameters are even more difficult to achieve given the current industry trends - smaller dimensions with higher aspect ratios, increasing complexity in structures, multi-layer film stack etches, and new devices requiring a wider variety in types of materials being etched. Ironically, it is because of these very issues that all aspects of successful etching must be met and even surpassed.

In response to these trends, a number of innovations have been made to enable such goals to be met. One major improvement has been the development of high density plasma sources, which in turn has been driven by the desire to etch at lower pressures.

There are several key advantages to low pressure systems, including much improved control of critical dimension (CD), and minimized microloading effects (known to cause unwanted etch variations between areas of isolated versus dense features). Traditional etch systems operating under higher pressures (typically ranging from 50 to 150 mT) result in non-directional ion bombardment. In this pressure range, moving etchants in and by-products out of openings less than a quarter nanometer becomes very difficult. With higher aspect ratios, the problem is amplified, and etching tends to slow or in some cases stop at

the bottom of the feature. The result is that the feature sidewalls bow outward. In contrast, low pressure systems enable a longer mean free path of highly directional ions, enabling the etching of deep channels with submicron widths. Furthermore, scattering collisions are reduced, resulting in better profile control. Thus, etching at lower pressures is more anisotropic, allowing for the use of cleaner chemistries which consequently reduces contamination. Low pressure etch systems also result in plasma by-products that are more volatile, making their removal much easier.

Using traditional systems under lower pressures results in a drop in ion density, which lowers the etch rate and throughput of the system. However, using a high density plasma, high electron densities can be created at a lower bias, leading to decreased substrate damage and often yielding etch rates exceeding those of previous methods. The high density plasma sources are more efficient in coupling input power with the plasma, generating greater dissociation of chemical species. These sources allow manufacturers to reap the benefits of operating at lower pressures without a loss in productivity.

Another challenge in the etch market arises from the many processes that involve different chemistries, with hardware and software requirements being process specific. In general, materials to be etched in a silicon-based integrated circuit can be designated into one of three categories- polysilicon, metals, or dielectrics. It is typical to use a chlorine-based chemistry for etching polysilicon, silicides and metals, and a fluorine-based chemistry to etch oxide and nitride. Polysilicon accounts for 25 percent of the market, aluminum and aluminum alloys comprise 31 percent, while the major share accounting for 44 percent is taken by oxide (dielectric) etching.

Substantial growth is projected in all three etch market areas. The global market is expected to reach 5.3 billion U.S. dollars by the year 2000 (market figures and growth rates based on Dataquest and VLSI Research Inc.) [2].

2.3. Plasma Etch Equipment

2.3.1. High Density Plasma Sources

A variety of semiconductor fabrication processes utilize plasma generation. These include etching, resist stripping, passivation and deposition. Plasma generation involves inducing electron flow to ionize process gas molecules. Kinetic energy is transferred through individual electron-gas molecule collisions. Typically, electrons are accelerated in an electric field. In a conventional parallel plate plasma etcher, a semiconductor wafer is placed on a lower electrode and a plasma is generated by applying radio frequency (RF) energy between the lower and a parallel upper electrode. However, one drawback to using an electric field normal to the wafer is that the conversion of kinetic energy to ions is inefficient, especially at low frequencies and pressures under 100 mT. Most of the electron energy is lost through electron collisions with chamber walls, or with the wafer itself, which is not only wasteful, but can also cause wafer heating.

Various methods have been developed to make energy conversion for generating plasmas in semiconductor applications more efficient. One method involves microwave resonance chambers that use ultra high frequencies to shorten the electron oscillation path, thus making electron energy more likely to be transferred to process gas molecules instead of a chamber wall or a wafer. Electron cyclotron resonance (ECR) and helicon resonance utilize a controlled magnetic field external to the chamber to induce a circular electron flow within the process gas. Although both methods are efficient in terms of energy conversion, both also suffer the disadvantage of producing a highly non-uniform plasma. This difficulty can be overcome by flowing the plasma some distance before exposure to the wafer. However, although the additional flow path can make the plasma more uniform, it also results in some ion recombination which reduces the effectiveness of the plasma. Furthermore, these methods have a limited pressure operating range. Microwave resonance chambers operate between 1-760 Torr, while ECR chambers have a 0.0001 to 0.1 Torr range. Additional disadvantages include the increased cost and design complexity incurred by the need for extra flow distance, and the problem of controlling the magnetic field in ECR systems. There are other approaches to increase energy conversion efficiency - magnetically -enhanced plasma systems and inductively-coupled electron acceleration are among these.

Magnetically-enhanced plasma systems utilize the combined forces of a constant magnetic field parallel to the wafer surface and a high frequency electrical field perpendicular to the wafer surface. Electrons flow in a cycloidal corkscrew path thus increasing the distance travelled as compared with straight path which would be due to electric field alone. Again, although ion generation is relatively efficient, there are difficulties posed by maintaining a large uniform magnetic field, and the system is generally limited to an operation range of 0.01 to 0.1 Torr. Inductively-coupled plasma systems also cause electrons to flow an extended path. Two techniques fall in this category, both use alternating current to transformer couple energy to a gas. The first uses a ferrite magnetic core to enhance transformer coupling between a primary winding and a secondary one which consists of a closed path through the gas. This technique uses low frequencies - below 550 KHz. The second employs a solenoid coil surrounding the gas to be ionized. This technique can either use low frequencies, or frequencies in the range of 13.56 MHz. Unfortunately, neither technique provides a uniform plasma adjacent and parallel to wafer surface. Gas ionization is non-uniform, and exposure to the wafer occurs downstream.

Inductively coupled plasma (ICP) and transformer coupled plasma (TCP) source technologies claim the ability to independently control plasma density and ion energy. To accomplish this, the TCP technology allows separate control of plasma density generated by the main source, while a radio frequency (RF) source below the wafer is used to control the energy propelling the ions toward the wafer surface.

Various issues are addressed by the different source technologies, including plasma uniformity, the width of the process window, and overall cost and complexity of the system. Regardless of the source technology, the main benefit remains - the ability to operate under low pressures, while maintaining a plasma sufficiently dense to sustain the high etch rates needed to boost productivity.

2.3.2. TCP Etchers

Although data analyzed in this work were obtained from various different sources and different machines, a significant portion of the experimental work and data collection was conducted on plasma etchers utilizing the transformer coupled plasma (TCP) technology.

This section is intended to give a brief description of the equipment and its operation to provide the reader with an understanding of the experiments that follow, as well as a feeling for what the sensor signals mean.

2.3.2.1. Equipment Description

Much of the data collected comes from Lam's TCP product line of high density, low pressure etch systems. The machines are fully automated, single-wafer plasma etching systems using a transformer coupled plasma source. This technology is capable of generating high flux uniform, planar plasma over a broad pressure range with little or no directed ion energy. The system includes a reaction chamber bounded by a dielectric shield or window with a TCP coil and an RF source coupled to the coil. The etching process occurs as wafers are exposed to the plasma generated in the reaction chamber under vacuum conditions. As the etching begins, gases are mixed in an orbital gas panel and pumped into the chamber through a ring of gas outlets (gas ring) around a lower electrode. RF power is delivered by the TCP coil and tuned by the upper RF match assembly, ionizing the gases. Similarly, RF power is delivered by the lower electrode and tuned by the lower RF match assembly. A DC bias is induced on the wafer to control the ion direction and energy.

The combination of chemical reactions and ion bombardment on the wafer surface causes removal of material not protected by a photoresistive mask. During this etching process, the plasma and RF electrical field are completely contained in the reaction chamber. A turbo pump is used to remove waste material from the chamber and to pump unreacted gases out of the chamber after the process is completed.

To provide a consistent etch environment, the reaction chamber is kept under vacuum at all times (except during maintenance) between two load locks. The entrance and exit loadlocks act as buffers between the cleanroom environment and the chamber, thus enabling the chamber to remain at vacuum for better etch repeatability. Vacuum pressure is controlled through a chamber plenum connected to the back of the chamber. The backing pump supplies vacuum to the turbo pump through a turbo isolation valve. A control gate valve controls vacuum supplied by the turbo pump to the chamber plenum. There is also a bypass isolation valve which can be used to supply vacuum directly from the backing pump

to the chamber; however this is kept closed during normal operation. First, the pressure is brought down to 3 Torr through the coordinated action of both isolation valves. Then the control gate valve is opened to supply vacuum from the turbo pump. Temperature is regulated with cartridge heaters in the upper chamber assembly.

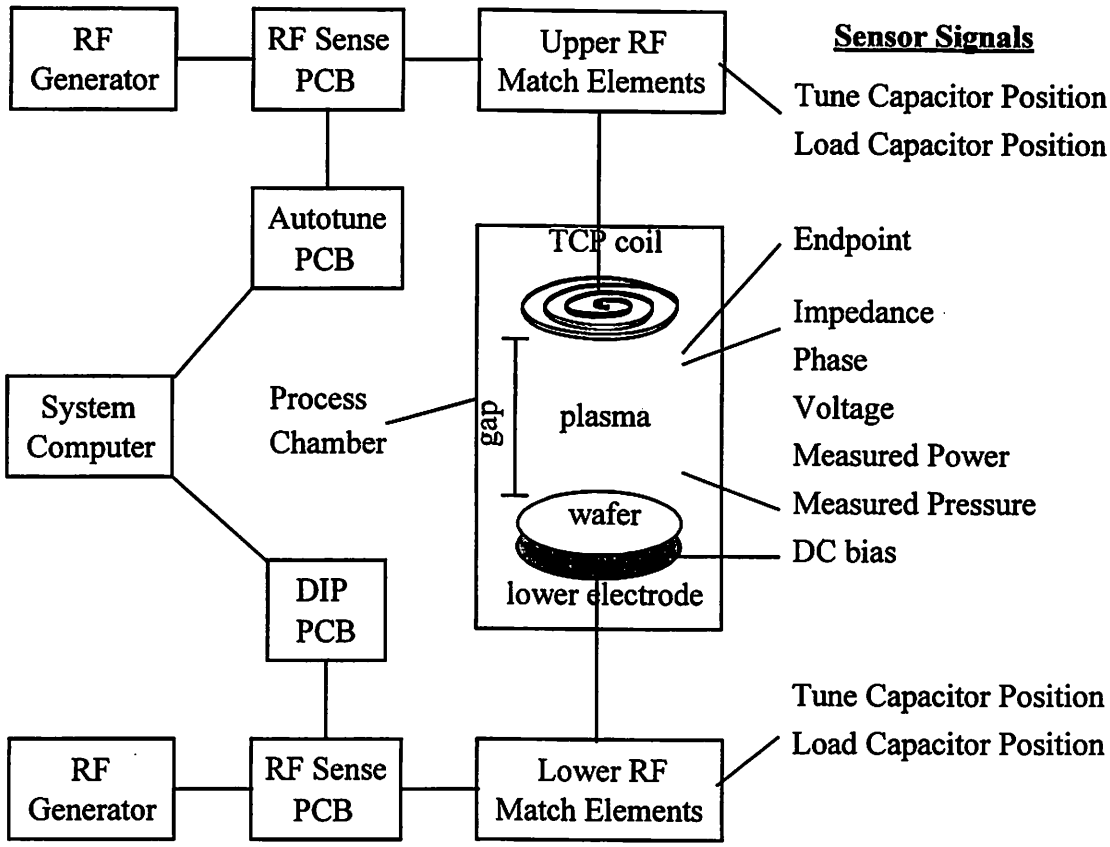


Figure 2-2. Components of a TCP plasma etcher and associated sensor signals

The RF system is comprised of two RF match assemblies and two RF generators, for the TCP coil and the lower electrode respectively. The RF generators are capable of supplying as much as 1250 watts at 13.56 MHz power.

The RF generators use an impedance matching circuit (RF match assembly) to maximize power transfer and a tuning circuit to provide for resonance at the operating frequency of 13.56 MHz.

2.3.2.2. System Operation

The system controls the etch process through pre-programmed recipes determined by user-defined, software controlled settings. Etching recipes are comprised of a series of steps to determine gas flow rates, chamber pressure, RF power, gap spacing, chamber temperature and helium backside cooling pressure. The values can be programmed as a recipe so that operators may select a desired recipe to start a specific etching process.

There are three basic process steps in an etching recipe. The first is a stability step used to stabilize the environment- the chamber pressure, position of the gap, and process gas flows must all be stable before the RF power is turned on. This step may also be utilized between recipe steps, where process chemistries or pressure may be changing. Stabilization is followed by the etch step, where chemical reactions and ion bombardment generated by the plasma and ignited by the power cause the wafer to be etched. An overetch step may also be employed to continue etching after the endpoint. In this case, a stability step may be inserted between the etch and overetch steps. Finally, the pump and purge step rids the chamber of gases and by-products generated during the etch step.

The chemistry and dynamics of the etch step are determined by six input parameters specified by the user. These basic settings are listed in Table 2-1. In addition, users can program RF tuning parameters to determine how impedance match conditions (between the chamber and the generator) will be met.

Parameter name	Units	Parameter controls...
Pressure	mt (millitorr)	chamber pressure
RF Top power	W (watts)	RF power to TCP coil
RF Bottom power	W (watts)	RF power to lower electrode
Gap	cm (centimeters)	gap spacing
Gas 1-8	ccm (cubic cm/min.)	process gas #1-8 flow rate
He clamp	t (torr)	helium cooling pressure

Table 2-1. Recipe Parameters

Because the chemistry and dynamics of the etch step are determined by the various equipment settings, the adjustment of these settings can be the key to achieving desired

goals. In particular, the width of the process window determines how far one can push various equipment settings- the source and bias power, pressure, flow rate, chemistry, ion energy and density, wafer temperature- to meet desired process goals of high etch rate, selectivity, anisotropy, minimizing residue and damage. Of course, specific process goals depend on the application - the material being etched, the material where the etch stops, over-etch requirements, concerns about damage. Process optimization through changing various equipment settings involves complex trade-offs. For instance, changing the pressure or power applied to the plasma can result in plasma density changes. Application of high power at low pressure can provide higher ion density and result in better residue control. In general, residue formation can be affected by varying pressure, gas flows, gas ratios, power, temperature, overetch time and backside cooling. In contrast, higher pressure with low plasma density results in better etch rate microloading control, higher resist selectivity, and a higher etch rate.

2.3.2.3. Sensor Signals

Table 2-2 lists the signals collected by various sensors located on the equipment.

The signals can be divided into groups according to origin and function. Optical emission sensors monitor plasma intensity and endpoint, chamber environment sensors report pressure, gas flows, temperature and backside cooling, and RF sensors associated with the TCP top match and bottom match assemblies provide information about the continually changing state of the plasma and the machine's condition.

The chemical species reacting in the plasma during the etching process produce optical emissions that provide useful information. In particular, a typical endpoint detector monitors the intensity of the plasma at a specific wavelength in order to determine when the etch has reached completion as indicated by a drop in the intensity profile. Because this intensity reading is sensitive to residue accumulating on the chamber window, this sensor signal is particularly vulnerable to effects of machine aging and long term chamber conditioning.

Origin/Function	Sensor	Description
Optical Emission	Endpoint	Plasma intensity at a particular wavelength
Chamber Environment	Pressure	Measured chamber pressure by a manometer
	Power	Applied RF power
Top TCP Match	TCP Tune	Tune capacitor position in upper match network
	TCP Load	Load capacitor position in upper match network
	Impedance	Impedance seen by the upper match network
	Phase	Phase error between current and voltage
Bottom RF Match	RF Tune	Tune capacitor position in lower match network
	RF Load	Load capacitor position in lower match network
	Impedance	Impedance seen by the lower match network
	Phase	Phase error between current and voltage
Other	Voltage	Voltage on the RF coaxial cables connecting the match modules to the generators
	DC Bias	Applied to lower electrode to direct ion energy

Table 2-2. Sensor signals collected for the Lam TCP 9600 plasma etcher

The chamber pressure is measured by a capacitance manometer. A valve controller compares this measurement with the setpoint value for pressure specified on the recipe and adjusts the control gate valve opening to maintain chamber pressure at that setpoint. In addition, process gas flows are individually controlled and monitored. Furthermore, during the process, helium is pumped through the lifter pin holes in the bottom electrode, to the backside of the wafer in order to conduct heat from the wafer to the cooler electrode. The wafer is secured to the bottom electrode by a clamp or by an electrostatic chuck. Helium flow greater than a certain maximum threshold can indicate a broken or misplaced wafer.

During etching, as chemical compositions change and by-products are generated, dynamic changes in the load impedance of the plasma result in reflected RF power to the generators. The top and bottom match assemblies function to independently optimize the load as seen by their corresponding RF generator in order to ensure efficient transfer of power to the plasma. As soon as the RF power is on, the match assemblies monitor the voltage and current of the applied RF power. As the plasma impedance changes during etching,

the match assemblies adjust the phase and magnitude of the forward RF power to optimize the load. For the RF generators, a 50 ohm load is ideal- this minimizes reflected power.

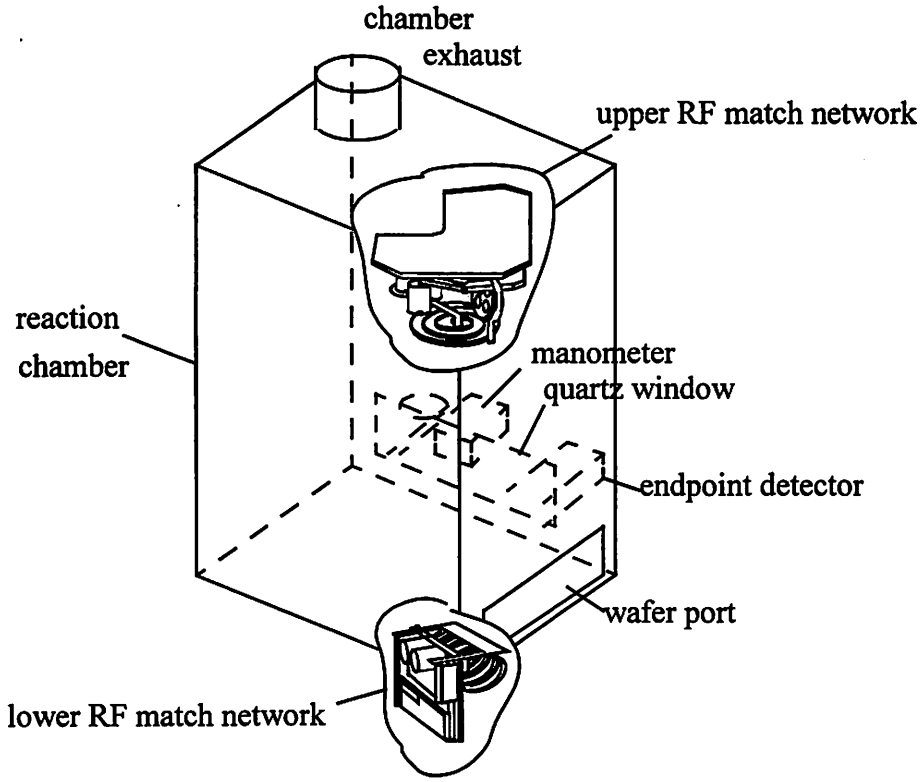


Figure 2-3. Reaction chamber - Lam TCP 9600 etcher

Each match assembly is comprised of a match module, a sense box and an autotune PC board (PCB).

The upper match, or TCP coil match module, has two servo-driven vacuum-sealed variable capacitors, a load capacitor and a tune capacitor, as well as load coil. These elements form a variable coupling transformer - the load coil is clamped once the power signal coupling to the secondary side has been adjusted. The load capacitor is used to transform the real part of the reflected plasma load impedance to 50 ohms, while the tune capacitor is used to cancel the reactive part. Positions of the variable capacitors are monitored and controlled by the match assembly.

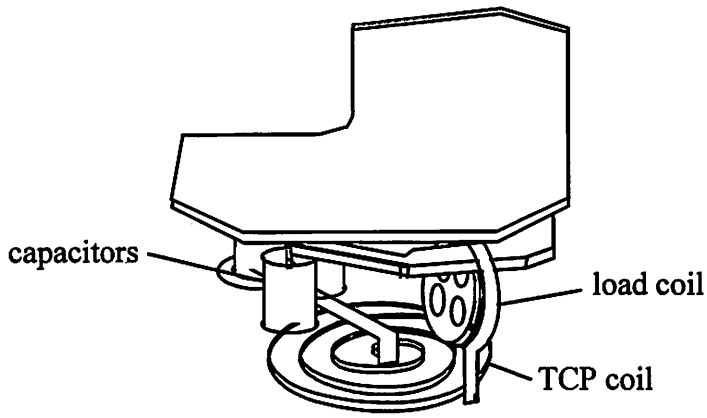


Figure 2-4. Upper match network

Similarly, the lower match, or lower electrode match module, is comprised by a high-current series-resonant circuit with a tuning vane and a load coil. Both are adjustable and have position feedback potentiometers. These positions are also monitored and controlled by the match assembly. In addition, there is a small DC bias PCB in the lower match, which monitors DC bias.

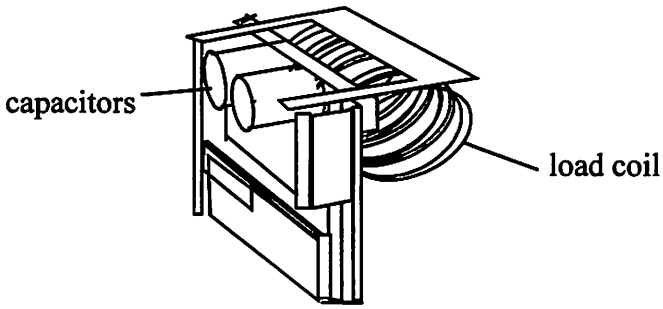


Figure 2-5. Lower RF match network

Each match module is connected to an RF sense box, which in turn is connected to the respective RF generator. These sense boxes contain a capacitive voltage divider and toroidal current sensor to detect the voltage and current on the RF coaxial cables connecting the modules to the generators. These signals are used by the match assemblies to autotune in

order to make the load the desired 50 ohms. This minimizes reflected power to the generators.

The DIP PCB (Drive-Impedance-Phase board) for each match module is used to control the phase and impedance of RF power to minimize reflected power. This is accomplished by adjusting the positions of the variable capacitors in the match networks. Circuits in the PCBs can determine phase and impedance errors from the voltage and current signals sensed in the RF sense boxes. The PCBs' drive motors on both the upper and lower match to adjust the tuning elements and correct these errors.

The user can program the RF tuning parameters and enter soft and hard tolerances to control the range of conditions for matching. Six parameters control the RF match for the TCP coil, namely, TCP RF mode, delay, tune saved, load saved, tune learn and load learn. The system computer sends the values in the tune saved and load saved parameters (in the recipe) to the autotune and DIP PCBs. These parameters represent the positions of the variable capacitors (tune and load) in the upper match assembly, and can be changed by the operator. The default value is 16383, a setting which represents a value in the midpoint of a range that spans from 0 to 32000 counts. This midpoint position is assumed to reduce adjustments required once autotuning commences. The tune learn and load learn parameters store the final tuned positions of the upper match module variable capacitors at the completion of the process. These values are often used in place of the default in order to reduce adjustments for autotuning in the subsequent wafer. They also allow process engineers to determine the final tuning positions for an automatically tuned step. These six parameters help to optimize RF match tuning. The user can switch to Manual mode, using the pre-position settings determined by the tune saved and load saved parameters. Users can also choose to delay the point of auto-tuning in order to save unnecessary "hunting" at the beginning of a step. The delay parameter allows time for the plasma to stabilize, preventing the RF system from attempting to tune to changing conditions in the reaction chamber. Thus, it basically controls how long the system delays before giving control to the auto-tune. During the delay, the RF match holds the positions specified in the tune saved and load saved parameters. Since these capacitor positions are the starting point for

the auto-tune step, tuning times may be optimized by pre-positioning the capacitors close to the expected tuning position.

Plasma etching is a complex process involving many variables and changing parameters. Although there have been attempts to predict behavior through physical models, these fall short of capturing or accounting for variability in the system. For this, there is some added value in using an empirical approach, the application of statistical techniques and tools to capture variability and relationships among available sensor signals. Furthermore, with this approach one can construct a decision making tool that can be automated using signals that are collected as part of normal monitoring. Credibility is enhanced by using familiar signals that have physical meaning to the operators.

Throughout this study, one issue continually of concern is that of repeatability. This issue arises in the reliable fingerprinting of a particular kind of machine problem or fault, and is especially aggravated by natural machine to machine variability. Extracting relevant information can be problematic, given the nature of the data and variability encountered. Although tackling this problem can be made easier with larger databases and thorough documentation of machine events (including regular maintenance, adjustments, and failures), the issue of repeatability over different machines may never be completely resolved. In this case, some model "training" will be needed to capture the individual character of the tool.

The general issue of modeling long term tool behavior is discussed in the fourth chapter. The next chapter is devoted to describing the data and the experimental design.

3 Data Description and Experimental Design

3.1. Introduction

The investigation to characterize, monitor and diagnose states for plasma etch equipment revolves around different types of datasets; the choice of the kind of data to be used is determined by the objective of the model. For instance, in order to capture normal operation, and in particular, to model machine aging and long term drift, we use data collected from marathon runs in a production environment over a time period extending over several cleaning and preventative maintenance events. Because this data captures machine drift, visible only over these extended periods of time, long term behavior can be observed, and models for filtering variability at this time scale can be constructed and validated. The resulting residuals are then analyzed using multivariate statistical process control techniques for fault detection on a lot-to-lot basis. Unfortunately, in this experiment, due to insufficient documentation relating to the processed lots, information is not available to assign causes to lots determined to be statistically out of control.

Finding assignable causes for detected faults requires datasets that are well-documented. The objective of the investigation is two-fold: (1) to assess the utility of sensor data in identifying and classifying machine problems reliably and (2) to extract the relevant information from the data, so that prompt action may be taken. The system is trained to find features which fingerprint the machine state, and to classify the data to a defined state. To construct such a system, it is clear that the state of the machine during the data acquisition process must be known, and hence the need for good documentation.

We consider two types of failure data arising from different sources. The first is caused by equipment miscalibrations, which result in small internal fluctuations in the plasma. We simulate the occurrence of this type of problem through DOEs (experiments designed to span the process input space) exploring a range of conditions around a nominal operating point. Classification techniques are then applied to predict the various operating conditions. The second type of failure arises from actual machine problems. Here we employ manufacturing data collected for machine qualification, where the faults, diagnosis, and action taken are all documented. We have taken this problem one step further by designing a failure mode experiment, which simulates the problem by physically changing the tool's condition to mimic a breakdown event.

3.2. Fault Detection and Classification (FDC)

The primary goal of fault detection is to identify when the process is no longer within operational bounds, indicating that action must be taken to correct the problem. In order to accomplish this goal, we need a model of the process that captures normal machine behavior under acceptable operating conditions. When measured values from sensor readings deviate significantly from our model predictions, we signal the detection of a fault. The identification of an anomaly is based on establishing control limits that act as bounds for acceptable deviation due to natural variation in the process. Fault classification schemes are then employed to identify the source of detected faults.

Our participation in the fault detection and classification (FDC) SEMATECH J-88-E Program at Texas Instruments conducted from 1995 through 1996 allowed us access to a source of valuable data and information for model construction and analysis. The goal of this project was to develop and evaluate techniques for FDC on a commercial semiconductor manufacturing tool using non invasive sensors and commercially available software. Automated data acquisition involving two RF sensors and machine state sensors embedded in the tool was implemented for a Lam 9600 TCP plasma etcher. Although wafer state sensors provide the most direct and easiest access to useful information, they are often not available on original equipment manufacturer (OEM) processing tools. Thus, this project concentrated on machine and process state sensors, which, by their nature, are both non-

intrusive and readily available. Due to the large volume of sensor data collected, one major difficulty lay in obtaining useful information from redundant correlated measures. There were also practical considerations regarding the choice of sensors. In general, it is desirable to stick to sensors that are low in cost, but highly reliable, especially for collecting data over extended time periods.

A key requirement in the development of models for fault detection and classification is that they be robust over time. One goal for the project was to account for the presence of long term trends. Over longer periods of time, a significant amount of normal variation in sensor readings can be expected, which is not related to either a process or a wafer state fault. Not accounting for this variation in the models can result in increased false alarms and decreased sensitivity to the real faults we wish to detect.

To properly address this issue, we note that the machine behavior can be described as evolving over different time scales, each with its own sources of process variation:

(1) maintenance-cycle-to-maintenance cycle

The highest time scale (encompassing the longest periods of time) can be considered as the change in machine behavior from one maintenance cycle to another. In particular, any given processing tool is subject to periodic cleaning, maintenance and repair over the course of its lifetime. The cycle time varies by tool, but is typically on the order of one to two months between major maintenance "events". In the interim, thousands of wafers may be processed through a tool. The maintenance event performed at the end of a cycle attempts to restore the machine to its original state. Practically speaking, this is rarely achieved, and hence there is some discontinuity between cycles as well as process variation from one maintenance cycle to another within a tool.

(2) within a maintenance cycle

We can consider the next time scale to be contained within one maintenance cycle. Within this time frame, gradual accumulation of residue in the chamber and normal wear and tear of machine parts characterized as consumable or replaceable constitute "machine aging". This behavior is clearly observed as a continuous slow drift in the process variables

as indicated by sensor measurements. However, it is important here to distinguish between process degradation (as reflected in the process variables), and sensor degradation. That is, there are natural limitations imposed by the use of sensors as imperfect measuring instruments of process variables. One clear example of this is in the continuous decay of the endpoint signal observed over the course of a maintenance cycle. This turns out to be an indication of window transmittance degradation due to residue buildup on the chamber window. Thus, the actual state, in this case, the intensity of plasma, is more stable than what is indicated by the sensor. It is the sensor measurement that is grossly affected by machine aging. Accounting for this in our models allows us to make adjustments to the fault detection mechanism so that it is robust over time.

(3) lot-to-lot

A typical lot consists of twenty four wafers and consumes between one to two hours of processing time. It is not uncommon to observe abrupt shifts from lot to lot. These shifts can be attributed to two sources. The first involves changes that happen in "upstream" processes resulting in different wafer states. In other words, the incoming material, the wafers themselves, are not identical from lot to lot. The second source is a change in the chamber state - this would be our primary concern. Lot to lot variability is further complicated by the fact that the data comes from a development lab where the devices are experimental and thus different from each other. This results in a greater likelihood that the incoming material will vary as compared with a typical production environment, in which only a few devices are processed in a highly repeatable fashion.

(4) within a lot

Within the course of processing a batch of twenty four wafers, there is typically minimal process variation. However, two phenomena are noteworthy. The first results from the gradual warm up or degassing from chamber walls, known as the "first wafer effect". In the case of the Lam 9600, this phenomenon appeared more as a first-eight-wafer effect, where the first eight wafers displayed the behavior normally confined to the first wafer. The second observation within a lot is the presence of a slow drift due to trends in upstream

conditions, resulting in a gradual change in the incoming materials (the wafers themselves).

(5) wafer-to-wafer

From wafer to wafer, variation occurs on the order of minutes. This variation is minor in general, with discontinuities that can be attributed to variations in the incoming materials. In this particular project, variation between odd and even wafers was observed. The cause of this was determined to be the use of an alternating track during an upstream lithography step. However, because DOE settings were changed from wafer to wafer, fault classification is generally concentrated on this time scale, with "true" faults manifesting themselves as deviations detected on a wafer-to-wafer basis.

(6) within the time scale of one wafer being processed

This is considered the "real" time scale, where process variable trajectories are monitored at 1 second intervals for a total duration of between 10-100 seconds for one wafer. Although it is important to implement real-time fault detection within the time scale of processing one wafer, the emphasis of the experiments is geared to the wafer-to-wafer detection, with faults injected at this level.

(7) within a processing step

Because there are different materials comprising the stack, there are often different "regions" in the etch step within the etching of a single wafer. These regions may or may not correspond to distinct "steps" in a recipe, but often reflect the different components of the thin film stack that is being etched.

(8) within a processing sub-step (region)

Sensor signals are typically unchanging within a region, but can have non linear features. Also, we have observed transient behavior during the plasma ignition step which can indicate a problem with the match network. However, data taken at the 1 Hz sampling rate is too slow for transient analysis.

Constantly changing conditions on these various time scales further complicate any automated procedure or method to analyze the data. Thus, it is crucial to distinguish between true faults and significant variation in the process unrelated to any fault event.

3.2.1. Data from Marathon Runs

Modeling machine behavior over the long run necessitates sensor data acquisition, as deterministically as possible, at regular intervals, over long periods of time. Unfortunately, the "routine" process generating the data for the J-88-E project was really anything but routine, as compared with any production environment. With device and process development as daily tasks in a development laboratory, the data naturally encompasses a large variety of wafers and devices processed through the chosen tool. Thus, "routine" was necessarily conducted and defined on a variety of different structures and devices. In contrast, in a true production environment, the tool would be more stable, focused on producing a limited set of devices in a repeatable fashion.

A primary etch tool has a typical loading of five to ten lots per day. Furthermore, the tool must be opened periodically for cleaning and maintenance. In addition, unanticipated problems require equipment hardware changes. The cleaning, maintenance and repair events include the preventative maintenance (PM), which involves opening the etch chamber, performing cleaning and maintenance, and resetting the wafer count to zero. In contrast, in the mini-clean (MC), the chamber is opened, some maintenance and cleaning are conducted, but the wafer counter is not reset.

The goal of long term modeling is to make the fault detection mechanism robust to system changes resulting from regular periodic activities, allowing identification of deviations in normal process conditions resulting from altered setpoints or injected faults. The data are collected from a Lam 9600 TCP plasma etcher running an aluminum stack etch process. In the main chamber a TiN/Al - 0.5% Cu/TiN/oxide stack is etched with a BCl_3/ Cl_2 process. From the process point of view, the key parameters are the line width of the etched Al line, (more specifically, the line width reduction compared to incoming resist line width), uniformity across the wafer, and oxide loss. Table 3-1 shows the standard recipe (Recipe 44) used for this process. In particular, note that there is a series of six menu

steps- the first two are used for gas flow and pressure stabilization, step 3 is a brief plasma ignition step, step 4 is the main etch of Al terminating at the Al endpoint, and step 5 is the over-etch for the underlying TiN and oxide layers. This is a single chemistry etch process, that is, the chemistry is identical for the main etch and over-etch steps. Step 6 vents the chamber. Figure 3-1 contains a typical process profile of the endpoint signal from the Lam-Station data set. This clearly shows the stabilization step followed by the three regions of the etch: Al, TiN and oxide etch steps.

Parameter	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6
Pressure (mT)	90	10	10	10	10	90
TCP (W)	0	0	1	350	350	0
RF (W)	0	0	100	132	132	0
BCl_3 (sccm)	0	75	75	75	75	0
Cl2 (sccm)	0	75	75	75	75	0
He Clamp (T)	0	9	9	9	9	0
Time (seconds)	15	30	3	Endpt	50	15

Table 3-1. Recipe for standard Al-stack etch in the Lam 9600

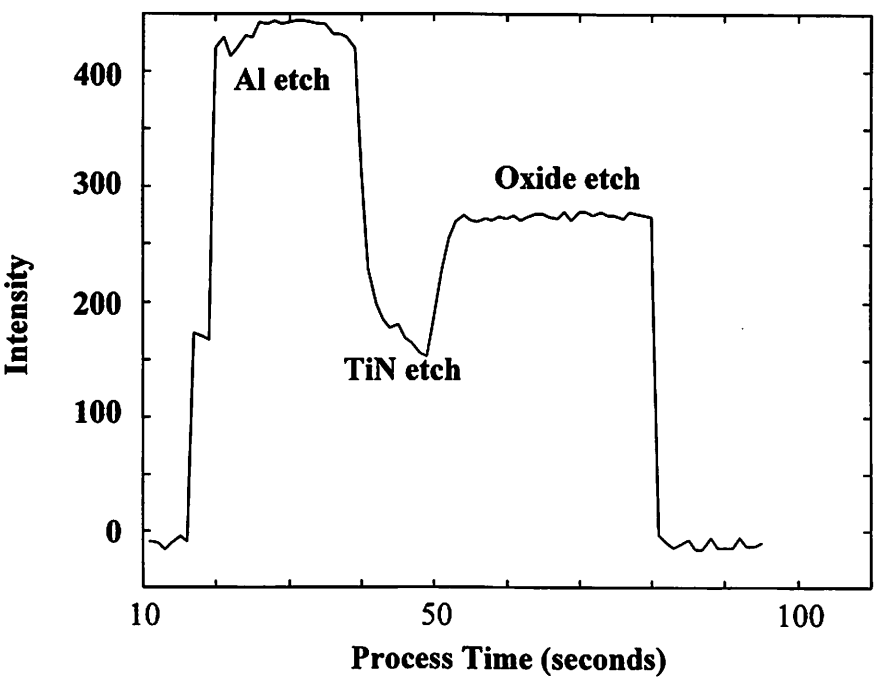


Figure 3-1. Endpoint trace of standard Al-stack etch process

The data from "routinely" processed wafers on a Lam 9600 etcher in a development lab were collected from a variety of devices. Thus, although the composition of the etched layer is similar, the nature and thickness of films can vary from lot to lot. Two routine processes (here used Recipes 44 and 45), differed only in the pressure setting, with the standard, Recipe 44, set at 10 mT, and the alternate, Recipe 45, set at 20 mT. Lots were run intermittently around the clock, with machine state signals available for analysis, but no wafer state data. During the period of monitoring this tool, between 1995 and 1996, 187 lots of data were collected, encompassing 8 maintenance cycles in total.

3.2.2. Designed Experiments

The long term characterization of the process enables fault detection that is robust over time. However, once an anomaly in processing conditions has been detected, the goal becomes that of identifying the cause of the problem. One source for training a system to detect such anomalies is obtained using injected faults in processing conditions, accomplished by changing the target machine settings. These faults are simulated using a DOE, or design of experiments, varying the settings over the operating domain of the process.

The DOEs conducted in this project are a five-level central composite design, blocked with resolution 2(5-1) + star points generated for five input variables, (TCP power, RF power, pressure, total gas flow and gas ratio - Cl2/BCl3). Appendix B contains the details of the DOE design. Operating ranges were chosen with the help of processing engineers, with pressure ranging from 7 to 20 mT to include the two routine processes (Recipes A and B).

In the first set of DOEs, experiment 30, the biggest problem resulted from the corner-points for cross-validation experiments, where pump and flow constraints prevented using low flow with high pressure, and high flow with low pressure conditions. These runs resulted in the process halting at ten seconds into the main etch due to pressure errors. However, the process could generally be restarted and continued after the error occurrence.

3.3. Manufacturing Fault Data

Obtaining actual production data with properly identified machine faults and diagnosed causes is extremely difficult. The laboratory often collects sensor data during processing; however, tracking of repair and maintenance is scant and unreliable. To partially circumvent this problem, we turned to an equipment supplier. Lam Research provided us with manufacturing data from TCP etchers obtained during qualification runs. These runs are conducted to ensure proper machine operation before shipment to the customer, and thus, all machine faults and actions taken as a result are recorded.

Table 3-2 summarizes the machine failures logged in during the qualification runs. The table lists causes of machine failures, as diagnosed by process engineers, observable symptoms of each type of failure, and action taken to deal with each problem. These data comprise an evidence library, depicted in Figure 3-2, where we have categorized and divided the data according to the corresponding type of machine failure.

Cause	Symptom	Result
Gas line bracket grounding	High/low clamp flow DC bias, RF load signals	Replaced outer screws on main chamber
Water on wafers	TCP tune signal	Adjust dry time
Frequency shift module (FSM)	Phase shift, bottom (RF) line impedance	?
?	Line impedance, clamp flow	Replaced DIP, TCP match, orifice for He
Gas O-ring	?	Low etch rate
?	RF load and RF tune signals	Put on lower match cover
Manometer	Chamber pressure	Readjusted

Table 3-2. Machine failures - causes, symptoms and results

Note that there are missing entries in Table 3-2. This is fairly typical in manufacturing and production, where the cause of a problem is unknown or never diagnosed, there are no observable symptoms in terms of monitored signals, or no action is taken. Incomplete data sets and poor records complicate the task of building a diagnostic system. Fortunately, in this set of data, we find that most of the problems fall into three basic categories - the base-

line, which represents normal operation and is used as a reference point, problems related to gas line grounding issues, and problems resulting from the match networks, both top (TCP) and bottom (RF) match modules.

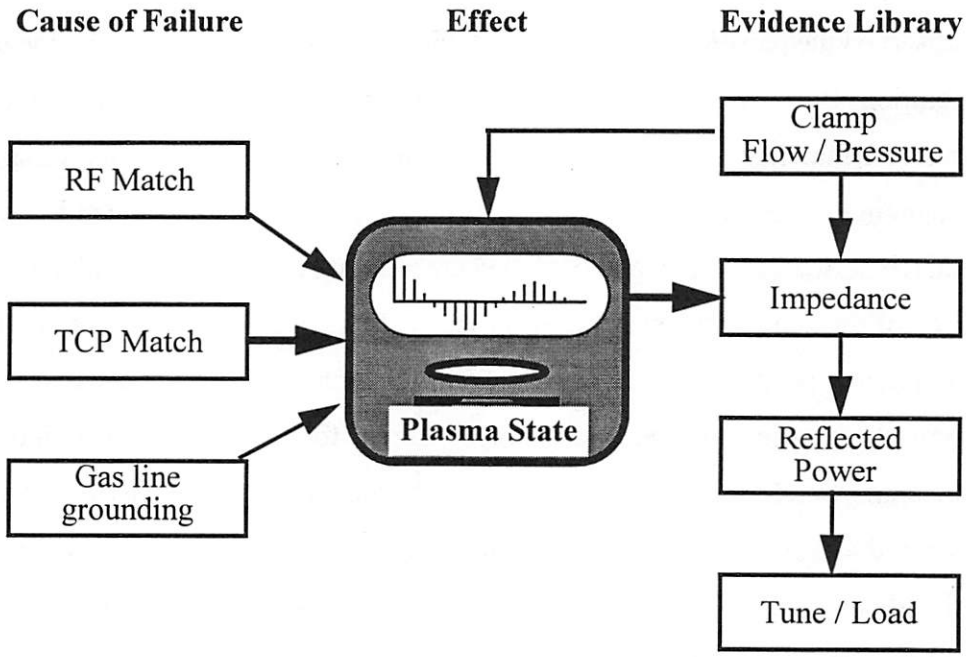


Figure 3-2. Identification of Failure Modes from Qualification Data

A further complication arises due to the different machine types. Unlike the experiments conducted in the J-88-E project, this qualification dataset is collected from different tools, which differ in both hardware and software aspects. Thus, variability occurs not only as a result of normal machine-to-machine differences, but also as a result of the change in hardware and software implemented for the machines. For our evidence library, we identified four different types of machines. Hardware differences resulted from the use of a clamp versus the electrostatic chuck to hold the wafer in place. Software differences affected values of sensor signals, such that some sensors could not be compared across machines utilizing different software.

3.4. High Speed Data

The objective in collecting this dataset is to identify cues relating to predictions of RF match problems, and conditions where the plasma will not ignite. To find such cues, we observe sensor responses to high and low preset values for the positions of the load and

tune capacitors in the RF match network, and simulate a failure mode where the capacitors fail to respond to command signals.

Upon examining manufacturing data from qualification and marathon runs, we have observed the presence of transient behavior in RF signals triggered by the onset of plasma ignition. Because the impedance of the plasma changes after ignition, the parameters of both RF match networks can also undergo drastic changes while attempting to adjust to the changing impedance. The nature of this transient is directly affected by the ability of the match networks to tune, and how they are reacting to each other, which in turn is a reflection of the state of the system. Specifically, Figures 3-3 and 3-4 contain plots of the load and tuning positions respectively, for an RF match network over a sixteen wafer run. Note how the transient behavior is visibly different for the baseline machine compared to a machine which was experiencing problems with its RF match network (this would be considered a failure mode). Thus, this observed transient, although it occurs during a transitional phase with a duration of less than a couple of seconds, contains important diagnostic information. However, with signals being collected at a frequency of about 1Hz, current data acquisition rates are insufficient in capturing this transient behavior.

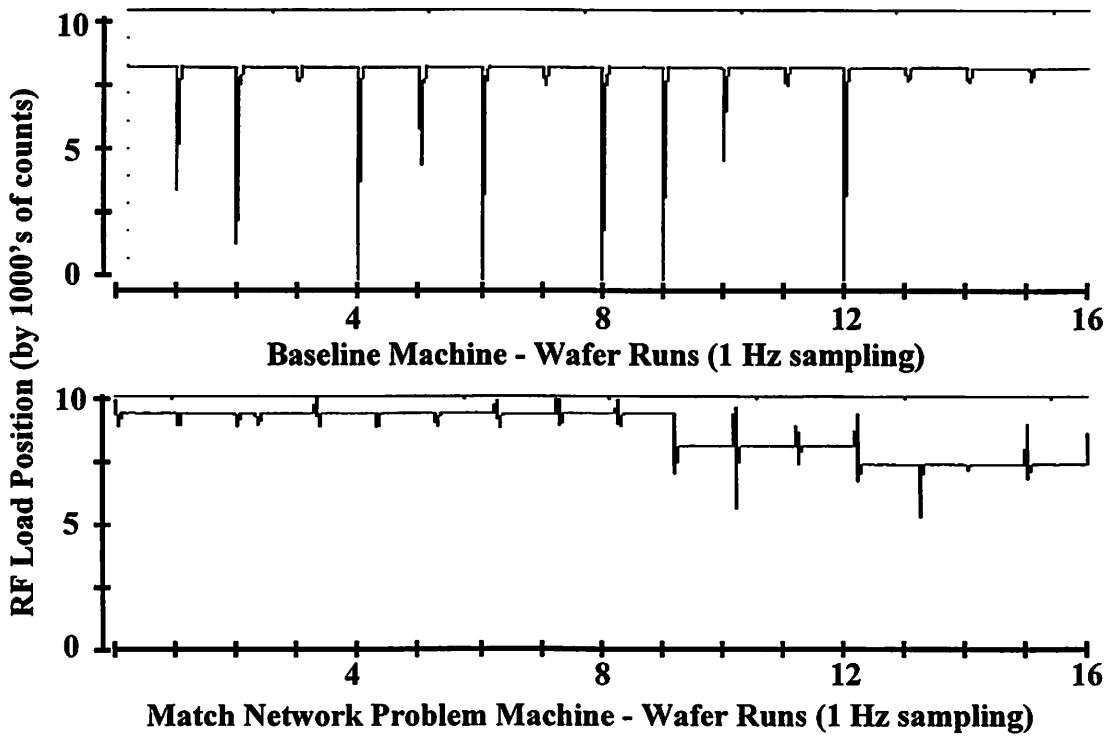


Figure 3-3. Transient behavior of bottom (RF) load position for a sixteen wafer run

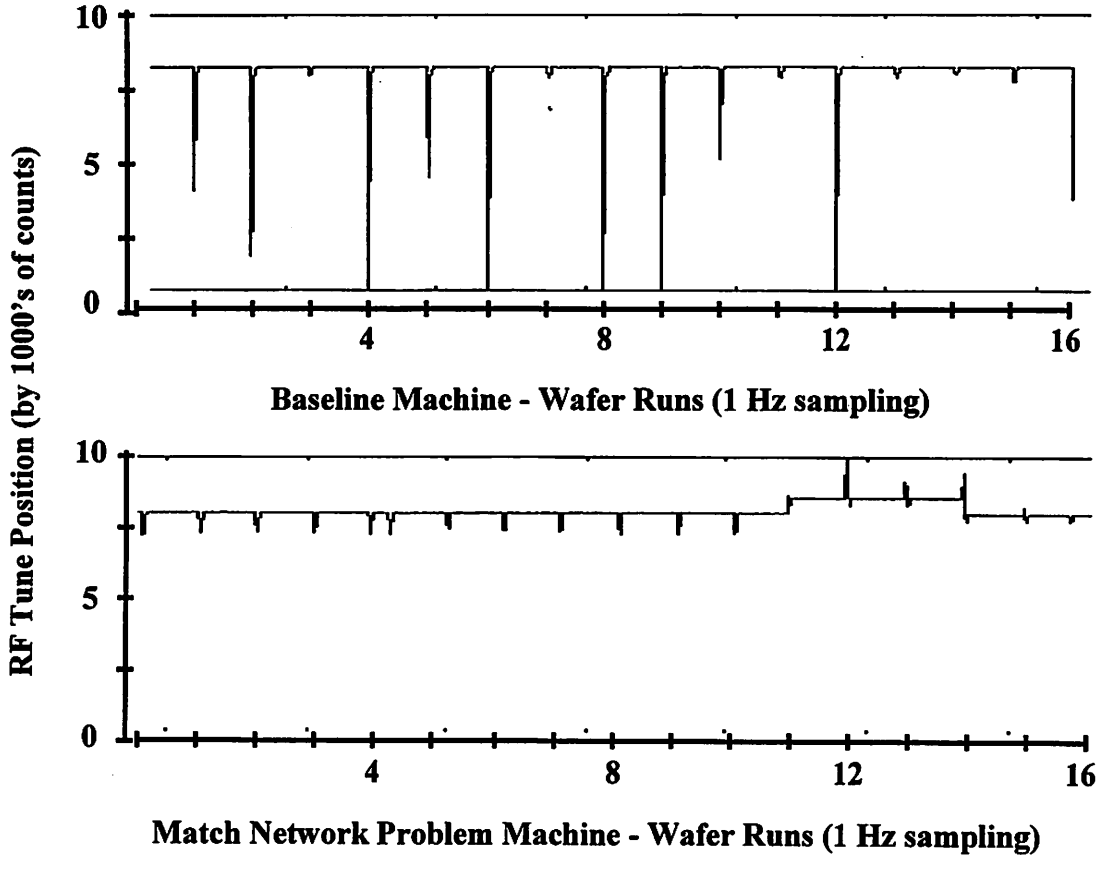


Figure 3-4. Transient behavior of bottom (RF) tune position for a sixteen wafer run

To further investigate this behavior and its potential use as a feature to characterize certain failure modes, we outfitted the Lam TCP 9400 plasma etcher in the Berkeley Microfabrication Laboratory, with instrumentation capable of collecting sensor signals at sampling rates that are much higher than the typical SECSII data acquisition rate. These high resolution signals are collected under various conditions of the tool, in particular, to simulate common failure modes that have been observed in field trials. Such an experiment enables us to explore the effect of varying RF network match parameters on the transient behavior. Once an accurate fingerprinting of the machine and its condition has been established, the performance of various modeling and diagnostic schemes can be compared.

3.4.1. Focusing on the Match Network Problems

A significant number of manufacturing problems occur due to failure in the match network. The purpose of this network is to match the impedance in the chamber such that the

reflected power back to the generator is minimized. The match network accomplishes this task by changing the positions of load and tune capacitors in response to the changing impedance of the chamber. The initial positions of the load and tune capacitors are determined by the conditions of the previous run. In other words, for a given wafer, after the plasma ignites, the network parameters (in this case, the load and tune capacitor positions) settle to a stable value to match the impedance, and conditions are met for the etching process to begin. These positions (where the network achieved a "matching" condition) are used as target values for the initial positions to process the next wafer.

Difficulties arise when the impedance of the chamber (after plasma ignition) changes between runs. This change can be caused by a variety of problems including machine aging, attributed to material deposited on the chamber, gas leaks, or differences on the wafer itself. If something does change the chamber impedance, the target values for the capacitor positions, which achieved a matching condition in the previous run will not be optimal for the current run, and we expect the values to change and settle in a new position to match the new chamber impedance.

Another problem associated with the match network involves the actual capacitors themselves. Capacitors may "bind", which means that although the computer may be instructing the capacitors to move, they are unable to change position. Thus, the situation may be that one capacitor is moving to adjust, while the other is stationary, and a matching condition may or may not be achievable depending on the circumstances.

Our experiment attempts to address these issues associated with problematic behavior in the matching network. First, because the adjustments to achieve a matching condition often take place in less than a few seconds, sensor data relating to RF network match parameters as well as other real-time sensor signals must be collected at an increased sampling rate of 100 Hz. Secondly, by varying the preset values for the positions of the tune and load parameters of the top match network, we simulate the condition of a "mismatch" to the chamber impedance. This also enables examination of the transient behavior of several signals in response to the adjustments being made by the match network. Finally, the

tune and load capacitors are disabled, one at a time, by loosening the connection to the driving motor.

Index	Fault Category	Tune Position (counts)	Load Position (counts)
1	baseline	target (+/- 1000)	target (+/- 1000)
2	HH extreme	32000	32000
3	LL extreme	0	0
4	HL extreme	32000	0
5	LH extreme	0	32000
6	HH midrange	+3000	+3000
7	LL midrange	-3000	-3000
8	HL midrange	+3000	-3000
9	LH midrange	-3000	+3000

Table 3-3. Categories defined by preset values for tune and load capacitor positions in match network

Table 3-3 shows the different fault groupings corresponding to preset (initial position) values for the tune and load capacitors. By convention, the capacitor positions are defined over a range of 0-32000 counts. The target value is based on the position for a match condition established in the previous run. Baseline (default value for position) is considered to be within 1000 counts of the target. We consider both high and low values referenced to the target value, for both tune and load. Extreme values are defined at the extremes of the ranges with high set to 32000 counts, and low set to 0 counts. Mid range values are defined at 3000 counts above target for the "high," and 3000 counts below target for the "low."

The actual capacitor positions for the first five categories in Table 3-3 are shown in Figure 3.5, where the presets are chosen at the extreme values of the range, and the arrows indicate the target value. These are plotted for the adjustment period (approximately six seconds), during which the positions are adjusting until they settle at a matching condition.

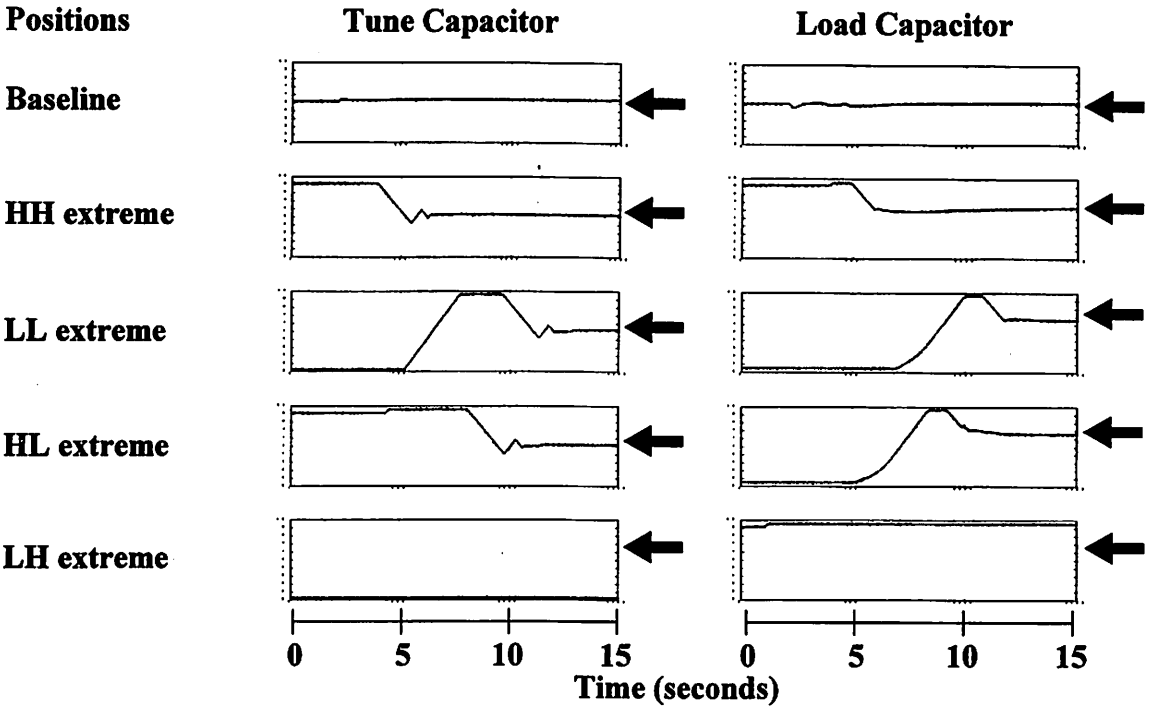


Figure 3-5. Capacitor positions for preset extreme values; arrows indicate target values

Figure 3.6 shows the impedance signal corresponding to the categories defined by the presets for tune and load capacitors. (Preset values for tune/load are superimposed on the same plot). Note that the preset conditions define our fault categories.

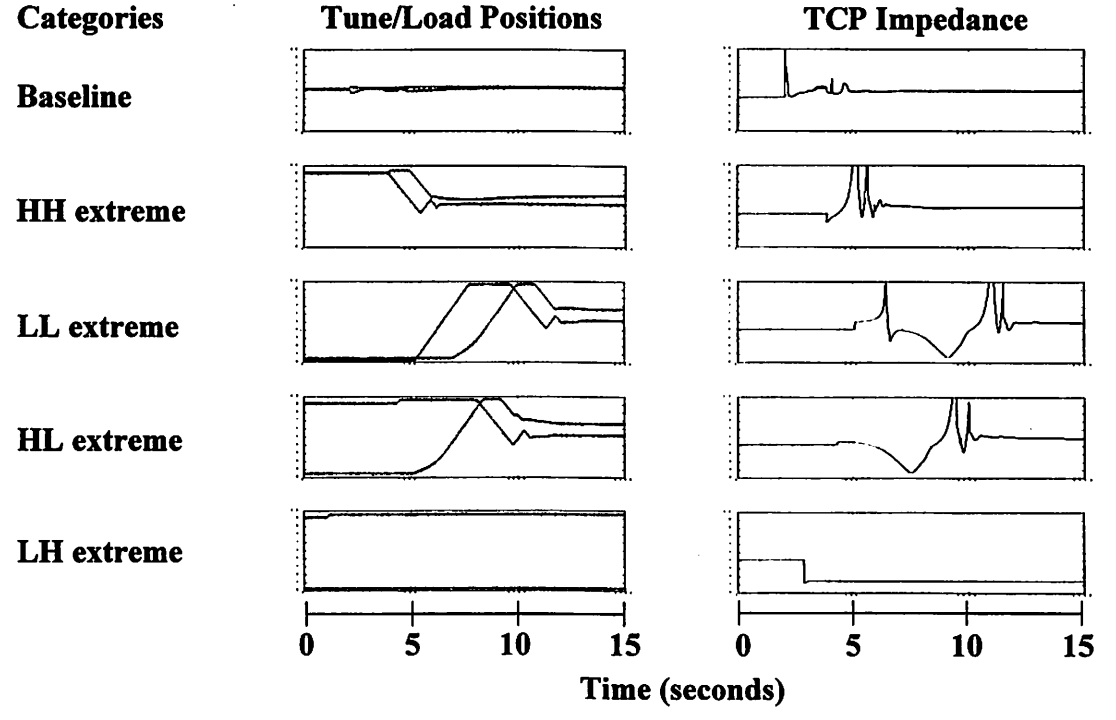


Figure 3-6. Impedance signal corresponding to categories defined by tune/load presets

The impedance signal is an example of transient behavior that appears to have some structure or pattern. Based on this structure, sensor signals can be grouped or classified, and we can draw a conclusion as to whether the preset values are low or high for the current chamber condition. This information is indicative of a change in the chamber impedance and can be used to predict whether a matching condition can be achieved. Finally, this provides some base behavior to compare the effect of capacitor binding, where only one capacitor is adjusting and the other is immobile.

3.5. Summary

In this chapter, we described the various types and sources of sensor data, as well as the experiments designed and conducted to produce them. Because our objectives are distinct in each case, with the focus of our study ranging from monitoring and fault detection, to fault diagnosis of diverse conditions, this naturally gives rise to differences in the data. It will soon be apparent that this also necessitates the employment of several modeling techniques, depending on the data, and that the development of a cohesive framework to merge disparate results together is a key contribution of this work. The next chapter addresses the issue of modeling, while the following chapter deals with the framework structure.

4 Modeling and Characterization of Long Term Behavior

4.1. Introduction

Real-time tool signals from etch equipment have been proven useful in modeling the plasma etch process and in providing a means of machine monitoring. In this chapter, the multivariate statistical control system, applied in the past to localized time scales (modeling the signal on a real-time, second by second basis, and wafer-to-wafer basis) is extended to deal with long term variability on a lot-to-lot basis. Long term trends in optical emission data collected from a plasma etch tool are characterized through data transformations and linear modeling techniques. By filtering the known effects of machine aging, these models facilitate the integration of optical emission data with other sensor signals, resulting in a fault detection system which is robust over time.

4.2. Background and Previous Work

Traditional statistical process control (SPC) techniques assume that the underlying process is stationary, i.e. that the mean and variance do not vary with time, and that the observations are identically, independently, and normally distributed (IIND) [10]. Assuming that these trends are present in data representing normal operating behavior, application of these techniques directly to machine data that contain trends results in increased false alarm and missed alarm rates. To avoid these increased false and missed alarm rates, past work used time-series modeling techniques to filter out the time dependent trends; traditional and multivariate statistical process control (SPC) methods were then applied to the resulting residuals to monitor the machine behavior. This system, known as real-time statistical process control (RTSPC), was shown in [11] to be effective in monitoring real-time

and wafer-to-wafer data. This investigation is motivated by the need to extend RTSPC to include long term variability on a lot-to-lot basis.

4.2.1. Statistical Process Control for Monitoring Data

In any production process there is inherent natural variability. This variation is considered common across all processes, and is attributed to noise in the system due to small, unavoidable causes, which are always affecting the process. A process is referred to as being in statistical control if its operation is affected only by such random, chance causes. Other sources of variability which plague processing include improperly adjusted machines, operator error, and defective raw materials. Disturbances to the process caused by these sources are usually large compared to background noise, and are often not randomly distributed. A process exhibiting a fluctuation caused by a non-random, well-defined event is considered to be out of control. In this case, the event is referred to as an assignable cause which shifts the process to an out of control state. A major objective of statistical process control is to detect shifts in the process state, and to find assignable causes so that corrective action may be taken.

Systems designed for SPC are used to monitor a process over time, ensuring that it remains statistically in control. In some sense, SPC techniques complement automatic feedback control methodology. The latter is also applied to reduce variability in the process, but uses a different mechanism to accomplish this goal. Feedback control seeks to compensate for the predictable component of a disturbance in crucial variables by adjusting other variables, effectively transferring the variability from important variables to less critical parameters. In contrast, SPC monitoring is applied on top of the process and its automatic control system to detect behavior which directly reflects the occurrence of a special event. In this case, the goal is to diagnose causes and eliminate them rather than to compensate for them. In this manner, long term improvements to the process can be achieved via changes in the system and operating procedures.

Many systems designed for SPC are based on a small number of variables usually associated with measurements on the final product. A control chart is an on-line graphical technique commonly used to track product quality variables, which are then examined one at a

time. For instance, the univariate Shewhart chart can be defined for monitoring variables by using a center line reflecting the average level for the parameter, and upper and lower control limits based on the natural variability in the process. However, as processes become more complex, the common manufacturing practice is to collect not only product data, but also process data, which often include measurements of many process variables. One major challenge is to extract relevant information from the growing mass of data to enable immediate action, preventing actual yield loss. Extracting information from large databases can be interpreted in different ways, (1) it can refer to the filtering or selection of the signals which give the most information, (2) it can refer to the combining of many signals to infer a conclusion, (3) it can refer to data compression, representing a large number of parameters with a smaller set, which captures most of the important features of the data.

4.2.2. Advantages of Multivariate Techniques

Multivariate statistical methods provide a powerful toolbox for extracting information from large databases in all three respects described above, leading to improved analysis, monitoring and diagnostic capabilities. In addition, there are other advantages to using multivariate techniques over univariate analyses. First, because the process variables reflect the state of the process, correlations exist among the different parameters. Examination of these variables one at a time treats them as if they are independent. For instance, a system built to detect and diagnose faults based purely on univariate models can only account for the magnitude of deviation in each variable, and is likely to produce false alarms or even miss true out-of-control situations. In contrast, multivariate techniques can extract information on magnitude of deviation and on directionality, accounting for how the variables behave relative to one another. This means that the multivariate test can be more powerful, where we define power as the probability of rejecting the null hypothesis when it is false (generation of a true alarm).

Figure 4-1 is a pictorial representation for the case of two variables showing the acceptance regions for the multivariate and univariate tests. Note that the acceptance region for the multivariate test is defined by an ellipse, with each point on the curve statistically equi-

distant from μ_0, demonstrating that distance is defined also in terms of direction, normalized by the covariance. The orientation and shape of the ellipse is determined by the correlation structure between the two variables. The two shaded regions demonstrate the benefits of multivariate testing which take this structure into account. These are cases where small effects in each variable fail to be significant if examined one at a time; however, when combined, the joint effect is significant. The lightly shaded region A shows the contribution to inflation of the false alarm error by the univariate test; the dark region B shows the contribution to greater power for multivariate testing.

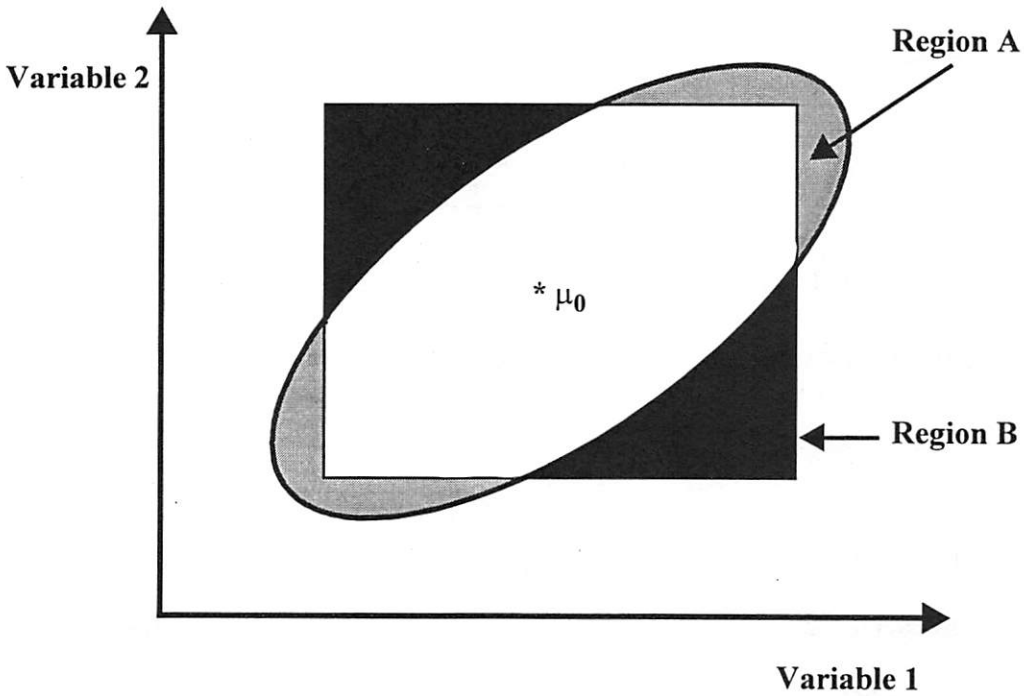


Figure 4-1. Comparison of univariate and multivariate testing acceptance regions

Region A: Accept multivariate test, reject univariate, when null hypothesis is true. Region B: Reject multivariate, accept univariate, when null hypothesis is false.

Because the quality of a product, or in the case of manufacturing, the state of a process, is defined not by each variable independently, but by the simultaneous values of all measured parameters, and because many of these parameters are often correlated, it makes sense to use multivariate techniques. To describe the multivariate testing used in this application, we first review the univariate case, then explain the extension to the multivariate process.

4.2.3. Hypothesis Testing for the Univariate Case

The idea of monitoring variables for the purpose of tracking the behavior of a system is formalized statistically by constructing a hypothesis test. The assumption is that a sample of n observations, $y_1, y_2, \ldots y_n$ is taken from a population distributed as $N(\mu,\sigma^2)$, where μ is the mean and σ^2 the variance. The mean, μ, is estimated by the sample average, $\bar{y}$; σ^2 is estimated by the sample variance, s^2.

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{4-1}$$

$$s^2 = \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{n-1} \tag{4-2}$$

To test the hypothesis that the mean, μ, is equal to a given value, μ_0, $H_0{:}\mu = \mu_0$ vs. $H_1{:}\mu \neq \mu_0$, the t-test uses the following test statistic:

$$t = \frac{\bar{y} - \mu_0}{s/(\sqrt{n})} \tag{4-3}$$

This statistic is distributed as t_{n-1} if the null hypothesis is true. We reject the null hypothesis if $t \geq t_{\alpha/2,\, n-1}$, where $t_{\alpha/2,\, n-1}$ is a critical value from the t-table. The expression in Equation 4-3 is known as the characteristic form of the t-statistic, representing a sample standardized distance between $\bar{y}$ and μ_0.

4.2.4. Extensions to the Multivariate Case

For the case where p variables are measured for each sample, the assumption is that we have n samples. We now have two indices, i and j, corresponding to the sample and the variable index respectively. Hence, we might have n samples, $y_{1,j}, y_{2,j}, \ldots y_{n,j}$ from a multivariate population $N_p(\mu,\Sigma)$ such that each $y_{i,j}$ contains p measurements on the ith observation. The $p \times 1$ mean vector, μ, is estimated by the sample average vector, $\bar{y}$; Σ is estimated by the sample covariance matrix, S. This $p \times p$ matrix contains the sample vari-

ances on the diagonal, and the sample covariances for all possible pairs of p variables on the off-diagonal.

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1p} \\ s_{21} & s_{22} & \cdots & s_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ s_{p1} & s_{p2} & \cdots & s_{pp} \end{bmatrix} \tag{4-4}$$

$$s_{ii}^2 = s_i^2 = \frac{\sum_{k=1}^{n} (y_{ki} - \bar{y}_i)^2}{n-1} \tag{4-5}$$

$$s_{ij}^2 = \frac{\sum_{k=1}^{n} (y_{ki} - \bar{y}_i)(y_{kj} - \bar{y}_j)}{n-1} \tag{4-6}$$

To test the hypothesis that the mean vector, $\bar{\mu}$, is equal to a given value, $\bar{\mu}_0$, $H_0: \mu = \mu_0$ vs. $H_1: \mu \neq \mu_0$, the extension of the univariate t-test is obtained by rewriting the univariate t as:

$$t^2 = n\frac{(\bar{y} - \mu_0)^2}{s^2} = n(\bar{y} - \mu_0)(s^2)^{-1}(\bar{y} - \mu_0) \tag{4-7}$$

By replacing $\bar{y} - \mu_0$ and s^2 by $\bar{\mathbf{y}} - \mu_0$ and **S**, the following test statistic is obtained:

$$T^2 = (\bar{y} - \mu_0)^T S^{-1} (\bar{y} - \mu_0) \tag{4-8}$$

The distribution of T^2 was first described by Hotelling (1931), and is indexed by the number of variables p and the degrees of freedom n - *1*. Extensions of the univariate Shewhart control charts to multivariate control are based on Hotelling's T^2 statistic. As $n \to \infty$, the T^2 approaches the χ^2 distribution.

The density of T^2 is skewed; the lower limit is zero and there is no upper limit. An upper control limit for T^2 can be found using the conversion of the T^2 statistic to an F statistic:

$$\frac{n(n-p)}{p(n+1)(n-1)} T^2_{p,n-1} = F_{p,n-p} \tag{4-9}$$

To find the upper control limit, one would use the upper $100\alpha\%$ critical point of the F-distribution with p and n - p degrees of freedom.

4.2.5. Time Series Modeling

Regardless of whether the testing is univariate or multivariate, to apply SPC techniques, the hypothesis is based on an IIND assumption. However, due to the nature of processing, sensor signals follow various time trends. When the value of a data point in a sequence is dependent on the value or values of data preceding it, the trend can be captured by a time series model. The most general form of this model is described by an autoregressive integrated moving average model, or *ARIMA (p,d,q)* model, where p is the autoregressive order, d is the integration order, and q is the moving average order:

$$w_t = \sum_{k=1}^{p} \phi_k w_{t-k} - \sum_{k=0}^{q} \theta_k a_{t-k}\ ,\ a_t \sim N(0, \sigma^2),\ \theta_0 = -1 \tag{4-10}$$

$$w_t = \nabla^d x_t,\ \nabla^1 x_t \equiv x_t - x_{t-1},\ \nabla^2 x_t \equiv \nabla^1 x_t - \nabla^1 x_{t-1} \tag{4-11}$$

The difference operators in Equation 4-11 are applied on the original data series, x_t. The two examples show the first and second difference operators, respectively. The differenced data is represented by w_t in Equation 4-10, where ϕ_k are the autoregressive parameters, θ_k are the moving average parameters, and a_t is the prediction error, assumed to be IIND.

Modeling the signals as a time series accomplishes two things: (1) characterization of the process, with a means of predicting future values or behavior and (2) filtering out systematic trends so that what remains is due to random noise. SPC techniques can be applied to the resulting residuals once the time dependent pattern has been filtered by the models.

4.2.6. Real-Time Statistical Process Control (RTSPC)

A key step to successful monitoring for fault detection is to effectively characterize the process for prediction purposes and so that patterns due to normal operation can be filtered out. Figure 4-2 displays a flowchart of the steps taken to process the sensor data for use in real time statistical process control.

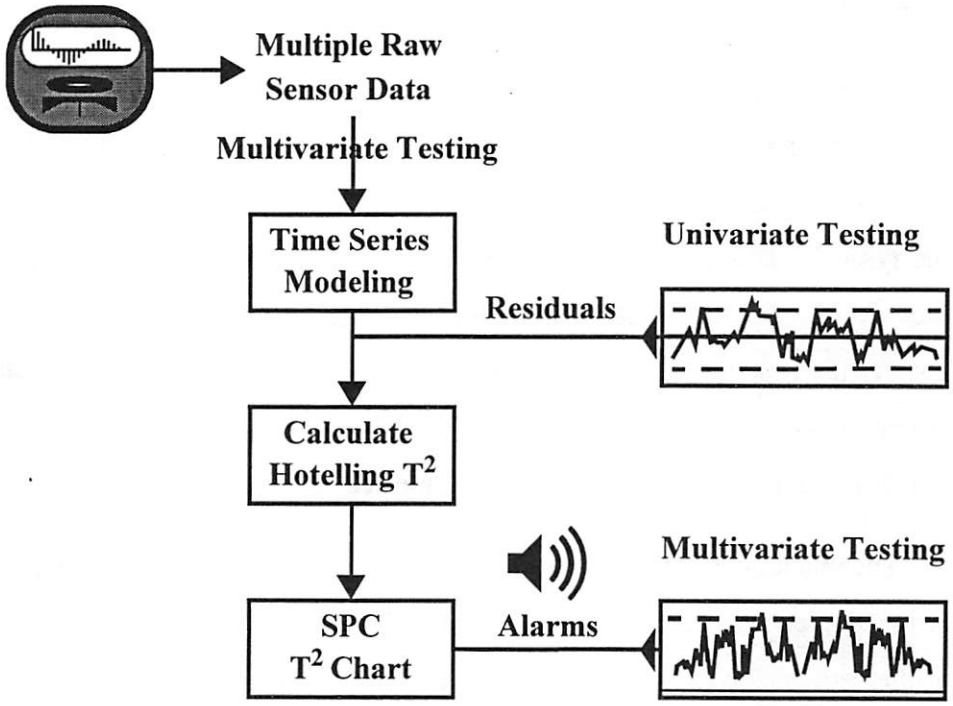


Figure 4-2. Flowchart for sensor data for monitoring and fault detection using SPC

ARIMA models to filter trends from the various sensor signals are built using sensor data obtained during normal operating behavior of the machine (baseline data). If the models are formulated appropriately, the resulting residuals resemble IIND random variables. These residuals can be monitored separately with univariate testing using Shewhart control charts. However, because the signals are measurements of the same physical process, there are cross-correlations among the different signal residuals. To account for these cross-correlations, the Hotelling T^2 statistic is used to combine the individual IIND residuals into a single statistical score. These scores are plotted on a T^2 chart; values exceeding the upper control limit generate alarms which signal the detection of a fault.

Note that this flowchart describes the construction of models used as a point of reference for comparison with incoming data sets from which we wish to draw conclusions. In other words, we must initialize the fault detection system using a set of baseline data meant to represent normal operating process behavior. Once the models are built and the SPC limits established from the baseline, we are in a position to monitor incoming production data, using the baseline models as a standard for comparison.

4.2.7. Evolution of Different Time Scales - Data Decomposition

Due to the nature of processing, the evolution of the system as described by the profile of the sensor signals over time can be viewed at different time scales. Because the signals are typically collected using a sample rate of 1-2 Hz, we monitor the data in "real time". However, if we are interested in drifts in the process, or abrupt shifting behavior caused by a specific event, these types of changes are more evident in large time scales, such as from wafer to wafer, or from lot to lot. By decomposing the data, we are able to monitor and detect faults in the process at different time scales. This is described below.

The machine data is comprised of a sequence of lots, each containing a series of wafers, with samples taken at 1 Hz for each wafer. Thus, we can decompose the signal by looking at the sequence of average signal values of each lot, and the sequence of average signal values of each wafer over time (adjusting for the lot effect by subtracting the average value for the lot containing the wafer). Similarly, the real-time signal is adjusted by subtracting the appropriate wafer and lot averages. In this way, the total signal is the sum of the lot average, wafer average, and real time signals.

The decomposition, shown pictorially in Figure 4-3, allows us to track and analyze each signal separately. This is important because different events affect the signals at different time scales. A fast equipment fluctuation due to changing chamber dynamics is visible in real-time, but may not affect the average value over the wafer, and certainly will not be detected in the average lot value. A machine drift or abrupt shift due to a problem instigated by an anomaly on a wafer, or by a problem in the conditions during one wafer run, is likely to be exhibited in the wafer average signal, but not necessarily in the real-time or lot average signals. Finally, long term drifts due to machine aging, or shifts in the machine

state due to chamber cleaning or preventative maintenance are visible in the lot average signals, while the real-time and wafer average levels show no significant change.

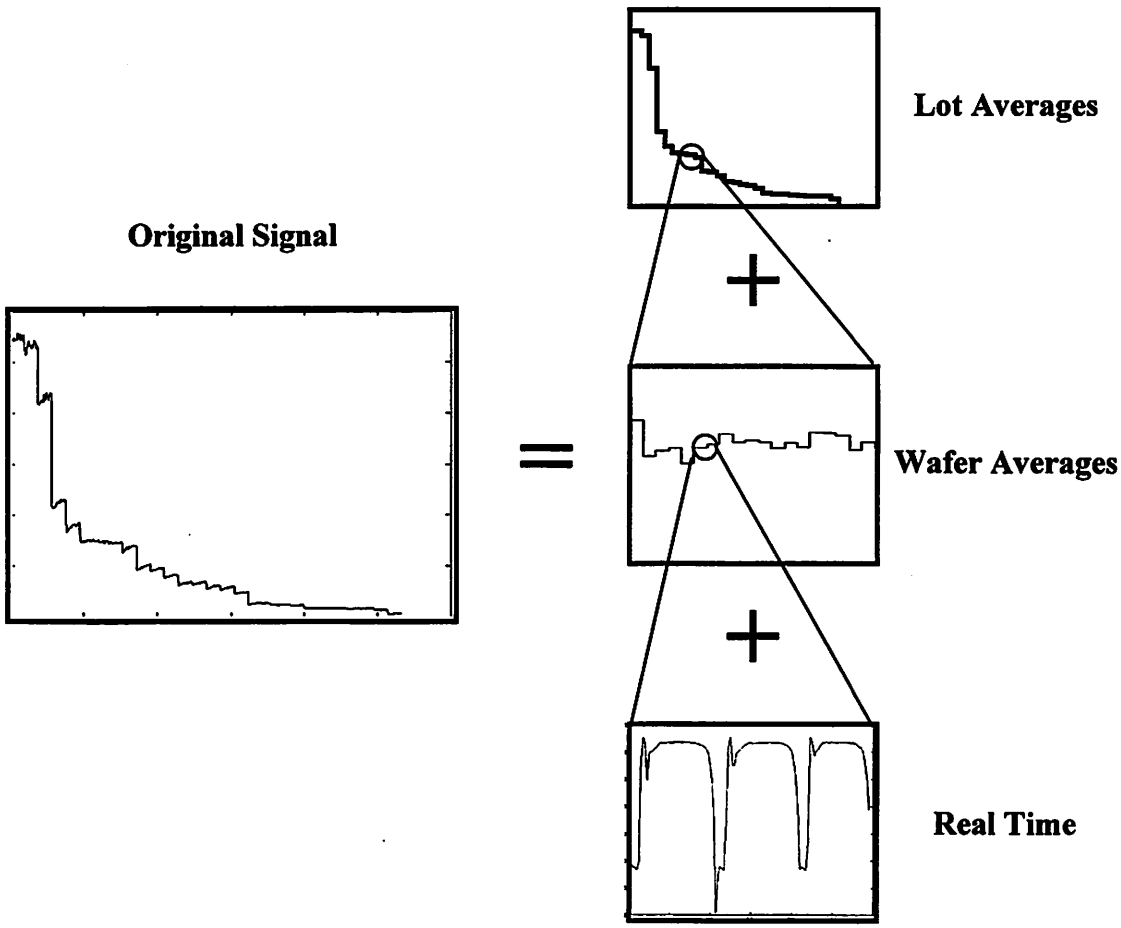


Figure 4-3. Data decomposition for an optical emission endpoint signal

Original signal plots data over one maintenance cycle. The beginning of a cycle is marked by a preventative maintenance event. The data are the sum of lot averages, wafer averages (shown here over one lot), and real time signals (shown here for three wafers).

4.3. Modeling Machine Aging

As wafers are processed, the state of the machine changes over time. This behavior is often referred to as machine aging; the drift in the machine state is associated directly with the accumulation of residue on the chamber walls and window.

4.3.1. Optical Emission Data

Recent efforts have focused on using optical emission data as a valuable source of information about the plasma state. However, measurements of this type exhibit atypical trends due to the confounding effect of window clouding and machine aging. This behavior is cyclical in the sense that the machine state can be "reset" by preventative maintenance (PM) events. This cycle of long term trends, when not properly taken into account, can result in an increased false alarm rate during fault detection. Models are developed, which characterize the behavior of optical emission signals over long periods of time. These models enable integration of these signals with other sensor data, so that real-time statistical process control techniques can be applied to perform fault detection. By specifically accounting for long term trends, these models partially decouple the machine state from the state of the plasma; such decoupling reduces the false alarm rate due to preventative maintenance events, thus resulting in a fault detection mechanism which is robust over time. A further advantage of this decoupling is that knowledge of the machine state in terms of aging can be combined with other information sources to provide prediction of equipment problems, and for scheduling preventative maintenance events. Machine state information combined with a more accurate knowledge of the true plasma state, after the effects of machine aging on the optical emission data have been removed, can also be better used to predict wafer output characteristics.

4.3.2. Long Term Trends

Examination and analysis of optical emission data over long periods of time shows a different type of trend than that typically handled by time series models. As depicted in Figure 4-3, the endpoint signal (a measure of the intensity of the plasma for a particular wavelength) exhibits an exponential decay. Figure 4-4 plots the average value of the endpoint taken over each lot with respect to the wafer count. Because the wafer countwafer count is reset to zero after a preventative maintenance (PM) event, the plot shows the endpoint signal evolving over the course of a maintenance cycle, where the chamber is initially clean but becomes progressively dirtier as more wafers are processed. The trend is clearly visible, and is repeatable as demonstrated by the five different maintenance cycles which

are overlaid in this plot. The data shown in Figure 4-4 span a total period of eight months, during which there were five PM events corresponding to chamber and window cleans.

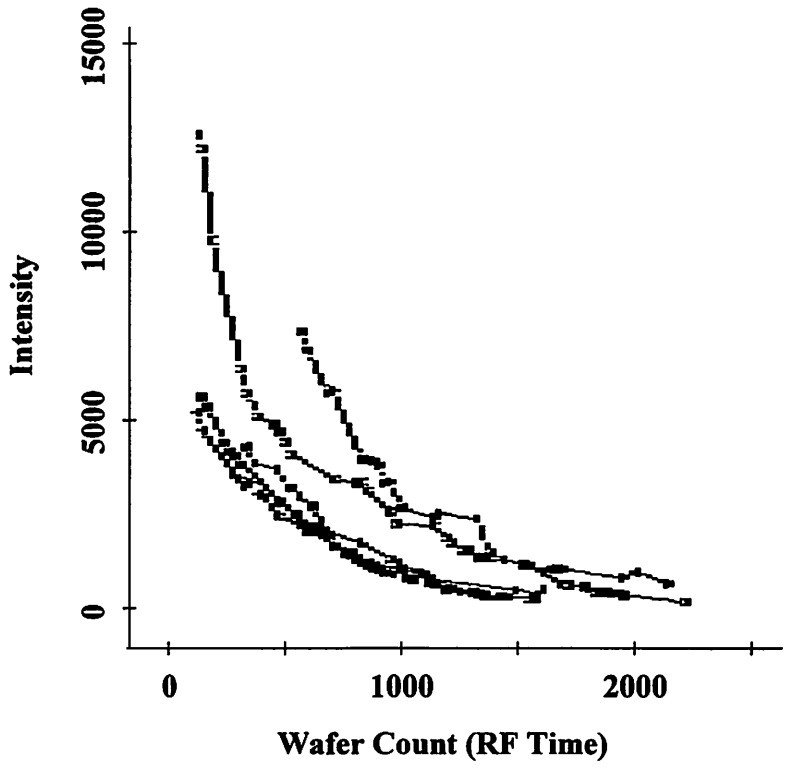


Figure 4-4. Lot averages of endpoint for five preventative maintenance (PM) cycles

4.3.3. The Effect of Window Clouding

Time series models are known to capture the dependencies among a sequence of data points, with the assumption that these readings are taken at regularly spaced intervals. However, because the processing of lots is rarely scheduled at such regular intervals, these models are inappropriate for dealing with optical emission data at long time scales. The problem is further complicated by the apparent exponential decay in the measured values.

The exponential decay visible in the lot average value of the endpoint signal suggests the use of the log transform as a method of linearizing the data. This is further supported by knowledge of the plasma etch process and its effects on the reading of optical emission data. Specifically, the chamber window becomes clouded as a result of progressive depos-

iting of material on the window surface as the wafers are being etched. This clouding in turn affects the sensor reading of the plasma intensity.

Mathematically, the plasma intensity measurement may be modeled by the following equation:

$$I(z) = I_0 e^{-\alpha z} \tag{4-12}$$

where the intensity, I, decreases exponentially with the thickness (z) of the deposited material. The exponential decay constant (α) is related to the absorption properties of the material. Assuming that the accumulation of deposited material varies as a linear function of time,

$$z = z_1 + z_2 \bullet RFtime \tag{4-13}$$

the expression for measured intensity as a function of RF time becomes:

$$I(RFtime) = I_0 e^{-\alpha z_1} e^{-\alpha z_2 \bullet RFtime} \tag{4-14}$$

Taking the logarithm of equation 4-12 results in a linear expression relating the log of the intensity to RF time.

4.4. Filtering Long Term Trends for Enhanced Monitoring Capability

To extend the monitoring system and fault detection capability (RTSPC) to accommodate lot-to-lot trends, the optical emission data are first filtered through a log transformation, and then modeled using linear regression techniques, followed by time-series modeling to remove the remaining time-dependent behavior.

4.4.1. Linearization of Optical Emission Data

The linear regression model uses wafer count as an input parameter in order to account for the effect of RF time. Figure 4-5 depicts the transformed data from Figure 4-4 for the five maintenance cycles. As expected, the transformation has linearized the data. After the linear trend is filtered out, the resulting linear regression model residuals are filtered using time-series models in order to remove the remaining time dependencies.

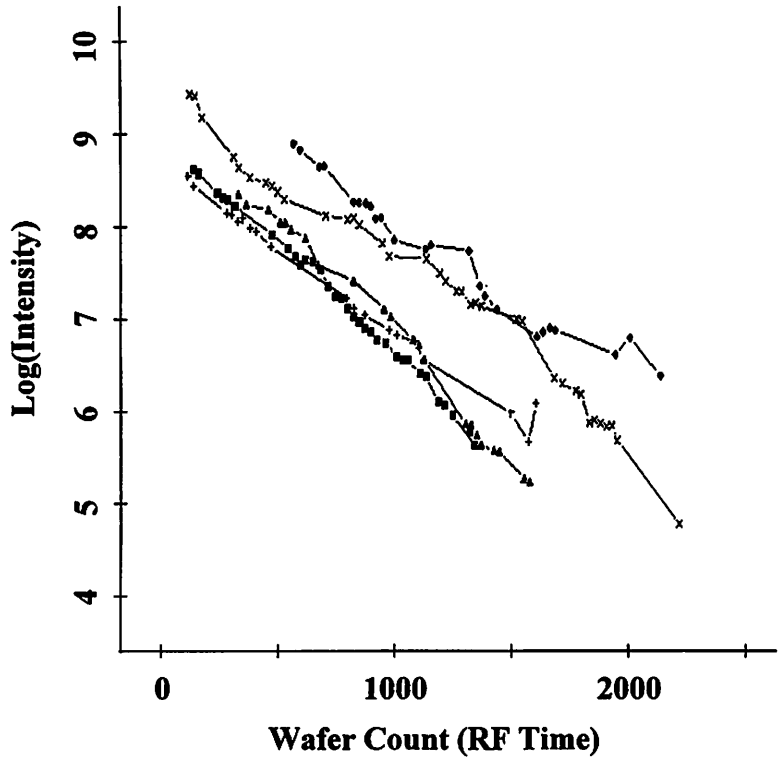


Figure 4-5. Lot averages of transformed endpoint for five preventative maintenance (PM) cycles

Note that a linear model had to be fitted to specifically account for wafer count or RF time. Had we not used this as a fitting parameter, the profile would have been skewed. This is because of the nature of the process, and the data available for analysis. Although wafers are processed at fairly regular intervals, and samples in real time are collected at a specific frequency, the processing of lots follows a much more sporadic schedule. It is not unusual to encounter long periods between datasets corresponding to successive lots. Thus, when tracking the long term behavior of the process, time-series models cannot be applied directly, as the assumption of regular sampling does not hold in this case.

Figure 4-6 summarizes the filtering process for the optical emission signals. We include plots of the residuals after each step, with histograms corresponding to their distribution. Note the spread of the residuals after being filtered by the linear model, versus the distribution after both linear and time series filtering. The residuals are cleaner and more tightly centered around zero, demonstrating the additional benefit of time series modeling.

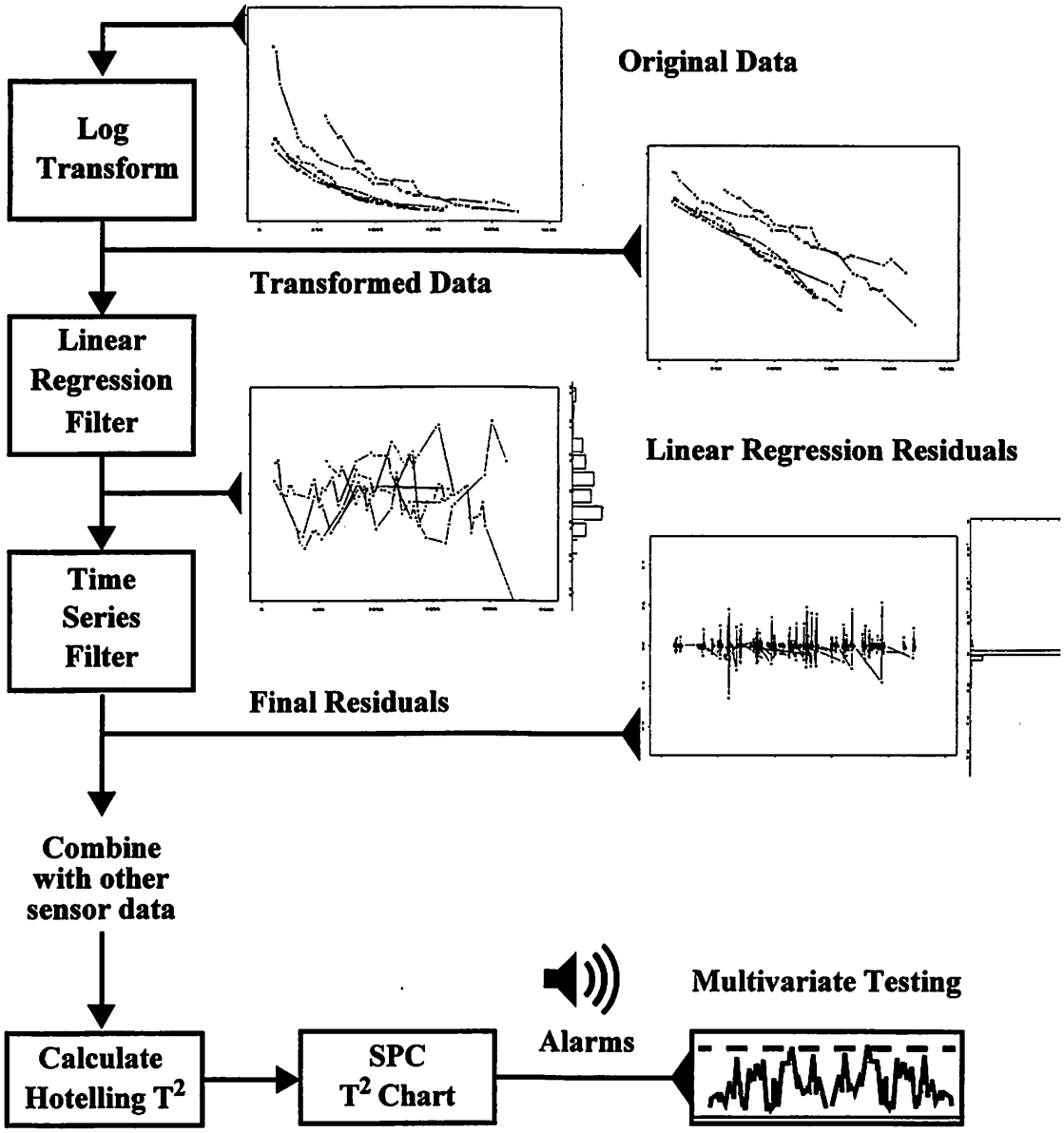


Figure 4-6. Filtering process for optical emission signals (OES)

4.4.2. Improved Fault Detection

The double T^2 chart for one maintenance cycle is shown in Figure 4-7. This plot was generated using only a time series model constructed from the original lot averages of the endpoint signal as a filter. The analysis used baseline data, and yet the model produced false alarms (dark bars) at the beginning of the cycle. Examination of individual signal residuals shows that the problem is indeed caused by the failure of the time-series model to accurately represent the apparent exponential decay in the endpoint signal.

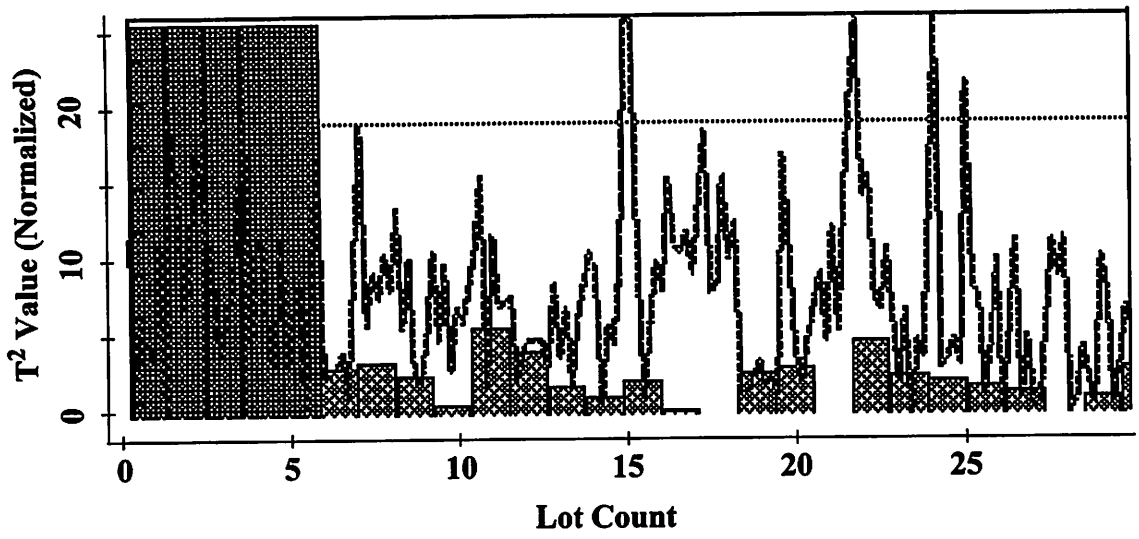


Figure 4-7. Baseline double T^2 chart using original data

Figure 4-8 depicts a similar double T^2 chart after the log transformation, followed by linear regression and time-series filtering as described above. The plot shows that the false alarms due to the decay have been eliminated, and thus, the models have effectively captured the long term trend.

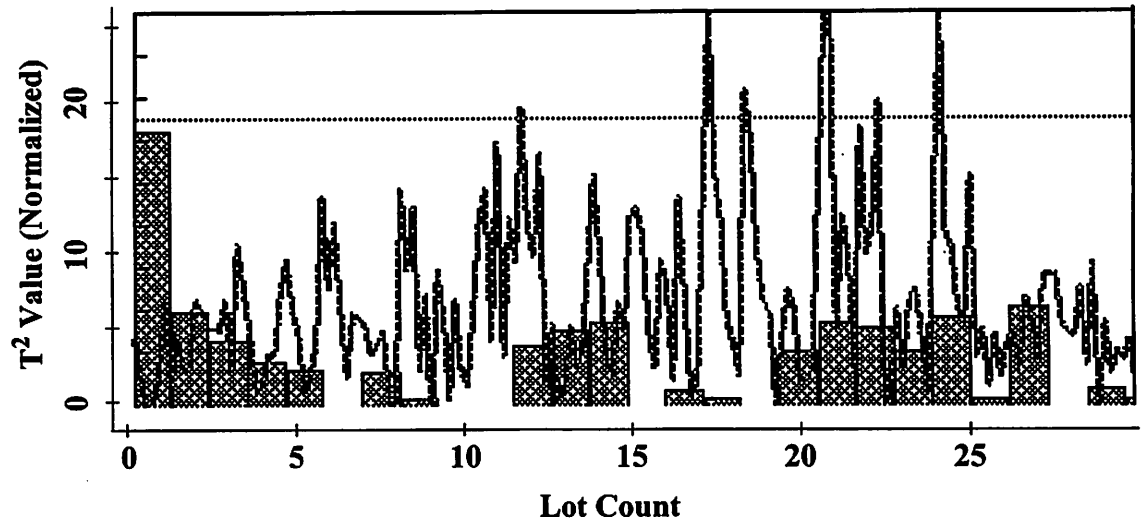


Figure 4-8. Baseline double T^2 chart using transformed data

Figure 4-9 is a plot of the T^2 chart of production data with known injected faults. The figure demonstrates that the known faults were detected on a lot-to-lot basis.

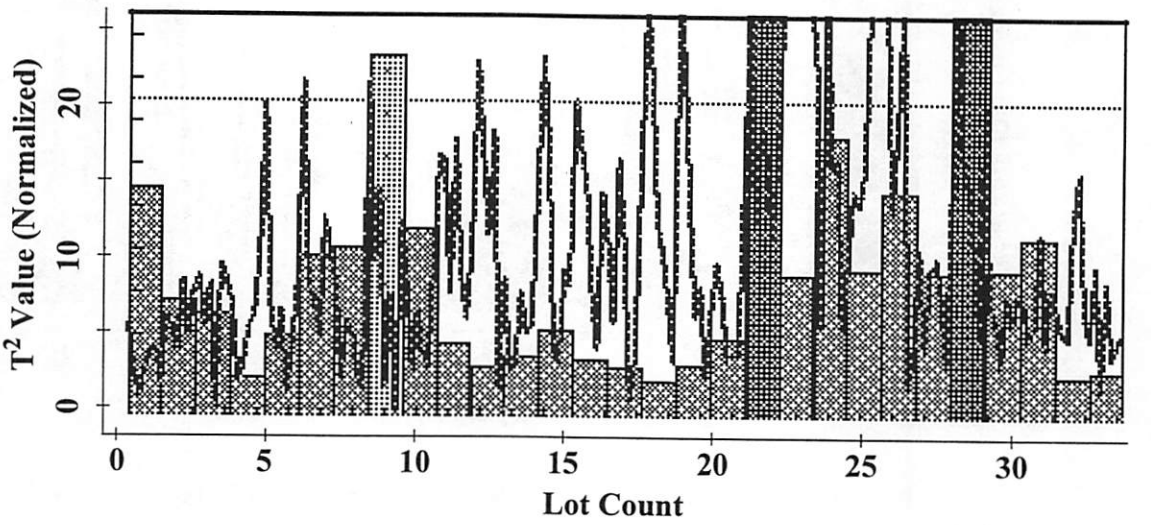


Figure 4-9. Production double T^2 chart using transformed data

4.4.3. Fault Detection Case Study

As a test bed for the improved RTSPC, we are using data taken over a period of seven weeks during which there were two window cleans. The data include two different recipes, comprising 40 lots total, with 19 wafers per lot, where pressure is the altered variable in the recipe. The events affecting the normal evolution of the system are DOEs and resetting of the throttle valve. These two events are very different in nature. We expect the DOEs to be fairly easily detected as an abrupt shift since the inputs to the machine are being varied drastically from wafer to wafer over a wide range of operating conditions. In contrast, resetting the throttle valve is a physical change to the machine, and thus is altering the machine's state. This could be exhibited as a subtle change, which may affect some sensor signals more than others. In addition, this change may be more pronounced in one recipe than another, since the process state is dependent on the input settings to the machine.

In the data set collected corresponding to the first recipe, the models constructed to represent the baseline condition include lots processed after the throttle valve had been reset. In this case, the T^2 chart did not produce alarms for these lots. After the baseline model had been established, production data (comprised of baseline data, plus lots processed as

DOEs) are analyzed following the flowchart outlined in Figure 4.2. The resulting double T^2 scores corresponding to wafer and lot level time scales for the baseline and production cases are plotted in Figures 4-10 and 4-11.

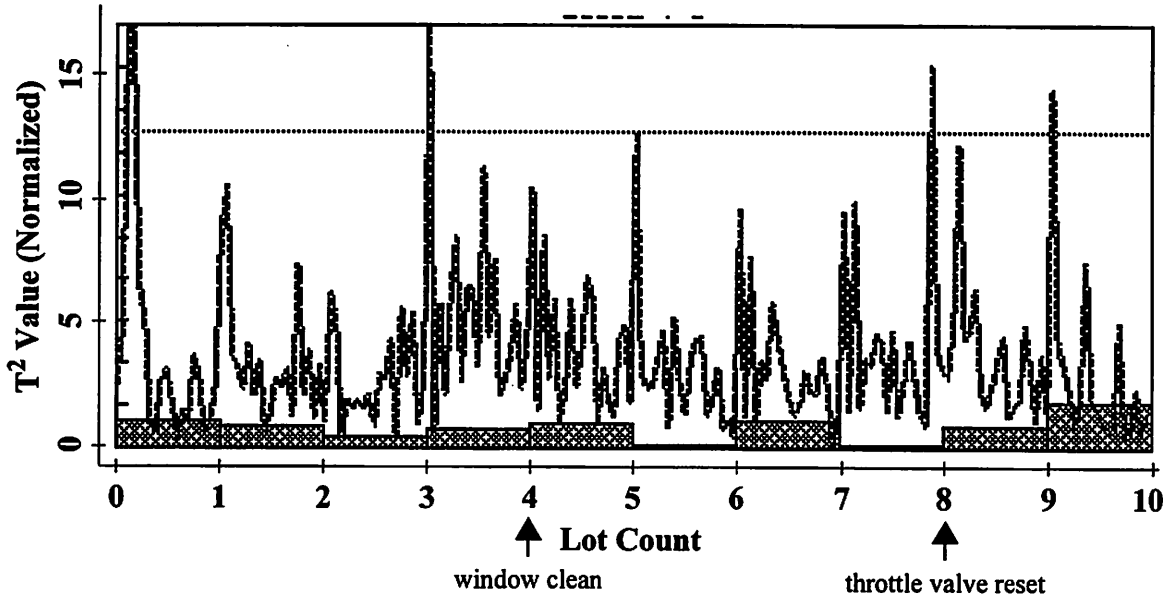


Figure 4-10. Baseline double T^2 chart using recipe 1 data [1]

Arrows indicate window clean and throttle valve reset.

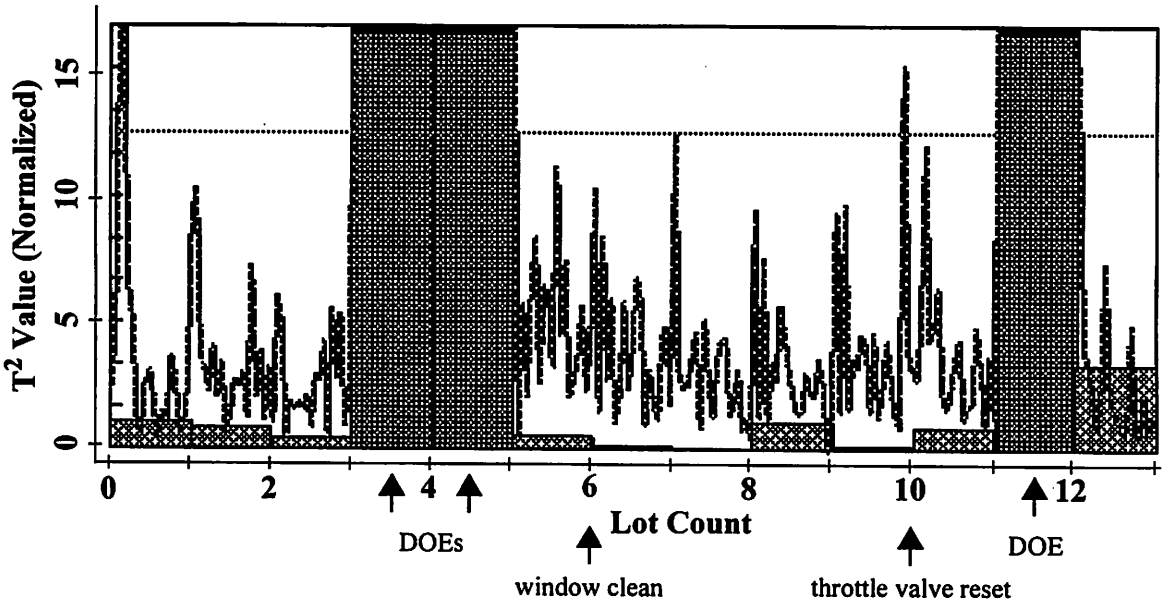


Figure 4-11. Production double T^2 chart using recipe 1 data

Arrows indicate injected faults, which in this case, are lots corresponding to DOEs.

[1] In this case, the lot to lot control limits appear to be inflated.

For this recipe, alarms are clearly generated for the DOE lots; the violations of the T^2 upper limit are present in both wafer level and lot level time scales, suggesting that the injected changes are drastic enough to affect the average value over both the individual wafers and the entire lot. However, as mentioned above, the resetting of the throttle valve did not signal a fault in this dataset. Close examination of the individual residuals for some of the sensor signals reveals that the only significant change in the lots processed after the throttle valve position change appears to be in the chamber pressure. The other sensor signals seem unaffected by this particular event. Figures 4-12 and 4-13 plot the production data for the chamber pressure and RF Tune signals respectively.

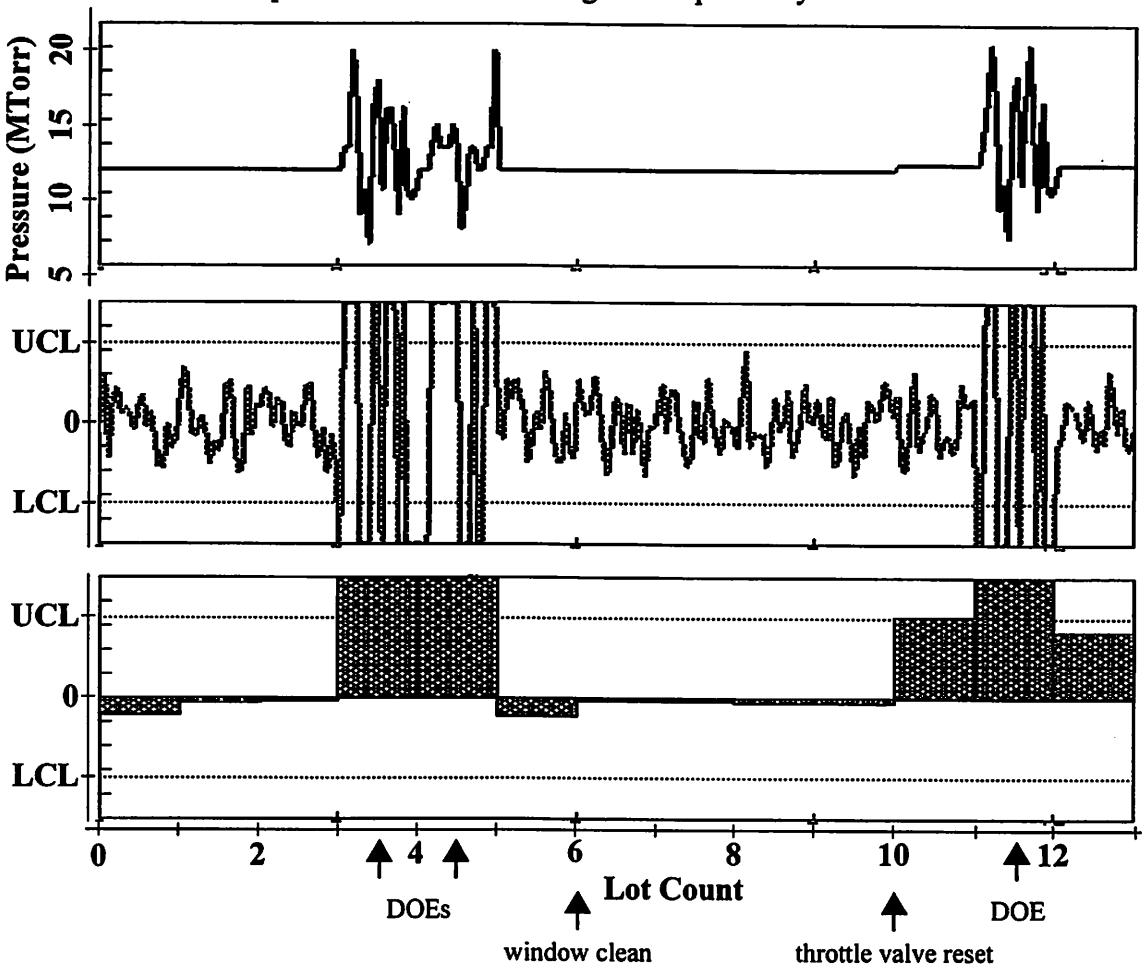


Figure 4-12. Univariate analysis for chamber pressure signal for recipe 1 data [2]

Original signal (top); Wafer-to-wafer level residuals after time-series filtering (middle); Lot-to-lot level residuals after time-series filtering (bottom). Upper and lower control limits (UCL and LCL) are shown for the residual plots.

[2] Again, the lot to lot control limits appear to be inflated in this case.

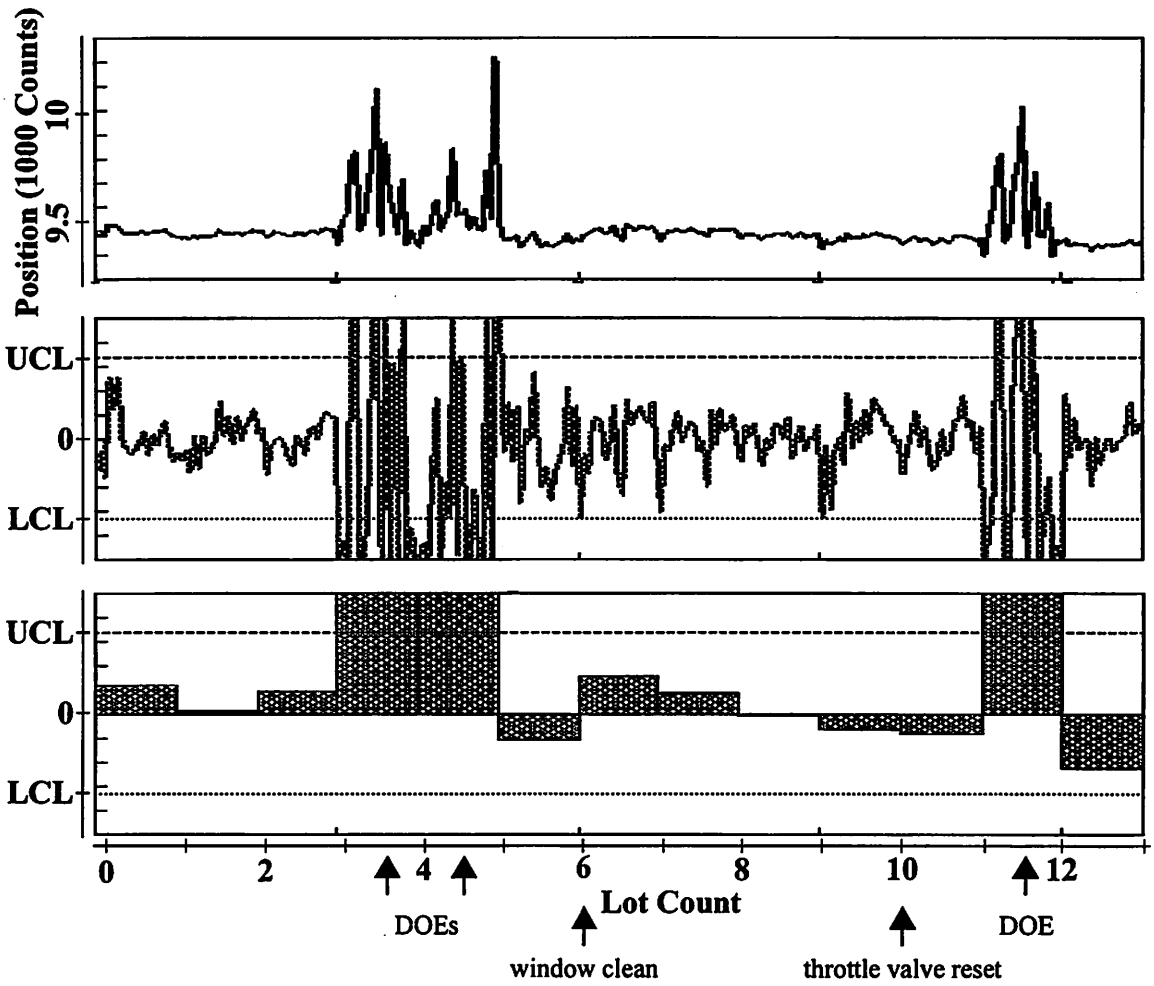


Figure 4-13. Univariate analysis for RF tune position signal for recipe 1 data

Original signal (top) is measured in thousands of counts; Wafer-to-wafer level residuals after time-series filtering (middle); Lot-to-lot level residuals after time-series filtering (bottom). Upper and lower control limits (UCL and LCL) are shown for the residual plots.

Analysis of the data taken from the second recipe produced different results. First, no DOE data are available for analysis using this recipe as a baseline. Secondly, models constructed to represent the baseline condition do not include lots processed after the throttle valve change. In fact, the inclusion of these lots generates alarms on the T^2 charts. Thus, the baseline data exclude these lots; instead they are added to the production data set. The events affecting the data collected under this recipe are resetting of the throttle valve, and a large change in RF power after processing of the first two lots. The double T^2 charts for baseline and production data are displayed in Figures 4-14 and 4-15 respectively.

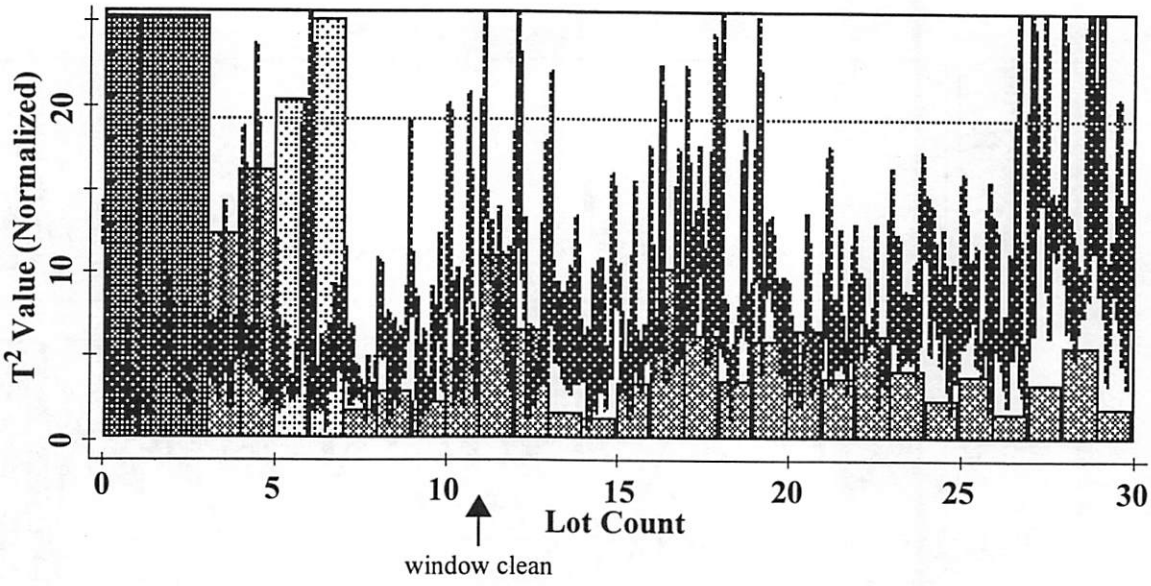


Figure 4-14. Baseline double T^2 chart using recipe 2 data

Arrow indicates window clean.

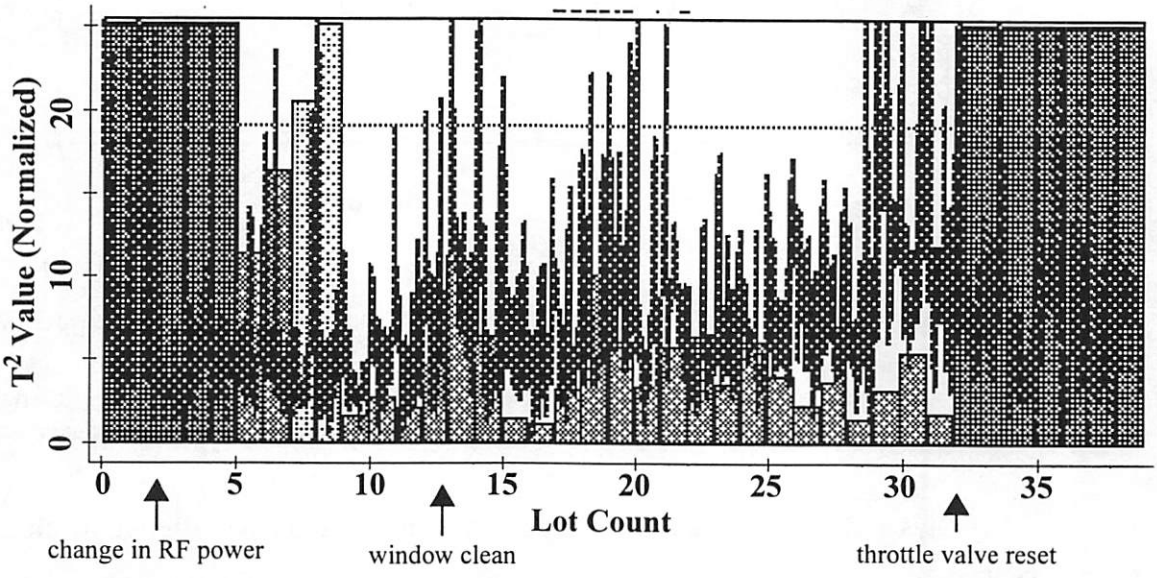


Figure 4-15. Production double T^2 chart using recipe 2 data

Arrows indicate faults, which in this case, are change in RF power, and throttle valve reset.

Examination of the individual sensor signals and residuals adds further insight for assigning cause to the alarms generated by the lots with high T^2 scores in Figure 4-15. Figures 4-16 to 4-18 plot some of the sensor signals and corresponding wafer and lot level time scale residuals.

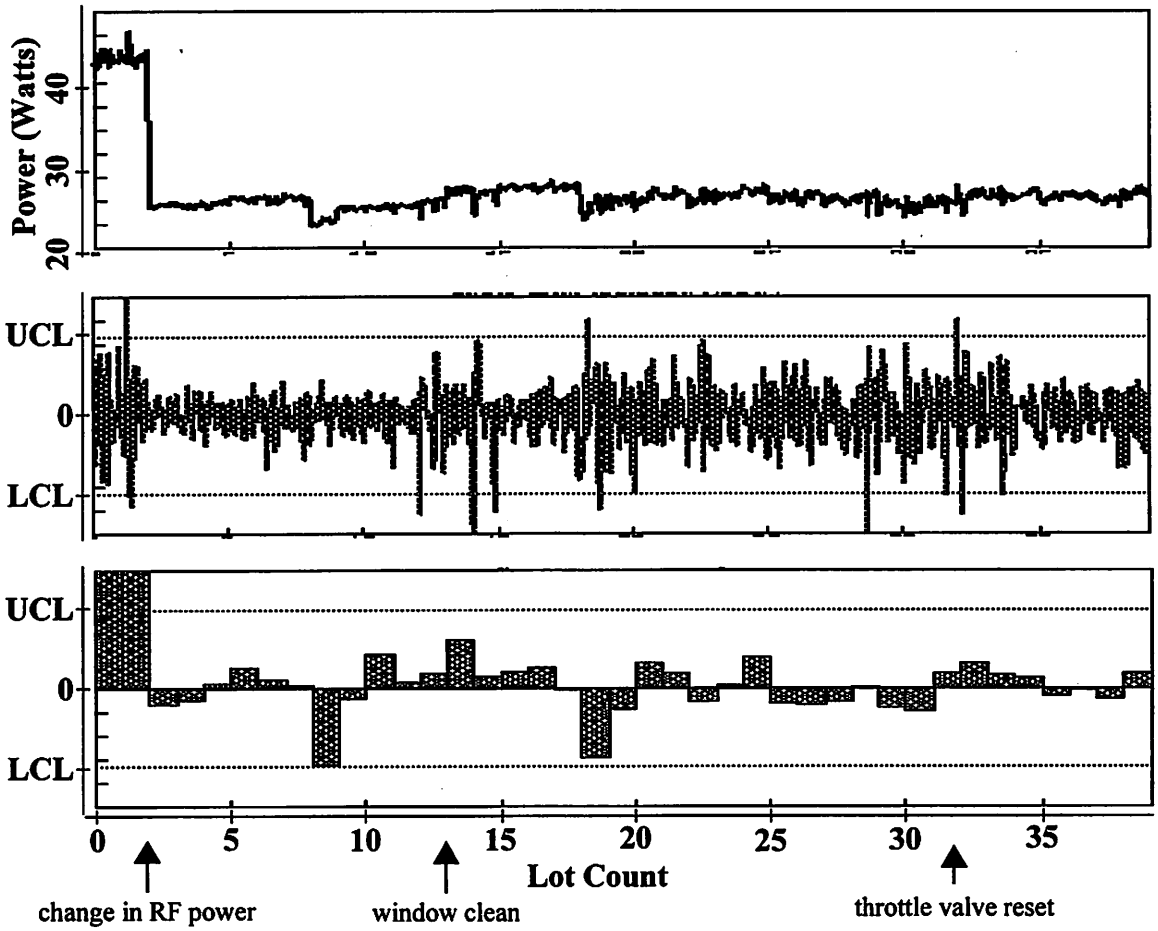


Figure 4-16. Univariate analysis for RF power signal for recipe 2 data

Original signal (top); Wafer-to-wafer level residuals after time-series filtering (middle); Lot-to-lot level residuals after time-series filtering (bottom). Upper and lower control limits (UCL and LCL) are shown for the residual plots.

Looking at the first two lots of the production sequence in Figure 4-16, it is clear that the large shift in the power influences many of the other RF signals, which are adjusting to the change in load caused by the power shift. This power change explains the alarms in these lots, but does not account for the alarms in the three lots following the first two.

Examination of the individual sensor signals and residuals is helpful in this case. Note that the RF coil signal in Figure 4-17 exhibits a clear drift in behavior following the abrupt shift after the first two lots. The corresponding residuals show that this drift is captured in the lot average residual, but does not affect the wafer average residual profile. This is a clear example of machine drift that is visible more clearly in one time scale than another.

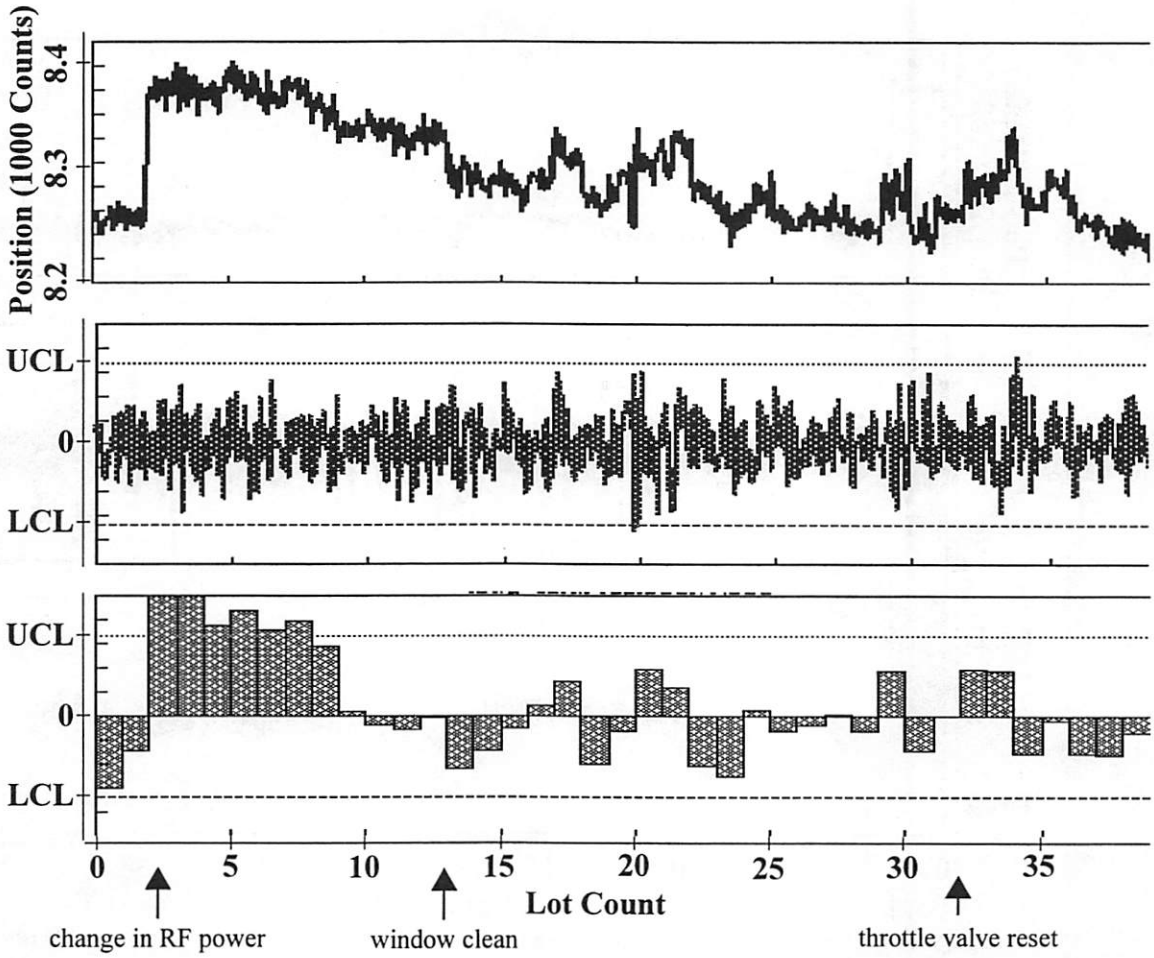


Figure 4-17. Univariate analysis for RF coil position signal for recipe 2 data

Original signal (top) is measured in thousands of counts; Wafer-to-wafer level residuals after time-series filtering (middle); Lot-to-lot level residuals after time-series filtering (bottom). Upper and lower control limits (UCL and LCL) are shown for the residual plots.

Finally, for the lots processed after the throttle valve reset, it appears that again the only sensor signal showing significant change is the pressure, shown in Figure 4-18, and this appears at the lot average level with no significant affect at the wafer to wafer time scale.

Although in both cases the endpoint signal did not appear to be a significant indicator of a failure event, the models and filtering procedure developed in this chapter allow this signal to be incorporated with the other signals in the double T^2 chart. Without this development, as shown in Figure 4-7, it is likely that there would have been false alarms in the portions of the baseline data immediately following chamber or window cleans, or preventative maintenance events.

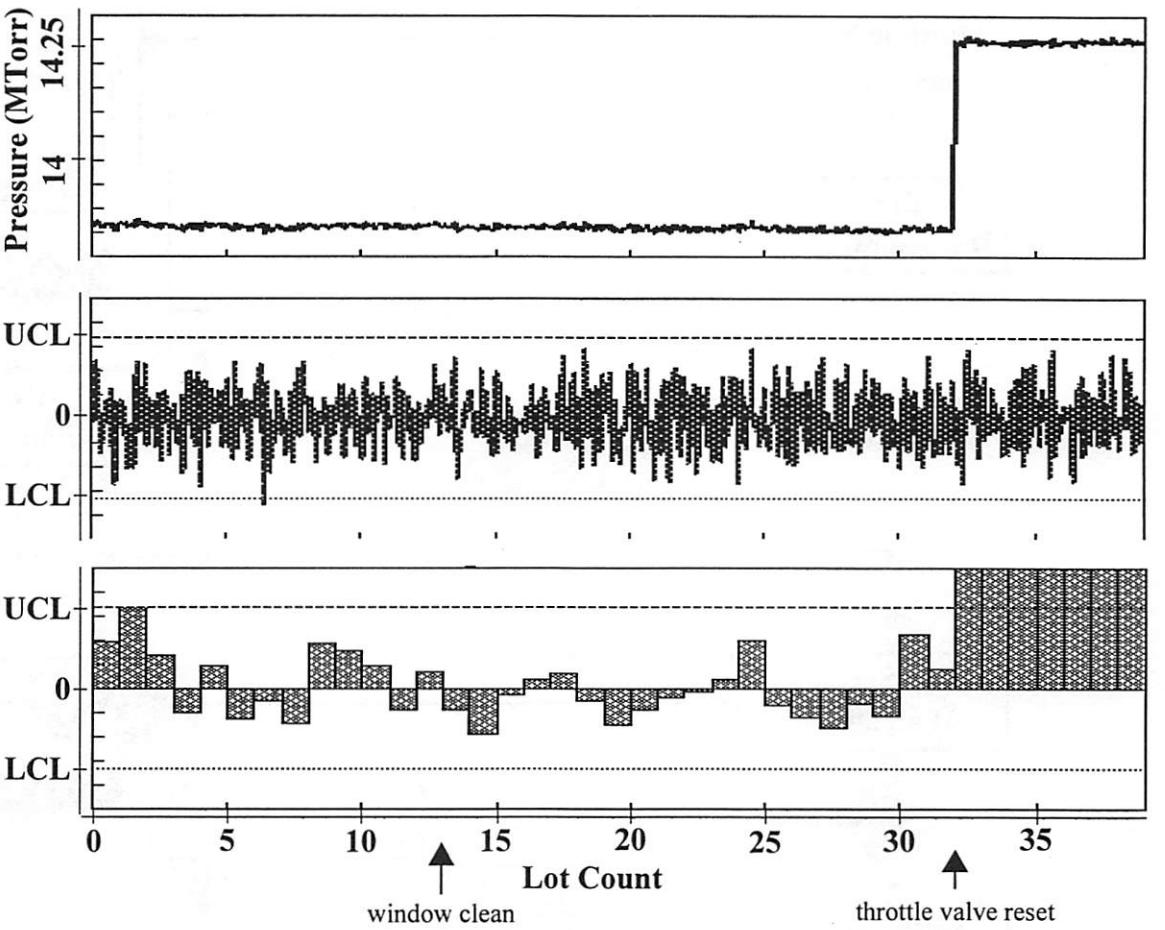


Figure 4-18. Univariate analysis for chamber pressure signal for recipe 2 data

Original signal (top); Wafer-to-wafer level residuals after time-series filtering (middle); Lot-to-lot level residuals after time-series filtering (bottom). Upper and lower control limits (UCL and LCL) are shown for the residual plots.

4.5. Summary

The models developed to account for long term trends are consistent with physical equations describing the window attenuation effect on the measured data. Furthermore, the results are repeatable over several preventative maintenance (PM) cycles, with little variation of the linear regression model from one cycle to the next. This suggests that a simple linear adaptive model may be used to effectively predict the behavior of a cycle, even after a change of the machine state as drastic as that produced by a PM event. This development enhances the current tool by enabling the optical emission signals to be combined with the other sensor data, and makes monitoring robust over time. A further advantage is that new models would not have to be reconstructed each time the chamber or window is cleaned.

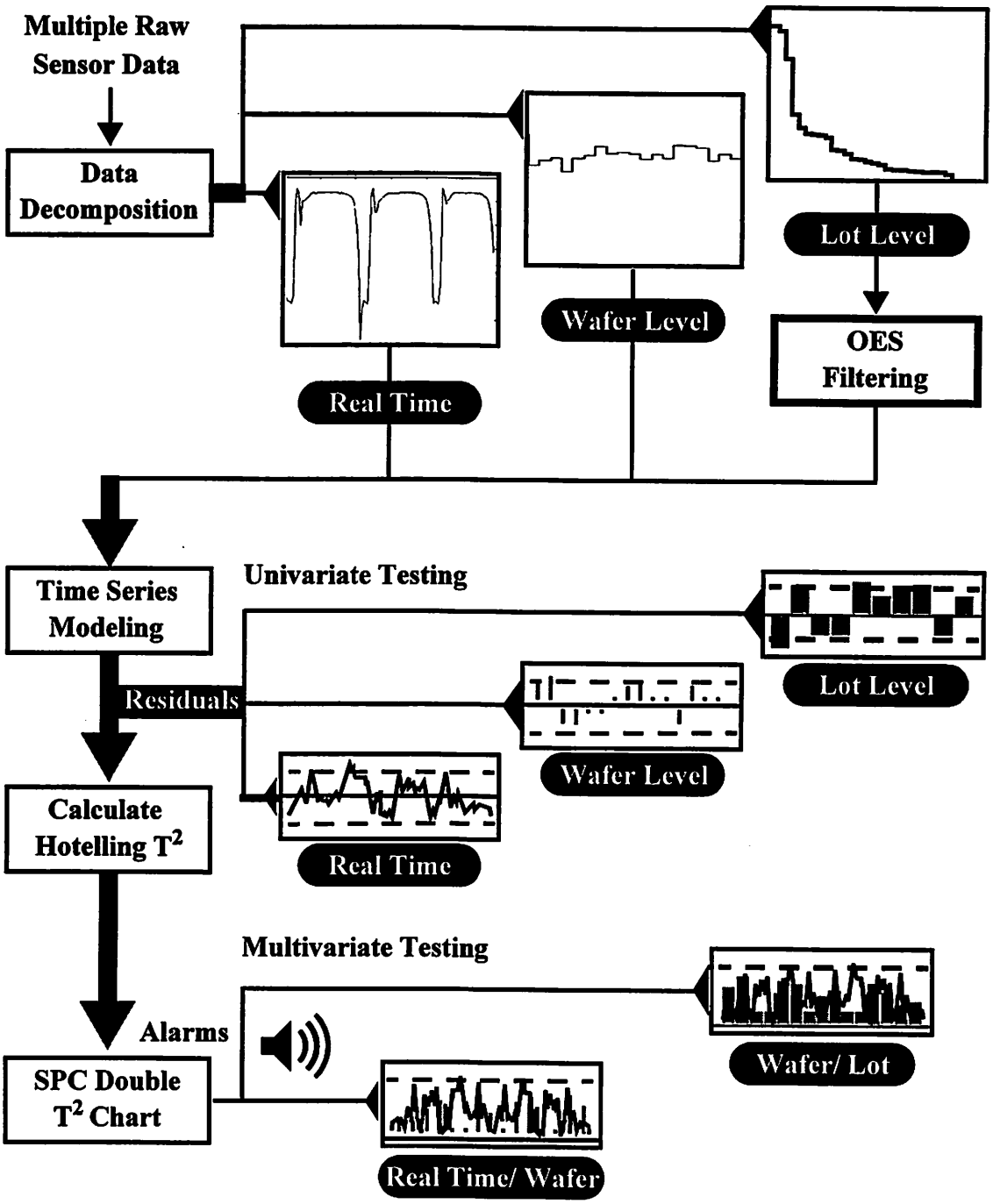


Figure 4-19. Flowchart for improved fault detection and analysis for three time scales

Our improved fault detection mechanism incorporates long term trends and allows analysis to be conducted at different time scales, resulting in a more powerful investigative tool for providing better insight for fault detection and diagnosis. A flowchart summarizing the procedure is shown in Figure 4-19.

5 Methods for Classification and Decision Making

5.1. Introduction

In the previous chapter we developed improvements to a system for fault detection based on monitoring real time tool signals for plasma etch equipment. In particular, it is clear that long term trends can be explained by machine aging and hence, through appropriate filtering, we are able to account for this drift. With the fault detection mechanism finely tuned and capable of integrating information at various time scales, the next task is to diagnose the problems causing the faults detected by the system. This task is complicated by variability, specifically, in the type of data, and its source. The different data types lend themselves to different modeling techniques, and, by exploiting varying levels of resolution and detail, features may be extracted for fault classification. This chapter reviews various methods and approaches for handling uncertainty, focusing on probabilistic models that accommodate the intermingling of techniques to extract information critical for decision making. In particular, we discuss the theoretical basis for construction of a decision support tool to enhance the engineer's ability to make crucial decisions based on timely identification of the machine state.

5.2. Data mining and sensor fusion

The process of extracting knowledge from data is often referred to as data mining. Rule bases, decision trees, and neural networks are among the representations used for data mining, employing techniques such as density estimation, clustering, regression and classification [13].

The general problem of classification of data into categorical groups in order to draw some conclusion or inference has been considered by researchers spanning many different

fields and applications. The term "*expert system*" has been used to describe a structure that combines various types of information for such a purpose. This process has also been referred to as *information* or *data fusion*, and in particular, for data from multiple sensors, as *sensor fusion.*

One key goal in sensor fusion is to reduce uncertainty. A distinction can be made here between uncertainty and imprecision [14]. Sensor uncertainty depends on what is observed rather than the sensor itself. Thus, missing features, an inability of the sensor to measure all relevant attributes, or ambiguous observations can all contribute to uncertainty. In theory, the advantage of multiple sensors is that the observations of each one may be combined into an improved estimate of the state compared to one derived from a single sensor.

Our goal is to build a diagnostic system for machine fault classification combining evidence from multiple sensors. One challenge in multisensor systems is in evaluating how sensors should be implemented, and the role each plays in data management and decision making. Each sensor becomes a potential contributor to a composite decision process. Although the benefits of sensor fusion have motivated much research in the area, a general purpose method for fusion across levels has yet to emerge [15]. The lack of consensus for a single approach can be explained by the various difficulties associated with multiple sensors. For instance, the sensors' outputs may have little in common, offer different resolutions of data, or have minimal or no relation to each other. The problem is further compounded by issues of sensor and measurement noise.

5.3. Methodologies for handling uncertainty

The approaches for handling uncertainty typically fall under one of the following theories: (1) *probability theory*, which includes *Bayesian theory*, (2) *Dempster-Shafer theory*, also known as *evidence theory*, and (3) *fuzzy set theory.*

Given the wide range of fields and applications for classification and sensor fusion, it is not surprising that different methodologies for representing and dealing with uncertainty issues have been developed as a result. We require a framework for combining evidence, whether from different sensors or from other sources, such as human experts, and for gen-

erating diagnoses from the extracted information. The approaches for evidential reasoning and decision making are typically considered to fall into one of three general theories.

5.3.1. Probability Theory

Probability theory, being the oldest and most established of the three, is often the benchmark to which other methods are compared [16]. More recently, developments in graphical modeling approaches based on probability theory have proven highly successful in the area of diagnosis and classification [17], [18], [19], [20]. Probability-based evidential reasoning systems assign probability values to events. Bayesian statistics refer to a set of techniques for inference that combine measured or observed data with subjective beliefs, using Bayes' theorem. With certainty in a feature represented as a probability function, faults can be linked to observations or evidence, and then Bayes' rule may be applied in order to calculate the likelihood of a particular fault. Examples of using a Bayesian approach can be found in [21], [22], [23]. In addition, a well formalized procedure exists for implementation of a diagnostic system based on Bayesian theory [24], [25]. One drawback, however, is that it requires the values of a large number of conditional probabilities. Proponents of other methods have also criticized this approach for its lack of an explicit representation of ignorance. There are a few instances applying non-Bayesian techniques specifically for sensor fusion, using point probabilities with alternate application-dependent decision rules [26], [27], [28].

As an example of how this approach might be implemented, let us look at a plasma etch application. Figure 5-1 is a depiction of the plasma etch process that includes input settings and some relevant sensor measurements.

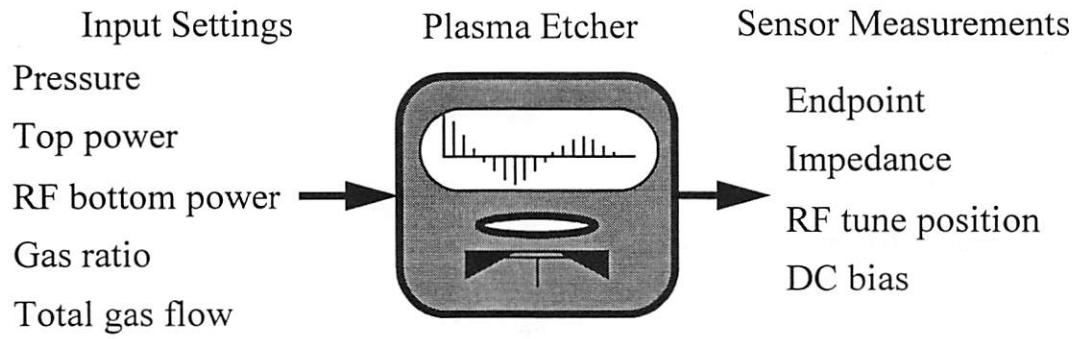


Figure 5-1. The plasma etch process

Suppose we vary the input settings of the plasma etcher, while collecting sensor signals. If we are concerned with identifying shifts in the inputs meant to simulate the occurrence of a fault, we would want to infer this from our observations of monitored sensor signals measured during wafer processing. To define this problem, we need to specify a *fault space* comprised of the different hypotheses, and an *evidence space* of observations.

Fault label	Fault hypothesis
F_1	wrong pressure
F_2	wrong top power
F_3	wrong RF bottom power
F_4	wrong gas ratio
F_5	wrong total gas flow
ζ	no fault

Evidence label	Observation
E_1	endpoint signal
E_2	impedance signal
E_3	RF load position
E_4	DC bias signal

Table 5-1. Fault and evidence spaces for the plasma etch process

The Bayesian approach employs Bayes' rule to calculate the *posterior probability* of a fault given the evidence.

$$P(F_i/E_j) = \frac{P(E_j/F_i) \cdot P(F_i)}{P(E_j)}, i = \{1, 2, \ldots, 5\}, j = \{1, 2, \ldots, 4\} \qquad (5.1)$$

Here the term $P(E_j/F_i)$ is the *conditional probability* of the evidence given the fault, also known as the class conditional or posterior probability, $P(F_i)$ is the prior probability of the fault, and $P(E_j)$ is a normalization term, which can be expressed in terms of priors and posteriors by the following equation:

$$P(E_j) = \sum_{i=1}^{5} P(E_j/F_i) \cdot P(F_i) \qquad (5.2)$$

Table 5-2 lists values of the different posterior probabilities for this example using the endpoint signal, E_1, as evidence.

Thus, for this example, substituting in Equation 5.2, we obtain

Prior	Value	Conditional Probability or Posterior	Value
$P(F_1)$	0.05	$P(E_1/F_1)$	0.13
$P(F_2)$	0.2	$P(E_1/F_2)$	0.76
$P(F_3)$	0.01	$P(E_1/F_3)$	0.01
$P(F_4)$	0.03	$P(E_1/F_4)$	0.03
$P(F_5)$	0.06	$P(E_1/F_5)$	0.02
$P(\zeta)$	0.65	$P(E_1/\zeta)$	0.05

Table 5-2. Prior and likelihood probabilities for fault categories and endpoint evidence

$$P(E_1) = (0.05 \cdot 0.13) + (0.2 \cdot 0.76) + (0.01 \cdot 0.01) + (0.03 \cdot 0.03) + (0.06 \cdot 0.02) + (0.65 \cdot 0.05) \quad (5.3)$$

which gives $P(E_1) = 0.1932$. Consequently, the posterior probabilities of each fault given the evidence observation of the endpoint signal are calculated using Equation 5.1, and the results are summarized in Table 5-3. In this case, we would conclude that given the observed endpoint signal evidence, the most likely fault cause is using the wrong top power setting (F_2).

Posterior	Value
$P(F_1/E_1)$	0.0336
$P(F_2/E_1)$	0.7867
$P(F_3/E_1)$	0.0005
$P(F_4/E_1)$	0.0047
$P(F_5/E_1)$	0.0062
$P(\zeta/E_1)$	0.1682

Table 5-3. Posterior probabilities of fault hypotheses given endpoint evidence

5.3.2. Dempster-Shafer Theory

Another popular approach was first proposed as an alternative to Bayesian probability by Shafer, and subsequently built upon by Dempster [29], [30]. The Dempster-Shafer (DS) theory, also commonly referred to as *evidence theory*, defines a finite set of mutually exclusive propositions on a domain called the *frame of discernment* (Θ). Evidence is repre-

sented as a Shafer *belief function* over <0.0, 1.0>. This interval is used for convenience, giving the appearance of a probability. Belief functions interpret evidence of some observation, and serve as a model for transferring belief, but they cannot be interpreted as probabilities of events [15].

One distinction in this theory lies in the concept of attaching portions of probabilistic measure to higher levels of abstraction than the focal elements of the problem. These abstractions are unions of the focal elements in Θ. Thus, for n focal elements, the set of all possible subsets of theta is the power set 2^n. In DS theory, two measures of uncertainty are computed for each element. *Supportability* (S) is defined as the degree of belief directly supporting a specific element of the power set. In contrast, *plausibility* (P) is the degree of belief not directly in contradiction of a specific element. With these definitions, it is possible to explicitly represent ignorance as the difference between plausibility and supportability of an event.[3] Dempster's rule of combination serves as the mechanism for combining independent sources of information.

Returning to our example of a plasma etch diagnosis application, we can consider the frame of discernment, Θ, as the fault space. Note that with our fault set of 5 categories, we now have $2^5 = 32$ possible subsets in our fault space. We also assume that we have a multivalued mapping function, Γ, that maps the elements in the evidence space, E, to the fault space, Θ, and that elements may be mapped to an individual hypothesis, or any subset of hypotheses. These evidence mappings can be specified by defining a *basic probability mass* distribution or BPMD [31]. A set of basic probability masses (BPM) are used to distribute belief from an evidence element to a set of hypotheses in the fault space. Any unassigned belief will be assigned to the entire set, Θ.

Using the BPMD, we can extract intervals, $[S(X), P(X)]$, for an individual hypothesis. The support and plausibility of a hypothesis X are specified by:

$$S(X) = \sum m_i(X_i) \qquad (5.4)$$

[3] In contrast, classic probability theory assigns wide confidence intervals to estimated probability values.

$$P(X) = 1 - \sum m_i(\neg X_i) \quad (5.5)$$

where $X_i \subseteq X$, $\Theta = X \cup \neg X$ and $\neg X_i \subseteq \neg X$. Thus, Equations 5.4 and 5.5 say that the total support in X is given by the sum of supports assigned to X and all subsets of X.

For our example, let us take as evidence observations, the monitored sensor signals corresponding to the endpoint (E_1) and the impedance (E_2) of the plasma, and specify the BPMD's, m_1 and m_2, as the masses derived from the multivalued mappings, $\Gamma_1: E_1 \rightarrow \Theta$ and $\Gamma_2: E_2 \rightarrow \Theta$. Using Dempster's rule of combination [30], we can calculate the combined BPMD as follows:

$$m(Z) = \frac{\sum m_1(X_i)m_2(Y_j)}{1-k}, \quad X_i \cap Y_j = Z \quad (5.6)$$

$$k = \sum m_1(X_i)m_2(Y_j), \text{ if } X_i \cap Y_j = \varnothing \quad (5.7)$$

Equations 5.6 and 5.7 define the BPM of the intersection of X_i and Y_j as the product of the BPM's of X_i and Y_j, with a normalization factor of (1-k) to account for the belief which would have been assigned to the empty set.

Using the following evidence mappings for $\Gamma_1: E_1 \rightarrow \Theta$ and $\Gamma_2: E_2 \rightarrow \Theta$:

$$m_1(F_1, F_2 \cup F_5, F_3, F_4, \zeta, \Theta) = (0.05, 0.8, 0, 0, 0.1, 0.05) \quad (5.8)$$

$$m_2(F_2, F_1 \cup F_3, F_4 \cup F_5, \zeta, \Theta) = (0.25, 0.4, 0, 0.15, 0.2) \quad (5.9)$$

we can calculate the combination of m_1 and m_2. Table 5-4 lists the propositions (subsets of the fault space) from m_1 along the first column, while those of m_2 are given along the top row. Thus, the cells of the table show the intersection of the corresponding propositions associated with m_1 and m_2. Note that the intersection with the whole set, Θ, simply returns the original proposition. Assuming independent evidence sources, the values of the intersections of the propositions are given by the product of the values of the propositions, and these are summarized in Table 5-5.

Equation 5.6 gives us the following for the combination of m_1 and m_2:

$m_1 \setminus m_2$	F_2	$F_1 \cup F_3$	$F_4 \cup F_5$	ζ	Θ
F_1	$\varnothing$	F_1	$\varnothing$	$\varnothing$	F_1
$F_2 \cup F_5$	F_2	$\varnothing$	F_5	$\varnothing$	$F_2 \cup F_5$
F_3	$\varnothing$	F_3	$\varnothing$	$\varnothing$	F_3
F_4	$\varnothing$	$\varnothing$	F_4	$\varnothing$	F_4
ζ	$\varnothing$	$\varnothing$	$\varnothing$	ζ	ζ
Θ	F_2	$F_1 \cup F_3$	$F_4 \cup F_5$	ζ	Θ

Table 5-4. Sets formed from the intersection of propositions associated with m_1 and m_2

$m_1 \setminus m_2$	0.25	0.4	0	0.15	0.2
0.05	0.0125	0.02	0	0.0075	0.01
0.8	0.2	0.32	0	0.12	0.16
0	0	0	0	0	0
0	0	0	0	0	0
0.1	0.025	0.04	0	0.015	0.02
0.05	0.0125	0.02	0	0.0075	0.01

Table 5-5. Corresponding belief mass values for the sets formed from the intersection operation

$$m(F_1, F_2, F_3, F_1 \cup F_3, F_4, F_5, F_4 \cup F_5, \zeta, F_2 \cup F_5)$$

$$= (0.0632, 0.4474, 0, 0.0421, 0, 0, 0, 0.0895, 0.3368) \quad \textbf{(5.10)}$$

Using Equation 5.5, the corresponding probability intervals for each fault hypothesis are:

$F_1[0.0632, 0.1263]$, $F_2[0.4474, 0.8052]$, $F_3[0, 0.0631]$, $F_4[0, 0.021]$, $F_5[0, 0.3578]$, and $\zeta[0.0895, 0.1105]$. Again, these results show that the most likely fault cause is due to using the wrong top power setting, F_2.

Bayesian theory and DS theory have both been used successfully in a number of sensor fusion applications. Although the majority of these systems represent sensor evidence probabilistically and use Bayes' rule for inference [21], [22], [23], [24], [25], a significant portion rely on the DS framework and consider sensor evidence in terms of belief [15], [22], [32], [33], [34], [35], [36]. An application for monitoring, maintenance and diagnosis

for a low pressure chemical vapor deposition (LPCVD) process using the DS approach can be found in [31].

There is much in the literature discussing the advantages and drawbacks of each theory [37], [38], [39], [40]. In particular, some researchers contend that evidence theory is either more powerful or that it can address some problems that probability theory cannot. Others view evidence theory as having limitations, claiming that Bayesian theory is a more effective and efficient method. A direct comparison can be found in [41], which applies Bayesian and evidential reasoning to the same target identification problem requiring multiple levels of abstraction. The two reasoning methods are compared in terms of convergence for a number of aircraft identification scenarios including missing reports and misassociated reports. These results show that probability theory can accommodate all issues dealing with uncertainty and converge to a solution faster than evidence theory.

5.3.3. Fuzzy Set Theory

The third approach is motivated by the claim that probability and statistics do not adequately deal with certain kinds of uncertainty. Fuzzy set theory (FST), first advocated in 1965 by L.A. Zadeh, has mainly been established in applications of control theory and artificial intelligence [42], but has more recently been applied as an alternative to traditional statistical methods in areas such as statistical quality control, linear regression, forecasting and reliability [43]. One claim is that this theory serves as a bridge of communication between man and machine. For example, in applications involving diagnosis, inference, systems and control, observations are often expressed in linguistic terms or human expert opinions. These data suffer from uncertainty and ambiguity due to subjective judgment and interpretation, as opposed to the measurement noise, imprecision, or natural process variation that contribute to randomness in the statistical sense. While statistical variation is based on the distribution of data, in cases where the occurrence of an event is unclear, or the total data has no meaning, there is no distribution. The observation that "the tomato is red" or a statement that "the tomato is almost ripe" are not readily handled by probability theory. Thus, the most salient aspect of FST lies in its ability to represent the gradation of boundaries of states, relationships, constraints and goals, where the range or interpretation of the definition is vague.

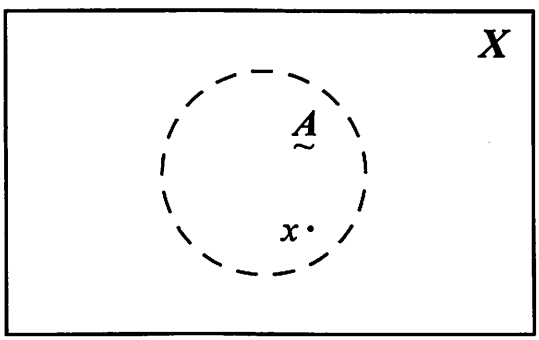
In the case of diagnosis and inference, through FST techniques, one would hope to express human experience in a form easily evaluated by a machine, and moreover, to convert the output of the machine into a form people can understand. Some systems are even designed to imitate human judgment and understanding. However, because a model derives its value from being a concise expression capturing the essence of a real problem, the first issue to consider is what part of the system (if any) would be better represented by FST, and what form this conversion will take.

A system model is composed of many types of variables, dependent and independent, state, input, output and decision nodes, and must incorporate transitions, cause and effect. The procedure for building a fuzzy model takes place in two stages: (1) definition of the sets of variables and logical relationships and (2) conversion to fuzzy sets and fuzzy relationships. To construct a fuzzy set, one must first identify a membership function that assigns a grade of membership between zero and one to each element in a set. Mathematically, the membership function is a mapping from the space of elements to the unit interval, again, giving an appearance of a probability. Table 5-6 summarizes a few notations used in defining fuzzy sets; a formal definition is as follows [44].

The function $\mu : X \rightarrow [0, 1]$ is given the label $\underset{\sim}{A}$ and $\underset{\sim}{A}$ is called a fuzzy (sub)set of X. μ is called the membership function of $\underset{\sim}{A}$, and defines the extent of membership of element x into $\underset{\sim}{A}$. **D (5.1)**

$x \in X$

$0 \leq \mu_{\underset{\sim}{A}}(x) \leq 1$



$x \in \underset{\sim}{A}$
with a fuzzy membership
$\mu(x) \rightarrow [0, 1]$

Figure 5-2. Fuzzy Subset $\underset{\sim}{A}$

Notation	Meaning
X	whole set
A	subset of X
$\underset{\sim}{A}$	fuzzy subset of X
$\varnothing$	empty set
{0,1}	the set of zero and one
[0,1]	the real-number interval from zero to one
χ_A	the characteristic function of set A in X
$\mu_{\underset{\sim}{A}}$	the membership function of set $\underset{\sim}{A}$ in X
$a \wedge b$	the min of a and b
$a \vee b$	the max of a and b

Table 5-6. Notation for fuzzy sets [44]

Figure 5-2 depicts a fuzzy subset $\underset{\sim}{A}$ [44]. The rectangle represents the whole set X, and the dashed circle, the ambiguous boundary of the fuzzy subset $\underset{\sim}{A}$. A member of the set is the element x, whose membership function μ gives the degree or extent to which x is a member of $\underset{\sim}{A}$.

It is important to note the difference between a fuzzy set and a standard set, also referred to as a "crisp" set. Returning to the data from our plasma etch example, let us take the fault hypothesis, F_2, the label corresponding to using the wrong top power setting on the etcher, to illustrate this difference. Our data for the top power setting consist of several values. We need to specify which of these are "wrong" and which are correct. Moreover, we wish to make a further distinction between a setting that is "too high" versus one that is "too low". In order to implement this quantitatively, we require a method of determining appropriate threshold values. In effect, we are specifying the boundaries between the correct setting, and values that are either too high or too low.

Nonfuzzy sets have been called crisp sets, due to their clearly defined boundaries. Characteristic functions may also be used to define the membership of an element to a crisp set. In particular, if C is a crisp subset of X, the characteristic function of C is given by:

$$\chi_C(x) = \begin{cases} 1; & x \in C \\ 0; & x \notin C \end{cases} \qquad (5.11)$$

This is equivalent to a membership function of C with a grade that is two-valued. In other words, the element x either belongs to C with a grade of one, or it does not belong, and its grade is zero.

If we think of the decision rendered by the characteristic function of a crisp set as making a determination between black and white, then the membership functions for fuzzy sets assign grades of membership by distinguishing among shades of grey. In other words, membership functions are an extension of characteristic functions in that they allow for a membership grade within the range [0,1], as opposed to {0,1} for crisp sets. One consequence of this is that a given element of X may simultaneously hold non-zero grade values of membership in multiple sets. That is, the boundaries between sets are vague or fuzzy. Figures 5-3 and 5-4 illustrate the difference between crisp and fuzzy sets.

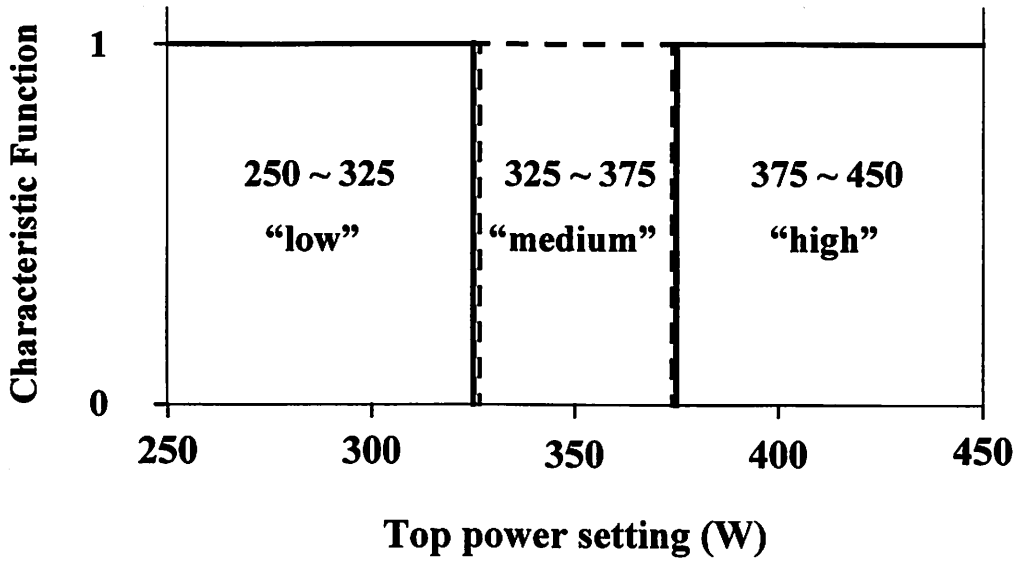


Figure 5-3. Characteristic functions of crisp sets "low", "medium" and "high" top power

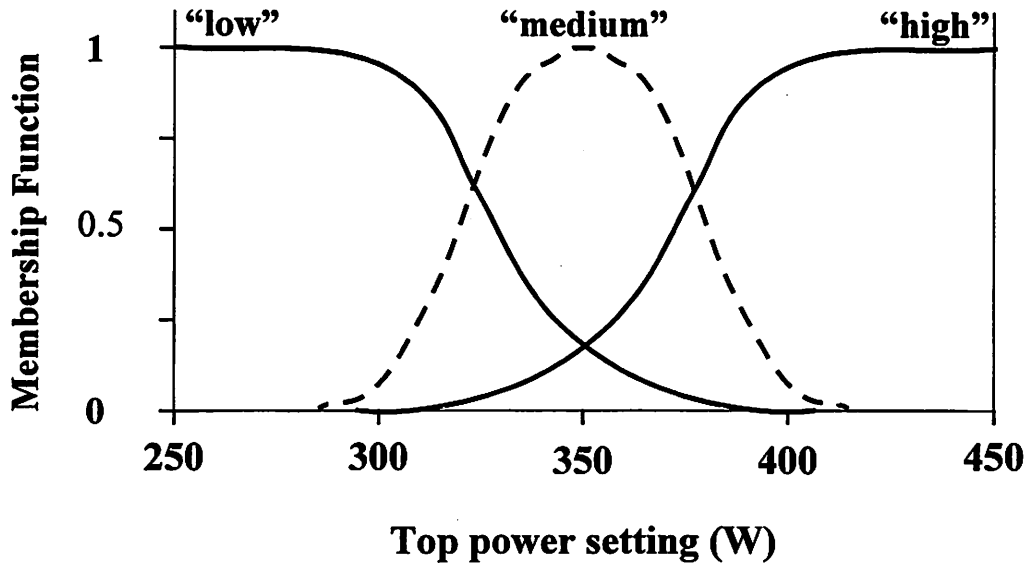


Figure 5-4. Membership functions of fuzzy sets "low", "medium" and "high" top power

The basic operations conducted on crisp sets are used to give unions, intersections and complements of sets. However, because fuzzy sets are defined by membership functions, operations conducted on fuzzy sets must utilize membership functions. The following definitions are necessary for this purpose:

Union of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$:

$$\mu_{\underset{\sim}{A}\cup\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \vee \mu_{\underset{\sim}{B}}(x) \qquad \textbf{D (5.2)}$$

Intersection of fuzzy sets $\underset{\sim}{A}$ and $\underset{\sim}{B}$:

$$\mu_{\underset{\sim}{A}\cap\underset{\sim}{B}}(x) = \mu_{\underset{\sim}{A}}(x) \wedge \mu_{\underset{\sim}{B}}(x) \qquad \textbf{D (5.3)}$$

Complement of fuzzy set $\underset{\sim}{A}$:

$$\mu_{\overline{\underset{\sim}{A}}}(x) = 1-\mu_{\underset{\sim}{A}}(x) \qquad \textbf{D (5.4)}$$

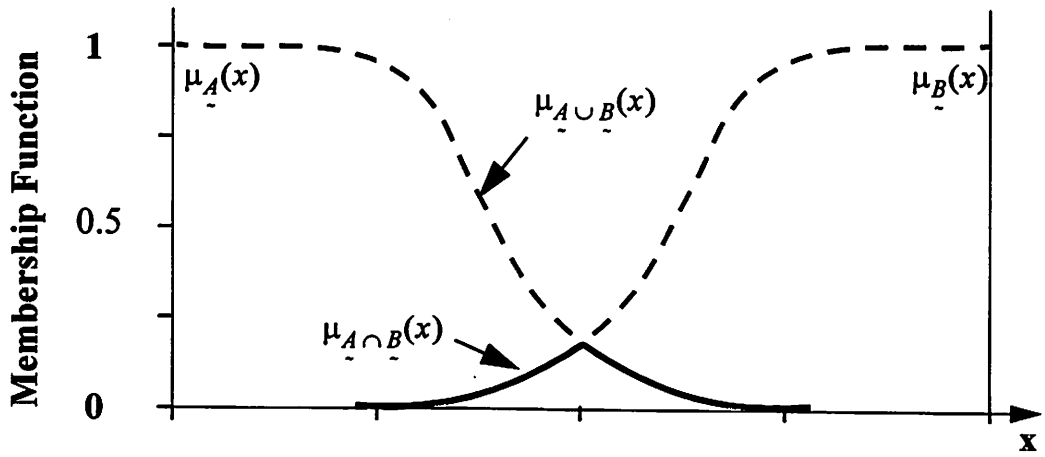


Figure 5-5. Union and intersection of fuzzy sets A and B

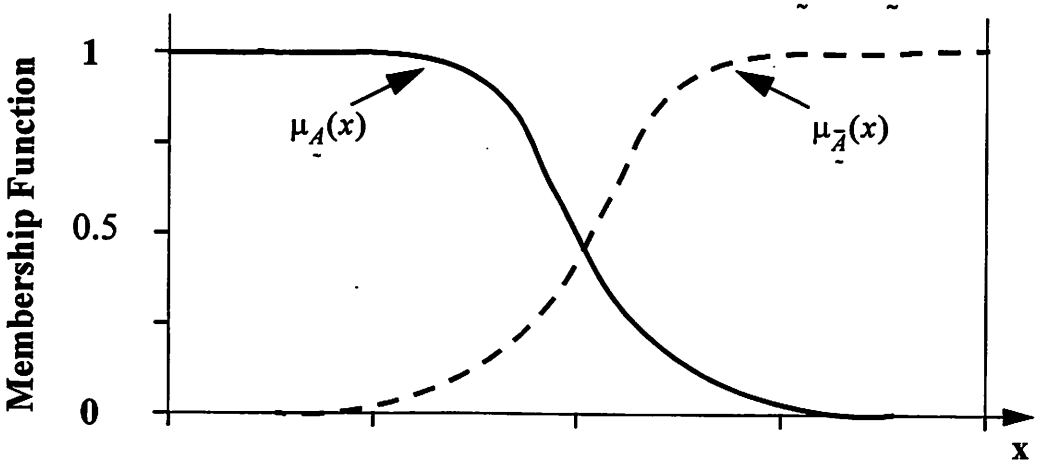


Figure 5-6. Complement of a fuzzy set

Figures 5-5 and 5-6 show graphs of these operations using fuzzy sets A and B, which could for instance, correspond to "low top power" and "high top power" respectively, as in our previous example.

In particular, we can see that these definitions are extensions of crisp sets. If we take the characteristic functions of two crisp sets, ***C*** and ***D***, we can define the union, intersection and complement as follows:

$$\chi_{C \cup D}(x) = \chi_C(x) \vee \chi_D(x) \qquad \textbf{D (5.5)}$$

$$\chi_{C \cap D}(x) = \chi_C(x) \wedge \chi_D(x) \qquad \textbf{D (5.6)}$$

$$\chi_{\bar{C}}(x) = 1 - \chi_C(x) \qquad \text{D (5.7)}$$

These are depicted in Figures 5-7 and 5-8.

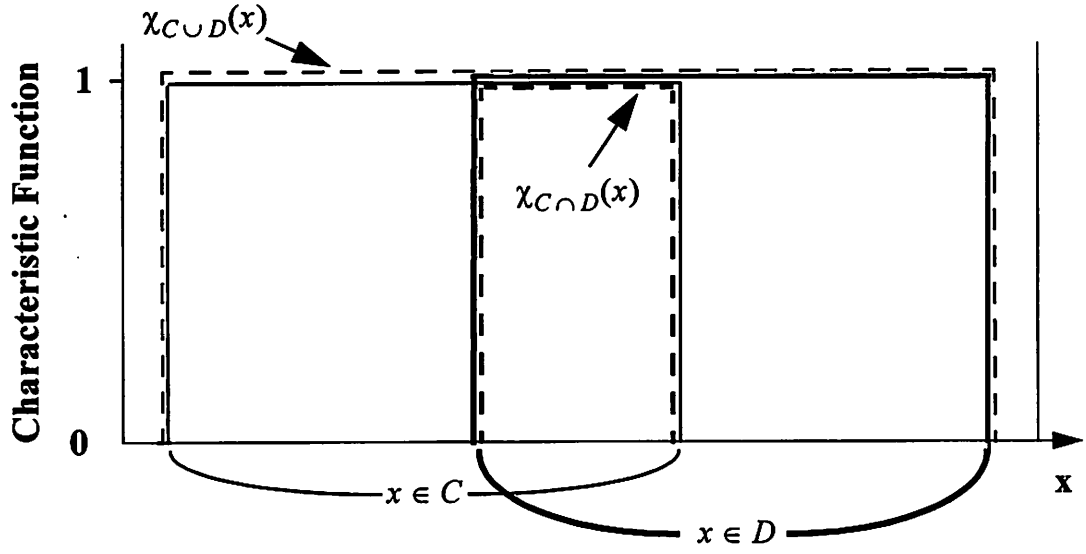


Figure 5-7. Union and intersection of crisp sets ***C*** and ***D***

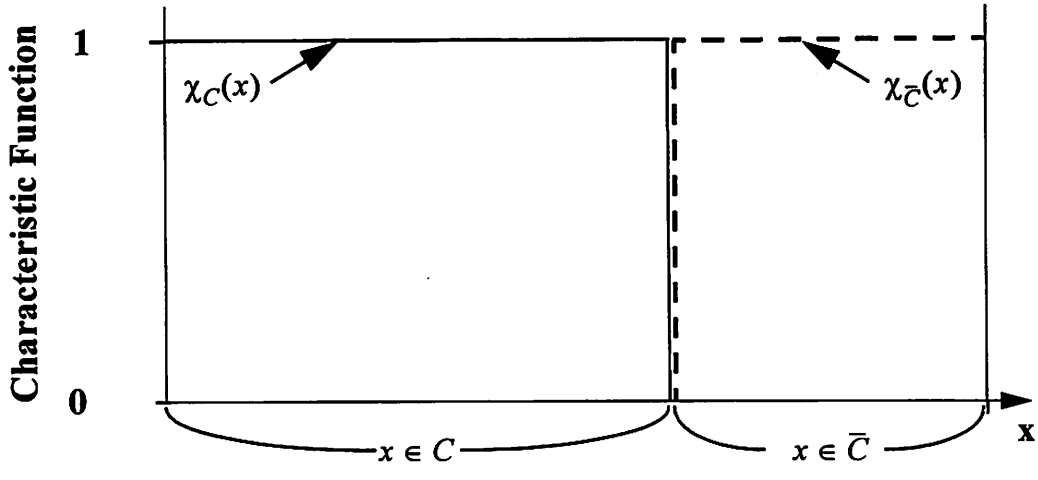


Figure 5-8. Complement of a crisp set

From the figures, it is clear that fuzzy complements do not necessarily share the characteristics of crisp complements. One important difference is that fuzzy sets do not follow

the excluded-middle law, nor do they follow the law of contradiction of crisp sets. More specifically, we have that for a fuzzy set $\underset{\sim}{A}$:

$$\underset{\sim}{A} \cup \underset{\sim}{\bar{A}} \neq X \tag{5.12}$$

$$\underset{\sim}{A} \cap \underset{\sim}{\bar{A}} \neq \varnothing \tag{5.13}$$

Moreover, the fuzzy membership functions for the whole and empty sets for all values of x are given by:

$$\mu_X(x) = 1 \tag{5.14}$$

$$\mu_{\varnothing}(x) = 0 \tag{5.15}$$

In contrast, crisp sets result in the following laws, necessary for two-valued logic [44]:

$$C \cup \bar{C} = X \quad \text{(excluded-middle law)} \tag{5.16}$$

$$C \cap \bar{C} = \varnothing \quad \text{(law of contradiction)} \tag{5.17}$$

For finite sets, given by $X = \{x_1, x_2, \ldots, x_n\}$, membership functions can be expressed in the following manner [44]:

$$\underset{\sim}{A} = \sum_{i=1}^{n} \mu_A(x_i)/x_i = \mu_A x_1/x_1 + \mu_A x_2/x_2 + \ldots + \mu_A(x_n)/x_n \tag{5.18}$$

Note that the elements of the set are written on the right side of the slash, and the corresponding grades of membership on the left. This notation allows us to represent operations. For instance, using a "+" to represent "or" results in an operation that assigns the maximum grade when the elements are the same:

$$a/x_1 + b/x_1 = a \vee b/x_1 \tag{5.19}$$

Finally, we need a few more concepts to demonstrate how FST can be applied to a classification problem. In order to define fuzzy relations, fuzzy reasoning, and fuzzy logic, we require some basic building blocks. A fuzzy proposition is an expression that makes a statement. A typical example might be "x is $\boldsymbol{A}$", where x is an element of the set, and $\boldsymbol{A}$ is a fuzzy predicate, or fuzzy variable. In the plasma etch example, x might be a sensor mea-

surement such as the endpoint intensity, and $\boldsymbol{A}$ might represent the set "high". The proposition "x is $\boldsymbol{A}$" would then be interpreted as the statement "the endpoint intensity is high". In addition, fuzzy propositions can make use of modifiers to change the predicate; this is represented as "x is $\mathbf{m}\boldsymbol{A}$", where the modifier $\mathbf{m}$ can be, for example, "very" or "not", resulting in a modified statement, "the endpoint intensity is very high" or "the endpoint intensity is not high". Propositions can be combined to produce composite propositions such as:

"x is $\boldsymbol{A}$" or "x is $\boldsymbol{B}$" = "x is $\boldsymbol{A} \cup \boldsymbol{B}$"

"x is $\boldsymbol{A}$" and "x is $\boldsymbol{B}$" = "x is $\boldsymbol{A} \cap \boldsymbol{B}$"

An implication is a combination formed using an "if" statement:

"if x is $\boldsymbol{A}$ then y is $\boldsymbol{B}$" = "(x, y) is $\boldsymbol{A} \rightarrow \boldsymbol{B}$"

where $\boldsymbol{A} \rightarrow \boldsymbol{B}$ is the fuzzy subset $X \times Y$, with a membership function given by:

$$\mu_{\boldsymbol{A} \rightarrow \boldsymbol{B}}(x, y) = (1 - \mu_{\boldsymbol{A}}(x) + \mu_{\boldsymbol{B}}(y)) \wedge 1 \qquad (5.20)$$

There are various implication formulae used in fuzzy reasoning; the interested reader is referred to [44] for a discussion of these formulae and their applications.

Returning to our plasma etch example, let us take an evidence observation, the monitored sensor signal corresponding to the endpoint intensity ($\boldsymbol{E_1}$) of the plasma, and represent this using the fuzzy sets $\boldsymbol{E_{1L}}$ for low and $\boldsymbol{E_{1H}}$ for high endpoint intensity, respectively. Moreover, let us take the fault hypothesis corresponding to using the wrong top power setting ($\boldsymbol{F_2}$) and model this using fuzzy sets. We can consider the membership functions in Figure 5-4 as specifying the fuzzy sets corresponding to $\boldsymbol{F_{2L}}$ = "low", $\boldsymbol{F_{2M}}$ = "medium", and $\boldsymbol{F_{2H}}$ = "high" top power. Figure 5-9 illustrates the membership functions for $\boldsymbol{E_{1L}}$ and $\boldsymbol{E_{1H}}$, representing low and high endpoint intensities. Although in both cases we have specified distinct membership functions to distinguish between "low" and "high" for the endpoint and top power respectively, note that alternatively, we could have employed the

complement operation. In other words, we would specify "low" and "not low" as opposed to defining a separate membership function for "high".

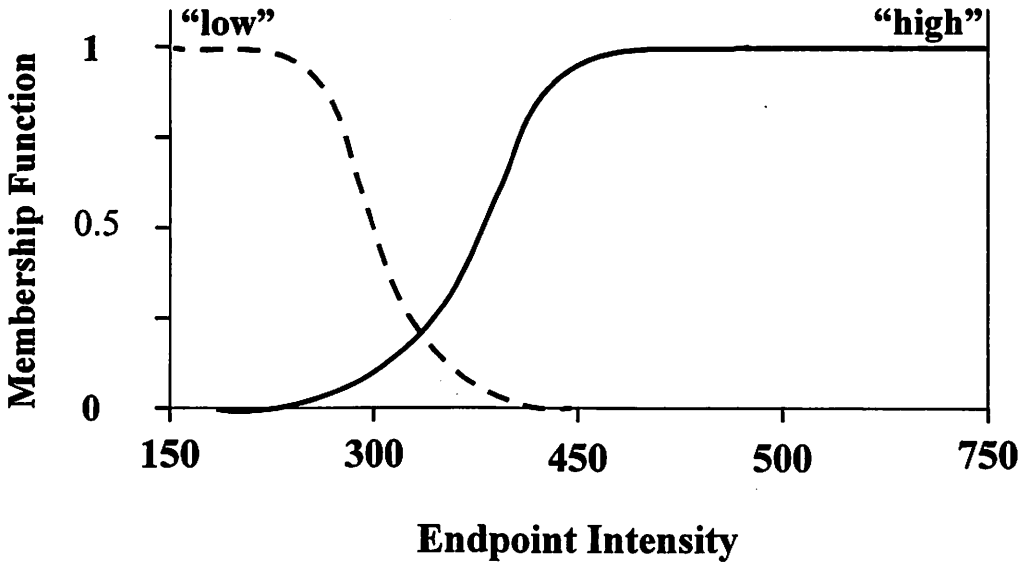


Figure 5-9. Membership functions of fuzzy sets "low" and "high" endpoint intensity

Using Equation 5.18, suppose our endpoint intensity values are given by the finite set $X = \{319.0, 173.4, 197.5, 408.7, 373.4, 534.4, 648.9, 288.3, 424.7, 177.4\}$. The corresponding membership functions for E_{1H} and E_{1L} are:

$$E_{1L} = 0.3/319.0 + 1/173.4 + 1/197.5 + 0.05/408.7 + 0.1/373.4 + 0/534.4 \quad (5.21)$$
$$+0/648.9 + 0.85/288.3 + 0.01/427.4 + 0.98/177.4$$

$$E_{1H} = 0.15/319.0 + 0/173.4 + 0/197.5 + 0.8/408.7 + 0.4/373.4 + 1/534.4 \quad (5.22)$$
$$+1/648.9 + 0.1/288.3 + 0.85/427.4 + 0/177.4$$

The finite set of top power settings associated with the measurements of the endpoint intensity above are $Y = \{350, 250, 275, 450, 350, 450, 450, 307.3, 350, 250\}$, which gives the following membership functions for top power:

$$F_{2L} = 0.2/350 + 1/250 + 0.98/275 + 0/450 + 0.2/350 + 0/450 \quad (5.23)$$
$$+0/450 + 0.9/307.3 + 0.2/350 + 1/250$$

$$F_{2M} = 1/350 + 0/250 + 0/275 + 0/450 + 1/350 + 0/450 \quad (5.24)$$
$$+0/450 + 0.18/307.3 + 1/350 + 0/250$$

$$F_{2H} = 0.2/350 + 0/250 + 0/275 + 1/450 + 0.2/350 + 1/450 \quad (5.25)$$
$$+1/450 + 0.01/307.3 + 0.2/350 + 0/250$$

Furthermore, suppose we have the following rules, based on experience:

"If the measured endpoint intensity is low, then the top power setting is low,"

"(x, y) is $E_{1L} \rightarrow F_{2L}$"

"If the measured endpoint intensity is high, then the top power setting is high,"

"(x, y) is $_{1H} \rightarrow F_{2H}$"

Using Equation 5.20, we can calculate the membership functions for $E_{1L} \rightarrow F_{2L}$ and for $_{1H} \rightarrow F_{2H}$. But now suppose we want to find the membership function for the medium values of endpoint intensity, corresponding to medium values (correct settings) of top power. We can use the operation for taking the intersection of "low" and "high" endpoint intensities, $\mu_{E_{1L} \cap E_{1H}}(x) = \mu_{E_{1L}}(x) \wedge \mu_{E_{1H}}(x)$, resulting in the following rule:

"If the endpoint intensity is low and high (medium), then the top power setting is medium,"

"(x, y) is $E_{1L} \cap E_{1H} \rightarrow F_{2M}$"

In other words, by taking the intersection of the sets "low" and "high", we are obtaining the higher values in the low set, and the lower values in the high set. Thus, these correspond to the medium values for endpoint intensity. Table 5-7 lists the measurement data and corresponding membership functions for the endpoint intensity and top power setting respectively, while Table 5-8 summarizes the results of calculations involving membership functions discussed in this example. Note that the diagnosis results represented by the membership function for the implication rules are all correct with the exception of the ninth

sample, given by "(x, y) is $_{1H} \rightarrow F_{2H}$". The rule "if the measured endpoint intensity is high, then the top power setting is high," fails in this case with a datapoint of (x, y) = (427.4, 350), which is clearly an exception to the rule. Of course, this is a simplified example used for demonstration purposes. A full classification system would employ several rule sets based on various sensor signal evidence in order to diagnose the fault hypothesis.

Index	Endpoint	$\mu_{E_{1L}}(x)$	$\mu_{E_{1H}}(x)$	Top Power	$\mu_{F_{2L}}$	$\mu_{F_{2M}}$	$\mu_{F_{2H}}$
1	319.0	0.3	0.15	350	0.2	1	0.2
2	173.4	1	0	250	1	0	0
3	197.5	1	0	275	0.98	0	0
4	408.7	0.05	0.8	450	0	0	1
5	373.4	0.1	0.4	350	0.2	1	0.2
6	534.4	0	1	450	0	0	1
7	648.9	0	1	450	0	0	1
8	288.3	0.85	0.1	307.3	0.9	0.18	0.01
9	427.4	0.01	0.85	350	0.2	1	0.2
10	177.4	0.98	0	250	1	0	0

Table 5-7. Data and membership functions for endpoint (X) and top power (Y)

Index	$\mu_{E_{1L} \rightarrow F_{2L}}$	$\mu_{E_{1H} \rightarrow F_{2H}}$	$\mu_{E_{1L} \cap E_{1H}}(x)$	$\mu_{E_{1L} \cap E_{1H} \rightarrow F_{2M}}(x)$
1	0.9	1	0.15	1
2	1	1	0	1
3	0.98	1	0	1
4	0.95	1	0.05	0.95
5	1	0.8	0.1	1
6	1	1	0	1
7	1	1	0	1
8	1	0.91	0.1	1
9	1	0.35	0.01	1
10	1	1	0	1

Table 5-8. Membership function results implementing implication rules for diagnosis

Given our discussion, one might reasonably conclude that the value added by the FST framework depends on the application and modeling goals. Specifically, in cases where the data is qualitative, subjective, or relies on a linguistic description subject to a range of interpretation, FST provides a structure to capture ambiguity and allow for manipulation in a form that can processed by a machine. However, if the data can be represented and interpreted probabilistically, based on statistical properties, the value of what is gained by employing the FST approach itself becomes ambiguous.

One final point to note is that memberships do not follow the laws of probability. In fact, one of the biggest differences is the idea of a continuum of membership - that an element can simultaneously hold nonzero degrees of membership in sets considered mutually exclusive. Thus, while FST violates the law of the excluded middle, some claim that this enables toleration of vagueness in data, especially for categorical or qualitative data. However, critics argue that there is ambiguity in the interpretation and definition of fuzzy quantifiers, and that fuzzy logic implementations are difficult to adapt to new sensing configurations [15]. A comparison is made in [43] between fuzzy methods and simpler alternatives based on traditional probability and statistical techniques. In this review, using examples applied to control theory and statistical quality control, the authors find no instances of FST being uniquely useful. In other words, no solutions using FST were found that could not be achieved at least as effectively using probability and statistics. Still, there are several examples and applications of possibilistic or fuzzy systems for sensor fusion [22], [45], [46]. Of course, the debate continues, and interested readers are referred to [43] for further discussion by proponents of both sides.

Considering that DS theory requires a fault space even larger than for Bayesian theory,[4] and given the difficulties associated with adapting fuzzy logic implementations, the framework we have chosen for this application has its basis in classic probability theory. In particular, we use graphical modeling approaches to capture probabilistic relationships

[4] Given n mutually exclusive, collectively exhaustive groups in a Bayesian fault space, the equivalent representation in the D-S framework (the frame of discernment, Θ) would consist of 2n elements, which is comprised of all possible subsets of Θ.

among variables. More importantly, this method can learn causal relationships, which are especially crucial in diagnosis work, to enhance understanding of the problem and result in better predictive capabilities. Finally, this approach facilitates the intermingling of different models, and when used in conjunction with statistical techniques, can encode dependencies, forming a unified and intuitive framework for data fusion.

5.4. Graphical Modeling Approaches

Graphical models have been described as a "marriage between probability and graph theory [47]." With the union exhibiting the virtues of each, the result is a powerful tool for handling both uncertainty and complexity. This unified framework for representing probabilities and independencies combines representation for uncertain problems with techniques for performing inference. Because the approach is inherently modular, that is, a complex system can be viewed as a collection of simpler parts, the model is ideally suited for the design and analysis of machine learning algorithms. Probability theory acts as the "glue" for holding the parts, providing consistency within the system, and an interface between models and data. The framework is supplied by graph theory, enabling the visualization of interacting sets of variables. The general graphical model formalism can take various forms. Influence diagrams represent decision processes; Bayesian networks are used for causal, probabilistic processes and expert systems, and data-flow diagrams for deterministic computation. Other special cases include mixture models, linear regression predictors, feed-forward networks, factor analysis, Kalman filters, hidden Markov models, directed graphs representing a Markov chain, and undirected networks such as a Markov field, used to capture correlation for images and hidden causes [48]. These cases span over many fields ranging from systems engineering and statistical mechanics, to information theory, pattern recognition, utility theory and decision theory. Specific applications include diagnosis, probabilistic expert systems, planning and control, dynamic systems and time-series, general data analysis and statistics. Moreover, the graphical model framework facilitates the transfer of techniques over different fields by providing a way to view all cases using the same underlying formalism.

A probabilistic graphical model is a graph whose nodes represent variables, and arcs represent dependencies between variables. Perhaps more important is the absence of arcs; when two variables are not linked, then they can be assumed to be independent. The graph is used to define a mathematical form for a joint probability distribution.

The decomposition of complex problems is based on the idea of independence.

X is independent of Y given Z if $p[(X \wedge Y)/Z] = p[X/Z]p[Y/Z]$
whenever $p(Z) \neq 0$ **for all** X,Y,Z. **D (5.8)**

The law of independence is a basic tool for structuring knowledge [16], [49]. A graphical model can be equated with a set of probability distributions that satisfy its implied constraints. Furthermore, two graphical models are equivalent probability models if corresponding sets of satisfying probability distributions are equivalent [50].

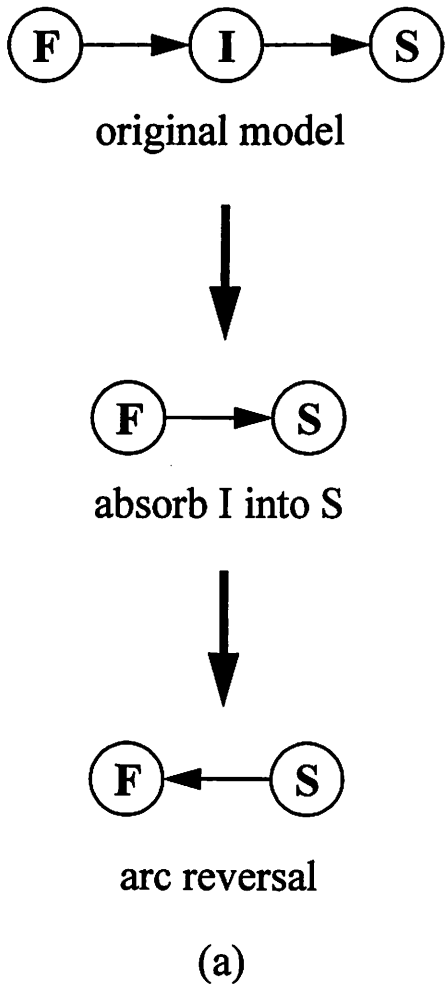
As mentioned above, for implementing diagnostic or probabilistic expert systems, two cases are of particular interest - influence diagrams, and more specifically, Bayesian networks. These are described next.

5.4.1. Influence diagrams

5.4.1.1. Definition

An influence diagram graphically depicts a diagnostic problem by explicitly revealing probabilistic dependence and the flow of information [51]. It enables the incorporation of expert knowledge in a framework to formulate problems as perceived by decision makers. It consists of a network with directed arcs and no cycles, where nodes are random variables and decisions. Arcs into random variables indicate probabilistic dependence, while arcs into decisions specify the information available at the time of the decision. The diagram is compact and intuitive, not only capturing the relationship among the variables, but also providing a complete probabilistic description of the problem. Bayes' theorem forms the backbone of the influence diagram inference procedure. The role of influence diagrams in diagnostic expert systems is to capture relationships between parameters, and to represent and exploit conditional independence where possible.

Influence diagrams can be solved in numerous ways [52], [53], [54]. In particular, in [54], Agogino compares the functional evaluation of Bayes' theorem and the topological transformation in an influence diagram. Figure 5-10 demonstrates a sensor-based inference comprised of a failure node, F, an intermediate node, I, and a sensor node, S.



$$P(S/F) = \sum_{\Omega_I} P(S/I)P(I/F)$$

$$= \sum_{\Omega_I} P(S \wedge I)/F$$

$$P(S) = \sum_{\Omega_F} P(S/F)P(F)$$

$$P(F/S) = \frac{P(S/F)P(F)}{P(S)}$$

(b)

Figure 5-10. (a) Topological transformation and (b) functional evaluation of sensor-based inference with goal: P(F/S) [Agogino, 88]

The probability of a failure, given sensor readings, P(F/S), is evaluated from known values, P(F), P(I/F), and P(S/I). Functional evaluations corresponding to two topological transformations, namely node removal and arc reversal, are shown in Figure 5-10 (b). Formal definitions are as follows:

Arc addition: Any number of arcs may be added to an influence diagram provided no cycles are generated. **D (5.9)**

Arc reversal: An arc between two state nodes may be reversed if there is no other arc generated from the origin to the designated node. D (5.10)

Node removal: Any state node may be removed by absorption into the preceding node, as long as the predecessor precedes only one node. The preceding node inherits all the direct predecessors. D (5.11)

Figure 5-11 depicts a simplified influence diagram for our plasma etch example. The diagram shows that application of power from the top match network influences the plasma state, and that the monitored sensor readings are dependent on the plasma state. We also infer from the diagram that the match network parameters, measured RF power and the DC bias, influence each other, while the endpoint signal is conditionally independent of the match parameters given the state of the plasma.

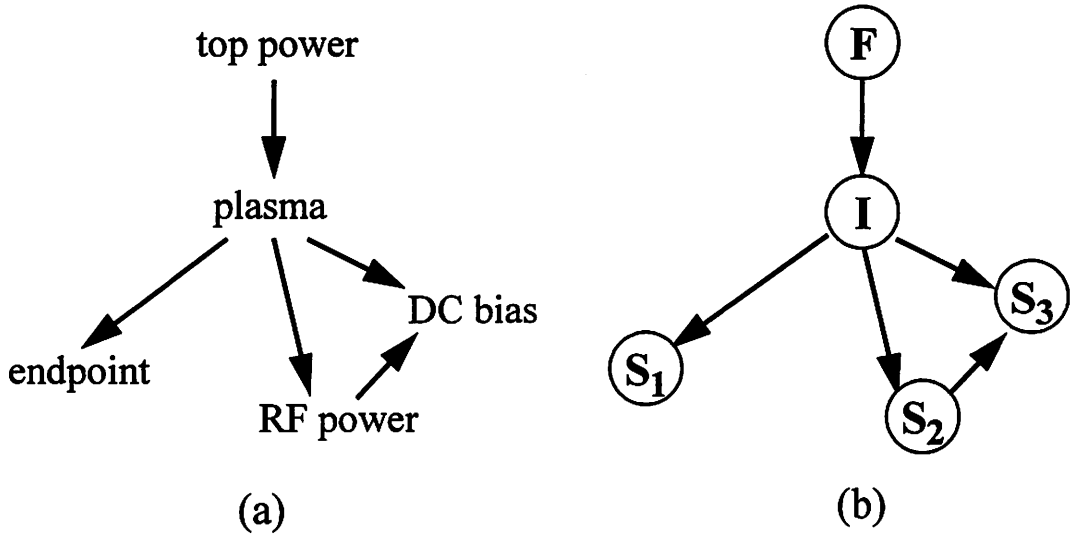
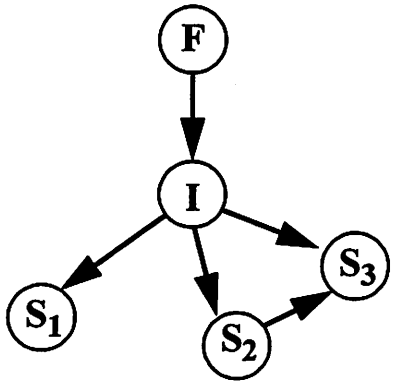


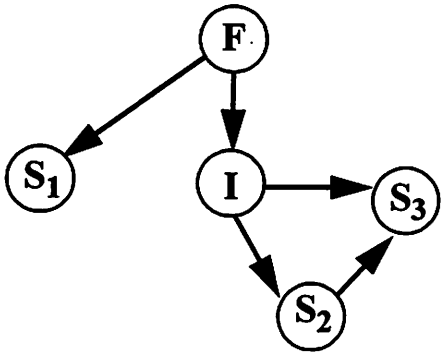
Figure 5-11. Influence diagram using (a) signal names and (b) using labels for failure, intermediate and sensor nodes

Now suppose we are interested in calculating the probability of a wrong top power setting given observations of the endpoint intensity, the measured RF power, and the DC bias. Figure 5-12 shows the topological transformation and functional evaluation for a "wrong top power" fault hypothesis.



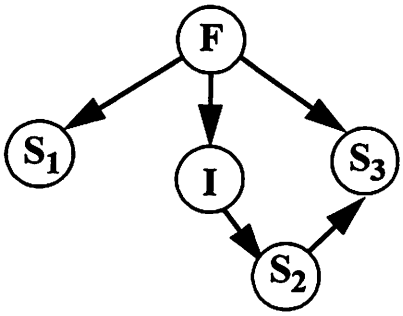
original model

Query : $P[F/(S_1 \wedge S_2 \wedge S_3)]$



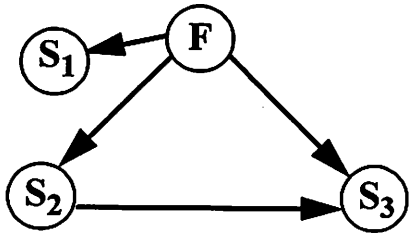
absorb I into S_1

$$P(S_1/F) = \sum_{\Omega_I} P(S_1/I)P(I/F)$$



absorb I into S_3

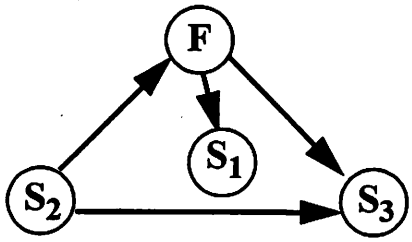
$$P(S_3/F) = \sum_{\Omega_I} P(S_3/I)P(I/F)$$



absorb I into S_2

$$P(S_2/F) = \sum_{\Omega_I} P(S_2/I)P(I/F)$$

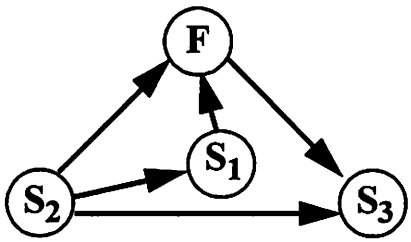
Figure 5-12. (a) Topological transformation and (b) functional evaluation for top power: $P(F/S_1,S_2,S_3)$



reverse arc from S_2 to F

$$P(S_2) = \sum_F P(S_2/F)P(F)$$

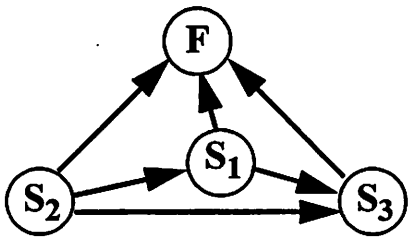
$$P(F/S_2) = \frac{P(S_2/F)P(F)}{P(S_2)}$$



reverse arc from S_1 to F

$$P(S_1/S_2) = \sum_F P(S_1/F)P(F/S_2)$$

$$P[F/(S_2 \wedge S_1)] = \frac{P(S_1/F)P(F/S_2)}{P(S_1/S_2)}$$



reverse arc from S_3 to F.

$$P[S_3/(S_2 \wedge S_1)] = \sum_F P(S_3/F)P[F/(S_2 \wedge S_1)]$$

$$P[F/(S_1 \wedge S_2 \wedge S_3)] = \frac{P(S_3/F)P[F/(S_2 \wedge S_1)]}{P[S_3/(S_2 \wedge S_1)]}$$

The ability to model joint probability distributions using sparse graphs to reflect conditional independence relationships is of key importance for decision theory applications. In addition, multi-attribute utility functions can be decomposed by creating a node for each term in the sum [48]. The parents would be all the attributes (random variables) on which the term depends. Utility nodes would have action nodes as parents. The result is an influence diagram used to compute optimal actions to maximize expected utility.

5.4.1.2. Application examples

There are quite a few successful expert system applications based on influence diagrams. HEATXPRT [55], a data-driven on-line expert system for diagnosing heat rate deg-

radation problems in fossil power plants, uses an influence diagram knowledge base to represent and process uncertainty. In addition, an application to semiconductor manufacturing processes can be found in [19]. The process model combines qualitative knowledge of human experts captured in influence diagrams, and neural networks for extracting quantitative knowledge relating process parameters. The result is an adaptive learning architecture for process modeling and recipe synthesis for deposition rate, stress and film thickness in low pressure chemical vapor deposition (LPCVD) of undoped polysilicon.

5.4.2. Bayesian networks

5.4.2.1. Definition

Bayesian networks, also known as belief networks, are influence diagrams without decision nodes. To define a Bayesian network, a set of variables $X = \{X_1, \ldots, X_n\}$ is specified along with a network structure S, encoding the conditional independence assertions about the variables in X. A set of local probability distributions P is also associated with each variable, and together the components specify the joint probability distribution for X. This graphical model uses directed arcs exclusively; the term directed acyclic graph (DAG) denotes a directed graph without directed cycles. Furthermore, the nodes in S have a one-to-one correspondence with the variables X. Like influence diagrams, Bayesian networks represent a conditional decomposition of the joint probability. We specify the conditioning context as M, which can represent the expert's prior knowledge, or choice of graphical model. Each variable is conditioned on its parents, with $parents(x)$ denoting the set of variables with directed arcs into x.

The general form of an equation for the joint probability distribution for X given S is:

$$p(X/M) = \prod_{x \in X} p[(x/parents(x)), M] \qquad (5.26)$$

Bayesian networks offer several advantages. First, by encoding dependencies among variables this method can handle updates given additional data. Bayesian networks enable learning of causal relationships, leading to better understanding about the problem domain. The approach facilitates the combination of knowledge and data, particularly causal prior

knowledge and causal relationships with probabilities. Finally, the combination of Bayesian methods with Bayesian networks and other types of models offers an efficient and principled approach for avoiding the overfitting of data.

5.4.2.2. Construction of a Bayesian network

The initial tasks in the process of building a Bayesian network are to:

(1) identify the goals of modeling (prediction, explanation, exploration)

(2) identify observations that may be relevant

(3) determine what subset is worthwhile to model

(4) organize observations into variables having mutually exclusive, collectively exhaustive states

The difficulties associated with these tasks are not limited to Bayesian networks, but are common to most approaches [56].

The next phase of Bayesian network construction involves building a directed acyclic graph that encodes assertions of conditional independence for the problem [57]. The mathematical basis for this is the chain rule of probability:

$$p(x) = \prod_{i=1}^{n} p(x_i / (x_1, \ldots, x_{i-1})) \qquad (5.27)$$

Determining the structure of a Bayesian network often entails the use of human expertise and prior knowledge. We specify that for every x_i, there will be some subset $\Pi_i \subseteq \{x_1, \ldots, x_{i-1}\}$, such that x_i and $\{x_1, \ldots, x_{i-1}\} \backslash \Pi_i$ are conditionally independent given Π_i [56]. Once the problem variables are ordered, we can determine the variable sets that satisfy:

$$p(x) = \prod_{i=1}^{n} p(x_i / \pi_i) \qquad (5.28)$$

One difficulty with this procedure is that if the variables are ordered poorly, the resulting structure will fail to encode important conditional independencies. In the worst case, one would have to explore n! possible orderings. An approach to circumvent this undesirable scenario is to examine the causal relationships among variables which often correspond to assertions of conditional independence. Thus, to construct a Bayesian network, one can simply draw arcs from causal variables to their effects. In almost all cases, the result is a structure that satisfies Equation 5.26. To a large extent, the success of Bayesian networks in the implementation of expert systems is due to the learning of causal relationships, also referred to as causal semantics.

The causal and probabilistic semantics in the model allow for the combination of prior knowledge. However, methods for learning causal relationships are still new and controversial. The causal Markov condition defines the connection between causal and probabilistic dependence. In particular, a directed acyclic graph is a causal graph for variables if the nodes are in a one-to-one correspondence and there is an arc from node X to Y if and only if X is a direct cause of Y. The causal Markov condition says that if C is a causal graph for X, then C is also a Bayesian-network structure for the joint physical probability distribution of X. Several researchers have found this condition to hold in many applications [56]. Thus, given the causal Markov condition, we can infer causal relationships from conditional-independence and conditional-dependence relationships learned from data.

The final step in the construction process is to assess local probability distributions of variables given their parents. In this case, specifying the parameters of the model for a Bayesian network means finding the conditional probability distribution (CPD) at each node. For discrete variables, these can be represented as a conditional probability table (CPT) [48].

Although the construction steps above have been described in simple sequence, in practice the steps are intermingled, and it often takes several iterations to formulate the problem based on assumptions of conditional independence, cause and effect.

5.4.2.3. The "Bayes Ball" algorithm

One way to aid in formulating the problem is to view conditional independence relationships encoded by a Bayesian network by the "Bayes Ball" algorithm [48]. Using this algorithm, each node is conditionally independent of its non-descendents, given its parents, and in fact, having this quality is often implied in a Bayesian network. Figures 5-13 and 5-14 illustrate this algorithm. The idea is that two nodes X and Y are conditionally independent (d-separated), given the parents, if a ball is unable to go from X and Y, where the allowable movements of the ball are depicted in the figures. In the first case (a), note that the arrows are directed into the node; by convention, this is a *leaf* node with two parents. If the node is *hidden* (not observed and hence, unknown), as in Figure 5-13 (a), its parents are marginally independent, and the ball cannot pass through. However, if the node is observed, the parents become dependent, and the ball passes through, as shown in Figure 5-14 (a). The second case (b) depicts a *root* node, with arrows directed outward. In this case, if the node is hidden, the children of the node are dependent, linked by a common hidden cause as in Figure 5-13 (b), while if the node is observed, the children are conditionally independent, and the ball cannot pass through as in Figure 5-14 (b). Finally, for the last two cases (c) and (d), the node is an intermediate node, and nodes upstream or downstream are dependent if and only if the intermediate node is hidden.

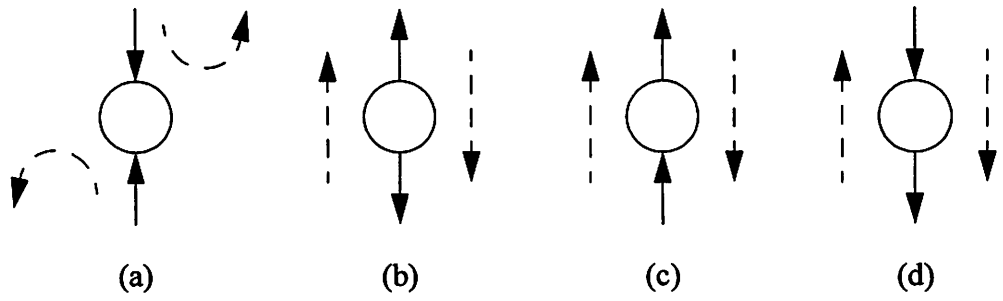


Figure 5-13. Allowable movements of "Bayes' Ball" for hidden nodes

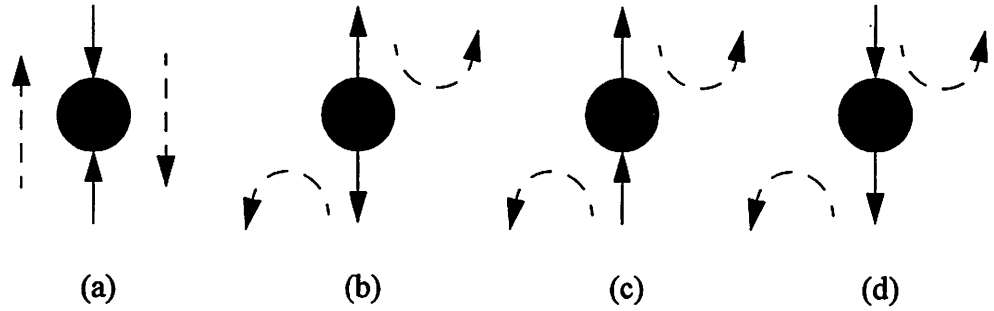


Figure 5-14. Allowable movements of "Bayes' Ball" for observed nodes

The network determines a joint probability distribution and in principle, can be used to calculate any probability of interest using the joint pdf (probability density function). In reality, this is not practical, and encoded conditional independencies are used to simplify the calculation and make it more efficient.

Probabilistic inference algorithms for Bayesian networks with discrete variables have been developed by several researchers. One example reverses arcs in the network using an algorithm that applies Bayes' theorem, so that the result can be directly read from the graph [58], [59], [60]. In contrast, another algorithm, described in [61], utilizes a message-passing scheme to update the probability distributions in the network after one or more variables have been observed. Other examples can be found in [62], [63], and [64], where the algorithms involve a transformation of the network into a tree, with each node in the tree representing a subset of the original variables in X. Mathematical properties of the tree are then used to draw inference.

5.4.2.4. Bayesian networks for probabilistic inference

The most common task, probabilistic inference, uses Bayes' rule to compute posterior probabilities. For discrete nodes with conditional probability tables (CPTs), maximum likelihood (ML) estimates can be calculated by simply counting the number of times an event occurs in the training set [48]. A key advantage of Bayesian networks is the small number of parameters, requiring fewer data points for their estimation. Choosing the form of the conditional probability distribution (CPD) can be nontrivial. One approach is to use

a mixture distribution; however, this introduces a hidden variable. Alternatively, a decision tree can be used, or a table of parent values along with their associated non-zero probabilities.

In order to ensure successful ongoing diagnosis, two components - mechanisms for learning and updating probabilities - must be integrated into the network structure. This means having the capability to refine the structure and local probability distributions given additional data. The idea is to use data mining techniques to combine prior knowledge with data and produce "improved" knowledge. The data are also used to update the probabilities of a given network structure.

One approach consists of updating the posterior distribution for a variable that represents the physical probability. Assuming the physical joint probability distribution is encoded in a network structure, given a random sample, one can compute the posterior distribution. The local distribution function associated with a node is often a probabilistic classification function. Consequently, a Bayesian network can be viewed as a collection of probabilistic classification models, organized by conditional independence relationships. Classification models that produce probabilistic outputs come in many forms including linear regression, generalized linear regression, probabilistic neural networks, probabilistic decision trees, kernel density estimation methods, and dictionary methods. In theory, any of these can be used to capture probabilities in a Bayesian network, and in most cases, Bayesian techniques can be used for learning as well.

Methods for training Bayesian networks from data are still evolving. Statistical methods for using data to improve models, methods for learning parameters and the structure of the network [65], and techniques for learning with incomplete data [66] are all among the active research areas. In terms of algorithm schemas for learning from data, much work has been done using Monte-Carlo methods for approximation, in particular Gibbs sampling and Gaussian approximations, as well as the expectation-maximization (EM) algorithm for finding the maximum likelihood (ML) or maximum aposteriori (MAP) estimates.

5.4.2.5. Application examples

Bayesian networks originally arose out of an attempt to add probabilities to expert systems, and this remains their most common use. Two better known examples include the Windows 95 "troubleshooter" for diagnosing causes of printer failure, and QMR-DT [17], a quick medical reference model, diagnosing diseases from findings, and utilizing what arguably may be considered the largest Bayesian network ever constructed. Yet another successful application is the Vista system [18] used at NASA.

5.5. Sampling theory versus the Bayesian approach

We made a reference in the previous section to "Bayesian techniques". What does this mean exactly? It turns out that there are two camps within probability theory, each with its own distinct approach to solving problems - one based on sampling theory, sometimes referred to as the maximum likelihood approach, and the other, the Bayesian approach. Although the two may sometimes yield the same prediction, particularly for a large number of observations, their conceptual basis is fundamentally different. The sampling theory approach attempts to estimate optimum values for the parameters of a density function by maximizing a likelihood function derived from the training data. Thus, the parameter θ is considered fixed (although unknown), and we must consider all data sets D of a size n that could be generated from the distribution given by θ. The maximum likelihood estimator selects the value of theta that maximizes the probability $P(D/\theta)$. In contrast, the Bayesian approach considers the data set D to be fixed, and we imagine possible values of θ from which the data could have been generated. Thus, the parameters are described by a probability distribution that is initially set to a prior, and then converted to a posterior through Bayes' theorem after observing the data. The final expression is given by an integral over all possible values of θ, weighted by the posterior distributions.

There is also a difference in how one views the idea of probability. The frequentist view (classical approach) defines probabilities in terms of fractions of a set of observations in the limit where the number of observations tends to infinity. In contrast, the Bayesian approach can use the term probability to express a subjective 'degree of belief' in an outcome. Cox [67] showed that a Bayesian formalism could be reached by imposing some

simple natural consistency requirements. Specifically, using the value *1* to denote complete certainty that an event occurs, and *0* for complete certainty that it does not occur, with values in between as degrees of belief, it was found that these behave like conventional probabilities. The mechanism for updating these probabilities with new data is provided by Bayes' theorem.

If sampling approaches are used for hypothesis testing, one main concern is the trade-off between sample size and type *I* and type *II* errors [68]. Although Bayesian theorists do not deal with type *I* and type *II* errors, they do need to assess prior probability distributions. If the classical approach is used to make inferences based on a few samples, the results may be subjective in the sense that a statistic may be significant at a five percent level, but not at a one percent level. Hence, with no strong preference for a specific level, one may be better off calculating the entire posterior with respect to the prior probability. While Bayesian advocates claim that the expectations taken in the classical approach do not make sense, given that we see only a finite data set, the classical statisticians argue that accurate priors, required by the Bayesian approach cannot be assessed in many situations.

It may be considered a blessing that a problem that proves difficult for one approach is sometimes considerably simpler using the other viewpoint. In particular, in the case of multivariate regression and Normal classification problems, with certain prior densities, using the likelihood ratio to test coefficients for significance results in a complicated distribution. However, using the Bayesian approach on the same problem requires a simple application of the Student t-distribution. Another example is the classification of a vector to normal populations with unequal and unknown parameters. Using finite samples and the classical approach leads to questionable results. In contrast, the Bayesian approach computes the posterior odds using the data in a ratio of two Student t-densities [68]. Yet, in other cases, such as principal components and canonical correlations, the Bayesian result is complicated, while application of sampling theory is relatively simple.

The classical and Bayesian approaches have different definitions for a good estimator, and so while both are self consistent, they do produce different estimates. In general, regardless of the model and of which approach is more simple, if priors are accessible, the

Bayesian approach provides a formalism for combining subjective judgement with observed data. If priors are difficult to assess, the classical approach might offer an advantage.

5.6. Model selection and model averaging

There are a number of issues that arise in any modeling problem. For instance, one must consider how to search for good models and how to determine the "goodness" of a model. This leads to the ideas of model selection and selective model averaging. In the former approach, one would select a "good" model and use it as if it were the correct model. In the latter approach, one would select a manageable number of good models and assume that these models are exhaustive. As discussed previously, one can narrow the field of models by considering causality and prior knowledge. In addition, model averaging using Monte-Carlo methods has been shown to yield efficient predictions [56]. Model averaging and model selection lead to models that tend to generalize well to new data.

A Bayesian classifier derives its name from the application of Bayes theorem to the joint probability to get a conditional formula. There are three components of interest. The *prior probability* is sometimes given by a subjective probability over the model parameters. The *sample likelihood*, based on the model assumptions and a given set of parameters, indicates how likely the data sample is. And the *evidence for model M*, forms the basis for most Bayesian model selection.

A Bayes factor gives the comparative worth of two models [50]. This approach can be extended to selecting a single decision tree, rule set or Bayesian network. The basic idea is to compare posterior probabilities of each model given by *P(M/sample)*. The computation requires the prior probability and evidence for each model. In the case of Bayesian hypothesis testing, a comparison would be drawn of the Bayes factor of the null hypothesis as compared with the alternative. In model averaging, predictions of individual models are averaged according to model posteriors.

Evidence and Bayes factors are fundamental to Bayesian methods. Often a complex "non-parametric" model (a model with many varied parameters) is used rather than a

simple model with a fixed number of parameters. Examples of these include decision trees, neural networks and Bayesian networks.

In a typical non-parametric problem one might learn class probability trees from data, and, form a representative set of several models averaged using the following identity:

$$(x/sample) = \sum_i P(M_i/sample)P(x/sample, M_i) \qquad (5.29)$$

Model selection and selective model averaging will prove to be critical in improving the performance of our implementation of a diagnostic system for plasma etch equipment.

5.7. Methods for feature extraction

There are numerous difficulties associated with using raw, unprocessed data directly as input to a classification system. Often, this is simply impractical given the vast amounts of data that are collected for a given application. In addition, the important discriminating information may not be apparent in the raw data, but rather in some summary statistic or transformation of the data into a new representation. Pre-processing and feature extraction refer to this type of action, where a large number of input variables is combined to make a smaller number of variables. This can be accomplished through linear, non-linear, or simple fixed transformations constructed by hand or derived from the initial measurements by automated procedures. Dimensionality reduction can be achieved by discarding a subset of the original inputs, through the use of prior knowledge, or by forming a linear combination of the input variables. Transforming the data into a new representation leads to better class separation, and hence improved performance of the classification network.

The goals of feature extraction and selection can be summarized as follows:

(1) reconstruction of original patterns

(2) parsimonious characterization of patterns

(3) effective discrimination between classes

A further complication is known as the curse of dimensionality. If more parameters are estimated, and more features extracted, more samples are needed to specify these values.

The result is a drop in performance after a certain point. In several instances, reducing the number of input variables can lead to improved performance for a given data set. If a fixed quantity of data is better able to specify the mapping in the lower dimensional space, this can compensate for the loss of information incurred by not using all possible inputs. This trade-off is a consequence of the effects of dimensionality, coupled with a limited data set size. To optimally select features after the extraction process requires some kind of feature evaluation. The probability of misclassification can be used as a criterion to reduce the number of features without reducing performance.

5.7.1. Covariance analysis

In chapter 4, we extended hypothesis testing for means of populations to the multivariate case by using Hotelling's T^2 statistic. The covariance matrix, as specified by Equations 4.4 to 4.6, allowed us to calculate the T^2 statistic and to combine the individual IIND residuals into a single statistical score. However, analysis and description of covariance structures are worth some attention in their own right. In particular, it would be useful to determine whether a common covariance structure exists for observations taken

(1) within the same machine type, and the same fault group

(2) among different machine types, but within the same fault group

(3) among different fault groups, but within the same machine type

In some ways, the covariance structure can be considered a feature of the data, and will give us information as to which variable combinations contribute most to distinguishing between different machines and different fault groups. It is also crucial to know whether we are dealing with identical covariance matrices in our sample groups, as this can determine which classification methods are most suitable.

5.7.1.1. Testing the equality of several covariance matrices

Suppose that we have k populations, and observations with p attributes. The null hypothesis given by

$$H_0 : \Sigma_1 = \Sigma_2 = \ldots = \Sigma_k \qquad (5.30)$$

of the equality of the covariance matrices of k p-dimensional multinormal populations, can be tested using a modified generalized likelihood-ratio statistic.

We take the maximum likelihood estimators for the sample mean and covariance matrix for the jth population as

$$\bar{x}(j) = \frac{1}{n_j}\sum_{i=1}^{n_j} x_i(j) \qquad (5.31)$$

$$S_j = \frac{1}{n_j - 1}\sum_{i=1}^{n_j} [x_i(j) - \bar{x}(j)][x_i(j) - \bar{x}(j)]' \qquad (5.32)$$

Thus, S_j is the unbiased estimator of Σ_j based on ν_j degrees if freedom, where $\nu_j = n_j - 1$ for the case of a random sample of n_j observation vectors from the jth population. When the null hypothesis, H_0 is true

$$S = \frac{1}{\sum \nu_j}\left(\sum_{j=1}^{K} \nu_j S_j\right) \qquad (5.33)$$

is the pooled estimate of the common covariance matrix. For equal sample sizes where $\nu_j = \nu$ for all $j = 1, \ldots, k$, the pooled estimate S simplifies to

$$S = \frac{1}{k}\sum_{j=1}^{k} S_j \qquad (5.34)$$

Define

$$M = \frac{|S_1|^{\nu_1/2}|S_2|^{\nu_2/2}\ldots|S_K|^{\nu_K/2}}{|S|^{\sum_j \nu_j/2}} \qquad (5.35)$$

m is a modification of the likelihood ratio, and varies between 0 and 1, where the value 1 favors the hypothesis. The test statistic, sometimes referred to as the *Box m* statistic, is given by

$$m_{test} = -2\ln m = \sum_j \nu_j \ln|S| - \sum_{j=1}^{K} \nu_j \ln|S_j| \qquad (5.36)$$

Again, for equal sample sizes where $\nu_j = \nu$ for all $j = 1, \ldots, K$, the test statistic simplifies to

$$m_{test} = \nu\left(k\ln|S| - \sum_{j=1}^{k} \ln|S_j|\right) \qquad (5.37)$$

Box [69] has shown that using the scale factor

$$c^{-1} = 1 - \frac{2p^2+3p-1}{6(p+1)(k-1)}\left(\sum_{j=1}^{k}\frac{1}{\nu_j} - \frac{1}{\sum_j \nu_j}\right) \qquad (5.38)$$

the product $m_{test}c^{-1}$ approximates a chi-squared distribution with degrees of freedom $\left(\frac{1}{2}(k-1)p(p+1)\right)$ when sample sizes are large. With equal sample sizes, if all $\nu_j = \nu$ the scale factor becomes

$$c^{-1} = 1 - \frac{(2p^2+3p-1)(k+1)}{6(p+1)k\nu} \qquad (5.39)$$

For k and p less than four or five, and each ν_j around twenty or more, the chi-squared approximation is reasonably good [74]. For larger k and p and small ν_j Box has proposed an F-distribution approximation. Tables of the upper 0.05 critical values of m_{test} have been calculated by Korin [70] for the case of equal ν_j; these have been reproduced by Pearson and Hartley [71]. Gupta and Tang [72] found the exact distribution of the likelihood-ratio

test statistic and tabulated the scale factor for a chi-squared approximation to the distribution of m_{test} for equal (and small) sample sizes.

5.7.2. Linguistic approaches

The methods used in this work all fall into the general category of decision-theoretic multivariate statistical procedures. However, despite its firm established theoretical foundation and countless successful applications, there has still been some criticism of this approach. For instance, some view that the focus on statistical relationships among scalar features has lead to neglecting other structural properties that characterize patterns. It has also been argued that the data compression is sometimes too severe, and that results lead solely to class designation rather than description, rendering the system unable to generate patterns belonging to a class.

As an alternative to the decision-theoretic approach, researchers have considered a linguistic or syntactic model. In this case, patterns are viewed as composed from a language with construction rules specified by a formal grammar. This requires a primitive extractor (as opposed to a feature extractor in the decision-theoretic approach) whose function is to transform the data into a string of symbols or some general relational structure. A structural pattern analyzer uses the formal grammar to parse the string of symbols, thereby constructing a description of the pattern.

Figures 5-15 and 5-16 depict the two approaches in flowchart fashion. Examination of these diagrams reveals some commonalities. For instance, the extraction of features in the decision-theoretic framework is akin to the extraction of primitives in the syntactic approach. Moreover, primitive extraction often involves statistical classification procedures. In addition, classification of patterns into categories in the decision-theoretic approach is similar to the association of patterns with generative grammars in the linguistic case.

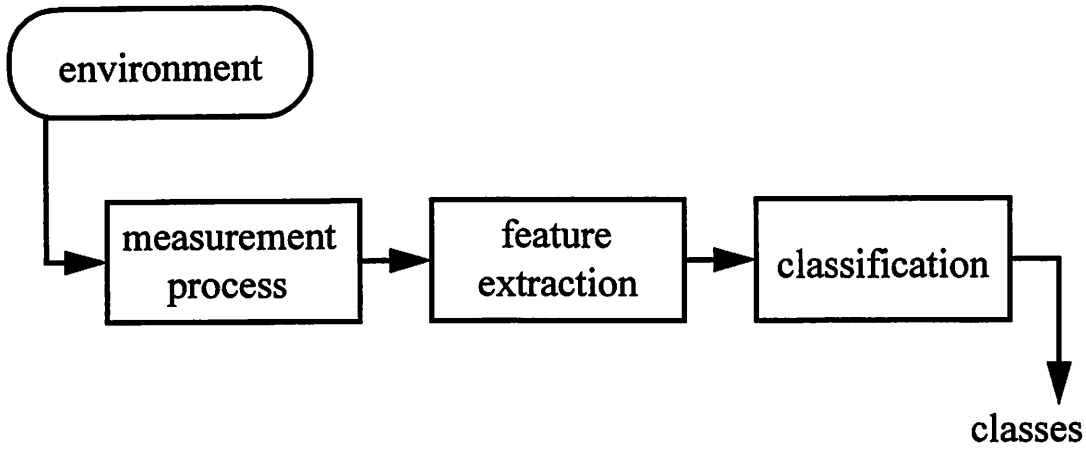


Figure 5-15. Decision-theoretic approach

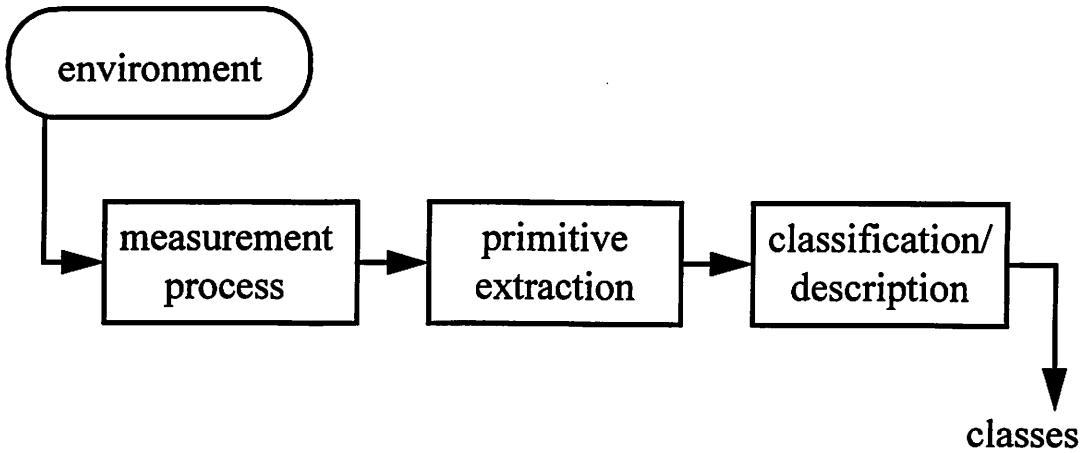


Figure 5-16. Linguistic approach

Of course, there are aspects in which the two approaches differ. The decision-theoretic method utilizes numerical measurements, has no need for explicit structural information and is used primarily for classification. In contrast, in the linguistic approach, primitives are subpatterns that are rich in structure, and the method is used for both classification and description.

Formal linguistic models can use other generative mechanisms including differential equations and finite state Markov chains. There are even stochastic-syntactic models that specify a discrete probability distribution over the formal grammar. Specifically, for prob-

lem involving n classes, there are n stochastic grammars, and each parse provides a structure along with a probability that the structure represents the input pattern. In this case, the input associated with the grammar is the most probable parse.

For an application of classification of sensor signals from plasma etch equipment which focuses on structural properties in profiles of the signals, the interested reader is referred to [73].

5.7.3. Pattern matching

In an effort not to neglect structural properties apparent in the profile of certain signals over time, we have attempted to extract features that capture these properties. The approach taken is one of matching the identified pattern to a template. The pattern is captured in a window, much like one might identify a primitive in the syntactic approach. The data points in the window are used to define a matched filter. By using this template against new observations, we can pinpoint the appearance of a similar pattern in the new signal. We obtain a metric for the goodness of fit of this matching procedure by using a normalized convolution. Essentially, this means that we find the location where the template has the greatest overlap with the profile of the observation.

Of course, there are flaws with this procedure, namely, that the result depends on the template, which is extracted directly from data. With more data samples, this procedure can be greatly improved by using a template that is generated from the data, but takes into account the variability in the different samples. This would require a generating mechanism that accommodates noise and natural variation. For this application, we have chosen a much simplified version to demonstrate that the patterns exist and can be used for classification and diagnosis of certain problems.

5.8. Methods for Classification

Classification models categorize an object based on a profile of its characteristics. If we specify a $px1$ vector of attributes $\vec{z}$, where $\vec{z}$ is an observation from one of k mutually exclusive populations, the problem is to formulate a procedure that discriminates among the populations and makes a decision as to which population $\vec{z}$ belongs. We briefly

describe three approaches for classification used for estimating CPD's and CPT's for the variables examined in this study.

5.8.1. Tree-based models

Tree-based modeling is an exploratory technique which can be used to devise prediction rules, to select variables for prediction, and to examine complex multivariate datasets. The algorithm implementing the construction of tree-based models must determine variables on which to divide, and how to split the space into partitions. It does this by partitioning the space of the predictor variables $\vec{x}$ into homogeneous regions, attempting to make the conditional distribution of the response $\vec{y}$ given $\vec{x}$, $f(\vec{y}/\vec{x})$, independent of $\vec{x}$. The algorithm accomplishes this task by using a criterion minimizing a measure of deviance. The predicted response can be viewed as a factor or as having a numeric value. In the former case, the model constructed is a classification tree, while in the latter it is a regression tree. There are several advantages to tree-based modeling over linear or additive models. In general, the predictor variables can be a mixture of factors or numeric values. The method is invariant to a "monotone re-expression" of the predictor variable. Missing values in the dataset are handled easily, and the factor response is not constrained to have only two levels. Interactions among predictor variables can also be handled by tree-based modeling.

Classification trees are based on the multinomial distribution. If we consider a set, for example, $y = [0, 1, 0]^T$, to represent the response $\vec{y}$ belonging to the second of three factor levels, then the probability corresponding to a response falling into each level would be given by $\vec{\mu} = \{p_1, p_2, p_3\}$, with the constraint $\sum p_i = 1, i = 1, 2, 3$.

The model consists of a stochastic component given by

$$\vec{y}_i \sim M(\vec{\mu}_i),\ i = 1,2,...,n$$

and a structural component

$$\vec{\mu}_i = \tau(\vec{x}_i)$$

The deviance is defined as minus twice the log likelihood

$$D(\hat{\mu}_i, \hat{y}_i) = -2 \sum_{k=1}^{K} y_{ik} \log(p_{ik}) \qquad (5.40)$$

Because the splits in a decision tree are based on maximizing the change in deviance, the mechanism determining the partitions is equivalent to maximum likelihood estimation.

Tree models are evaluated by how well the partition corresponds to the true decision rule. For classification trees, a count of the number of errors as a proportion of the training set provides an estimate of the misclassification rate. Similarly, a probability distribution over the classes is formed from the training set, and using a Bayes decision rule, the algorithm chooses the class with the highest probability as the prediction. Thus, the tree serves as a probability model by providing a probability distribution over each one of the classes.

5.8.2. Generalized linear models

Generalized linear models (GLM's) extend linear models to allow for nonlinearity and heterogeneous variances. In the case for diagnosis, the factor responses can be modeled as binary response data (by grouping two factors together and attempting to distinguish them from the third). This is the approach taken here.

Assuming that the response y is encoded as binary data, the presence or absence of a condition, for example high pressure versus not high (medium or low) pressure, can be treated as a "success" with a value "1", or "failure" with a value "0". This response data has a mean μ, the probability of success, and a variance that depends on the mean. This leads to defining a link function relating the mean to the linear predictors,

$$g(\mu) = \beta^T \hat{x} \qquad (5.41)$$

where the linear predictor is the logit link function

$$\eta = \log\left(\frac{\mu}{1-\mu}\right) \qquad (5.42)$$

or

$$\mu = \frac{e^{\eta}}{1+e^{\eta}} \qquad (5.43)$$

and μ is guaranteed to lie within the range [0,1].

The selection of the logit link is based on the binomial distribution and its corresponding log likelihood function.

Thus the logistic regression model is defined by the logit link and the binomial variance function:

$$V(\mu) = \mu(1-\mu) \qquad (5.44)$$

5.8.3. Sampling Theory and Bayesian Classifiers

Tree-based models and generalized linear models do not make assumptions regarding the distribution of the observations. In the case where we have populations that are normally distributed, there are other options for building classifiers. Specifically, suppose we have a population $\pi_j = N(\theta_j, \Sigma_j), j = 1, \ldots, K$ and (θ_j, Σ_j) are unknown parameters. In addition, we have independent p-variate observations $\{x_1(j), \ldots, x_{nj}(j)\}, j = 1, \ldots, K$. If the covariance matrices are equal, $\Sigma_1 = \Sigma_2 = \ldots = \Sigma_K$, it is easy to find likelihood ratio procedures. However, the distributions required to use these procedures are complicated, and although other techniques from the sampling theory viewpoint are available, these are also not simple to implement.

Fortunately, the Bayesian approach provides a viable alternative for this scenario. The results can be applied with great ease and entail no complicated distribution theory. Taking the definitions for the sample mean and covariance matrix for the jth population from Equations 5-31 and 5-32, we have the following result [68].

Theorem 5-1:

Let z: p x 1 be an observation from one of the populations $\pi_j = N(\theta_j, \Sigma_j), j = 1, \ldots, k$, where the parameters (θ_j, Σ_j) are unknown. If the prior distribution of the parameters is diffuse [68], the predictive probability density for classifying z into π_j is given by the multivariate Student t-density,

$$P(z/data,j) = \frac{k_j}{\left[1 + \frac{n_j}{n_j^2 - 1}(z - \bar{x}(j))' S_j^{-1} (z - \bar{x}(j))\right]^{n_j/2}} \qquad (5.45)$$

where k_j (a constant not depending upon $\hat{z}$) is given by

$$k_j = \left[\frac{n_j}{(n_j+1)\pi}\right]^{p/2} \frac{\Gamma\left(\frac{n_j}{2}\right) p_j}{\Gamma\left(\frac{n_j - p}{2}\right) |(n_j - 1)S_j|^{1/2}} \qquad (5.46)$$

where p_j is the prior probability of classifying $\hat{z}$ into $\pi_j = N(\theta_j, \Sigma_j), j = 1, \ldots, k$. The proof of this theorem can be found in [68].

From Equations 5.45 and 5.46, it follows that the predictive odds ratio for classifying $\hat{z}$ into π_i vis-a-vis π_j, is the ratio of the corresponding multivariate Student t-densities

$$\frac{p(z/data,i)}{p(z/data,j)} = L_{ij} \frac{\left[1 + \frac{N_j}{N_j^2 - 1}(z - \bar{x}(j))' S_j^{-1} (z - \bar{x}(j))\right]^{N_j/2}}{\left[1 + \frac{N_i}{N_i^2 - 1}(z - \bar{x}(i))' S_i^{-1} (z - \bar{x}(i))\right]^{N_i/2}} \qquad (5.47)$$

where L_{ij} is a constant given by

$$L_{ij} = \left(\frac{p_i}{p_j}\right)\left(\frac{|(N_j - 1)S_j|}{|(N_i - 1)S_i|}\right)^{1/2} \left(\frac{\Gamma\left(\frac{N_i}{2}\right)\Gamma\left(\frac{N_j - p}{2}\right)}{\Gamma\left(\frac{N_j}{2}\right)\Gamma\left(\frac{N_i - p}{2}\right)}\right)\left[\frac{N_i(N_j+1)}{N_j(N_i+1)}\right]^{p/2} \qquad (5.48)$$

for $i, j = 1, \ldots, K$.

There are two main advantages for taking the Bayesian approach. The sampling theory approach requires large sample sizes and often, also equal covariance matrices. In this case, Equation 5.45 is valid without these restrictions, and so the result is more generally applicable.

5.9. Summary

In this chapter, we review various methods and approaches for handling uncertainty, contrasting the implementation differences using probability theory, a Dempster-Shafer approach, and FST for classifying faulty behavior in plasma etch equipment. Each method has its merits and drawbacks. In particular, the DS approach requires specification of a large fault space that comprises not only individual fault hypotheses, but all possible subsets of these. In contrast, fuzzy set theory is perhaps best used for handling ambiguity associated with the interpretation of meaning in data, more commonly found in dealing with linguistic variables. In our case, the sensor data collected from plasma etch equipment are largely quantitative, and thus, readily lend themselves to probabilistic representation based on statistical properties.

Our investigation also includes the examination of various modeling techniques for extracting information from data. In particular, these serve as the building blocks of our diagnostic system, providing the mechanism for extraction of relevant probabilities required by our framework. Specifically, we make use of tree-based and general linear models to predict the likelihood of a fault hypothesis given the evidence embodied in the monitored sensor signals. These predictions are combined using graph theory based on exploiting the causal properties represented by a Bayesian network. In addition, this graphical framework is flexible in that it can accommodate the results of other techniques for the extraction of probabilities. Probabilistic procedures, such as covariance analysis, help to identify what assumptions can reasonably be made, and consequently, point to what method is most appropriate. For instance, the calculation of predictive ratios based on the Student-t distribution used in Bayesian classifiers is not dependent on the assumption of equality of covariance structures, and moreover, is much simpler to implement over the sampling theory approach, which requires estimates of parameters for likelihood ratio procedures using complicated distributions.

In summary, this chapter describes the parts of a diagnostic system for machine fault classification, and the glue that holds these parts together. We show how this approach facilitates the intermingling of different models, and when used in conjunction with statistical techniques, how it can encode dependencies, forming a unified framework for data

fusion by combining evidence from multiple sensors. We delve into greater detail in the following chapter, describing mechanisms for pre-processing and feature extraction, modeling and prediction, and implementation issues for application to plasma etch equipment.

6 Plasma Etch Fault Classification

6.1. Introduction

Applications of data mining to real problems exploit relationships among many variables. In the last chapter, we introduced the idea of using graphical models to encode the joint probability distribution for a large set of variables. In particular, our framework for classification is based on Bayesian networks, where we use various modeling techniques to extract the probabilities necessary for inference. This chapter discusses the implementation and these techniques within a framework to integrate information from multiple sensors in order to diagnose failure modes in a plasma etch equipment application.

6.2. Framework for Fault Classification

As stated in the last chapter, the initial tasks in the process of building a Bayesian network are to:

(1) identify the goals of modeling (prediction, explanation, exploration)

(2) identify observations that may be relevant

(3) determine what subset is worthwhile to model

(4) organize observations into variables having mutually exclusive, collectively exhaustive states

The goal of modeling in our case, as applied to plasma etch diagnosis, is to calculate the likelihood of a fault hypothesis given monitored sensor data. Our extensive study of the time-series behavior of these signals provides some information as to what observations may be relevant to achieve this goal. In this chapter, we expand on that initial examination

by investigating features that might prove to be important for classification of other kinds of faulty behavior. In order to do this, we rely on different types of data to identify these other failure modes. Although our models for prediction and feature extraction differ depending on the type of data, the basic classification framework based on Bayesian networks is the same in each case.

The fault classification problem for plasma etch equipment is complicated not only by the different types of data (sensor data which can be considered over various time scales), but also by the different sources of data. In an ideal world we would have access to complete information. That is, each identified problem category would come equipped with the same sensor data, from only one type of machine, processing a single product, under controlled and stable conditions. However manufacturing environments ensure that conditions are not ideal, and hence we must use a collection of heterogeneous data, from different machines, collected under varying circumstances. Accordingly, the tools we use are appropriate given the information available. Although we may employ different models depending on the data, the end result is the same. The model classifies sensor data into a discrete category, assigning a cause or fault to the observation, and because this is accomplished through a training set of data, there is an associated probability with each assignment.

A Bayesian network provides the framework for combining the predictions of these models. Figure 6-1 depicts the basic network used for fault classification for each of the different types of data. The idea is that, for a given case, we have access to a particular type of data that is symptomatic of a specific set of fault conditions. In other words, a fault condition causes a combination of different symptoms embodied by pieces of evidence. This causal effect is represented in the figure by an arc from a fault node to a node representing a combination of evidence. Perhaps even more noteworthy is the absence of arcs among the pieces of evidence. So, while the probability of a combination of evidence, in this case $P(C_j)$, corresponds to the probability of matching observations to the combination of evidence C_j, all the individual pieces of evidence are assumed to be independent of each other.

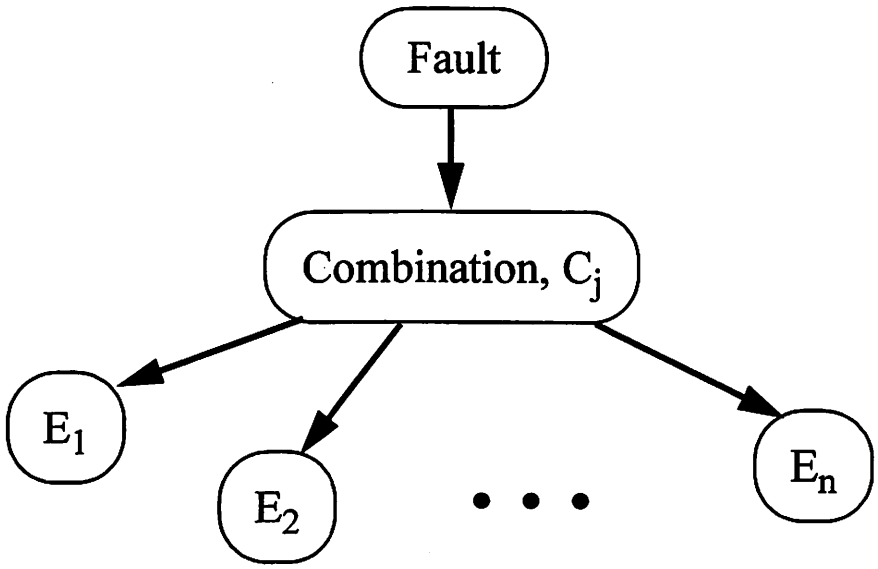


Figure 6-1. Bayesian network for classification

Hence, the probability of a combination, $P(C_j)$, is simply the product of the individual probabilities, $P(E_r)$, of each piece of evidence:

$$P(C_j) = \prod_{r=1}^{n} P(E_{r,s}) \tag{6.1}$$

Note that there are n pieces of evidence, such that $r = 1, \ldots, n$, and each evidence variable, E_r, can take $s = 1, \ldots, m$ values. Thus, the evidence space is divided into N mutually exclusive and collectively exhaustive combinations where:

$$\sum_{j=1}^{N} P(C_j) = 1, \quad N = nm \tag{6.2}$$

In order to make this description concrete, we need to define some terms, and in particular, to explicitly delineate the parts that comprise the fault and evidence spaces for each case under consideration. Recall that, in Chapter 3, we describe in some detail, the types of failure data arising from different sources. These different data are what we are referring to when we consider a particular case.

6.2.1. Case 1: DOE data

For the first case, miscalibrations in the equipment are simulated through DOEs, creating a range of conditions around a nominal operating point. The resulting internal fluctuations in the plasma are captured to some degree by the monitored sensor signals. Figure 5-1 depicts the problem, while Table 5-1 defines a fault and evidence space for a simplified example. In Table 5-1, the fault hypotheses are listed as incorrect settings, for example, "wrong top power". In our implementation, we expand this set of hypotheses to distinguish between incorrect settings that are too high, versus those that are too low. Thus, each fault variable can take one of three values, namely "high", "low" or "medium", where the medium value is assumed to be the correct setting range. Table 6-1 summarizes the fault space for DOE data collected from a Lam TCP 9600 etcher in the J-88-E project described in Chapter 3. Similarly, Table 6-2 contains the fault space for DOE data collected from a parallel plate Lam Rainbow 4400 etcher.

Fault Variable (i)	Fault Index (F_i)
Pressure	F_1
Top Power	F_2
RF Power	F_3
Gas Ratio	F_4
Total Gas Flow	F_5

Table 6-1. Fault Space for DOE Data - Lam TCP 9600

Fault Variable (i)	Fault Index (F_i)
Pressure	F_1
RF Power	F_2
Gas Ratio	F_3
Total Gas Flow	F_4
Gap Spacing	F_5

Table 6-2. Fault Space for DOE Data - Lam Rainbow 4400

Fault Variable ($F_{i,j}$)	i	High (k=1)	Medium (k=2)	Low (k=3)
Pressure	1	$F_{1,1}$	$F_{1,2}$	$F_{1,3}$
Top Power	2	$F_{2,1}$	$F_{2,2}$	$F_{2,3}$
RF Power	3	$F_{3,1}$	$F_{3,2}$	$F_{3,3}$
Gas Ratio	4	$F_{4,1}$	$F_{4,2}$	$F_{4,3}$
Total Gas Flow	5	$F_{5,1}$	$F_{5,2}$	$F_{5,3}$

Table 6-3. Fault indices, $F_{i,k}$, for values (k) taken by each fault variable (i) for TCP 9600

In this case, we find that by using a subset of the monitored sensor signals, tree-based modeling techniques can be combined with GLMs for prediction of failure modes corresponding to changes in the operating conditions. The predictions of these models can be viewed as pieces of evidence. Tables 6-4 and 6-5 list the evidence space for the two experiments mentioned above. Because the pieces of evidence are the model's predictions of faults based on sensor signals, the evidence space mirrors the fault space.

Evidence Variable (r)	Evidence Index (E_r)
Model Prediction of Pressure	E_1
Model Prediction of Top Power	E_2
Model Prediction of RF Power	E_3
Model Prediction of Gas Ratio	E_4
Model Prediction of Total Gas Flow	E_5

Table 6-4. Evidence Space for DOE Data - Lam TCP 9600

Evidence Variable (r)	Evidence Index (E_r)
Model Prediction of Pressure	E_1
Model Prediction of RF Power	E_2
Model Prediction of Gas Ratio	E_3
Model Prediction of Total Gas Flow	E_4
Model Prediction of Gap Spacing	E_5

Table 6-5. Evidence Space for DOE Data - Lam Rainbow 4400

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
Model Prediction of Pressure	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$
Model Prediction of Top Power	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$
Model Prediction of RF Power	$E_{3,1}$	$E_{3,2}$	$E_{3,3}$
Model Prediction of Gas Ratio	$E_{4,1}$	$E_{4,2}$	$E_{4,3}$
Model Prediction of Total Gas Flow	$E_{5,1}$	$E_{5,2}$	$E_{5,3}$

Table 6-6. Evidence indices, $E_{r,s}$, for values (s) taken by each evidence variable (r) for TCP 9600

The tree-based models and GLMs are constructed to provide predictions for each fault hypothesis. Thus, each model directly estimates the probabilities of each value for every fault variable. However, we can also obtain an estimate of these probabilities for a particular fault variable based on combinations of predictions for the remaining fault variables. For instance, suppose we are interested in calculating the probabilities for values taken by the fault variable F_x. In addition to the direct prediction given by the models, based on $P(E_{x,s})$, we are also interested in the combination of the remaining evidence variables, whose probability is given by:

$$P(C_j) = \prod_{r=1}^{n} P(E_{r,s}), r \neq x \tag{6.3}$$

Hence, the probability of a particular combination of evidence is also based on the predictions of the models.

The calculation of each fault probability is based on the relative frequency of the fault given a combination of evidence, denoted by the conditional probability, $P(F_i/C_j)$. A typical Bayesian approach would utilize Bayes' theorem to calculate this probability using the prior probability of a fault, $P(F_i)$, and the conditional probability of a combination given a fault, $P(C_j/F_i)$.

$$P(F_i/C_j) = \frac{P(C_j/F_i) \times P(F_i)}{P(C_j)} \tag{6.4}$$

Alternatively, machine experts can be used to provide direct estimates of conditional probabilities of faults. However, in our case, we derive the conditional probability, $P(F_i/C_j)$, from the data by counting the number of times a fault occurs with a given combination, and dividing this by the total number of times the combination occurs. Thus, the relative frequency of a fault for p observations is given by:

$$P(F_{i,k}/C_j)_p = \frac{\text{number of times } F_{i,k} \text{ and } C_j \text{ occur together}}{\text{number of times } C_j \text{ occurs}} = \frac{n_{F_{i,k}C_j}}{n_{C_j}} \quad \textbf{(6.5)}$$

This conditional probability is easily updated given a new observation, $p+1$, by:

$$P(F_{x,y}/C_j)_{p+1} = \frac{n_{F_{x,y}C_j} + 1}{n_{C_j} + 1}, \text{ for } F_{i,k} \text{ diagnosed as i = x and k = y} \quad \textbf{(6.6)}$$

$$P(F_{i,k}/C_j)_{p+1} = \frac{n_{F_{i,k}C_j}}{n_{C_j} + 1}, \text{ otherwise} \quad \textbf{(6.7)}$$

Finally, the probability of a fault is calculated using the following equation:

$$P(F_{i,k}) = \sum_{j=1}^{N} P(F_{i,k}/C_j) \times P(C_j) \quad \textbf{(6.8)}$$

where the conditional probabilities of faults are taken from the database, and the probabilities of a combination are calculated using the predictions of the models based on observed data and Equation 6.1.

The direct prediction of a fault probability, $P(E_{x,s})$, and the calculation based on the combination of evidence using Equation 6.8, are combined using the model averaging techniques described in the previous chapter. This process is done separately for each of the tree-based and GLM predictors, and the results of these models are also averaged. The calculation of the weights for model averaging is based on model performance. This is determined using misclassification rates obtained from running validation sets comprised of data not used during model construction. Figure 6-2 displays a flowchart outlining the steps for classification of failure modes based on changing input conditions.

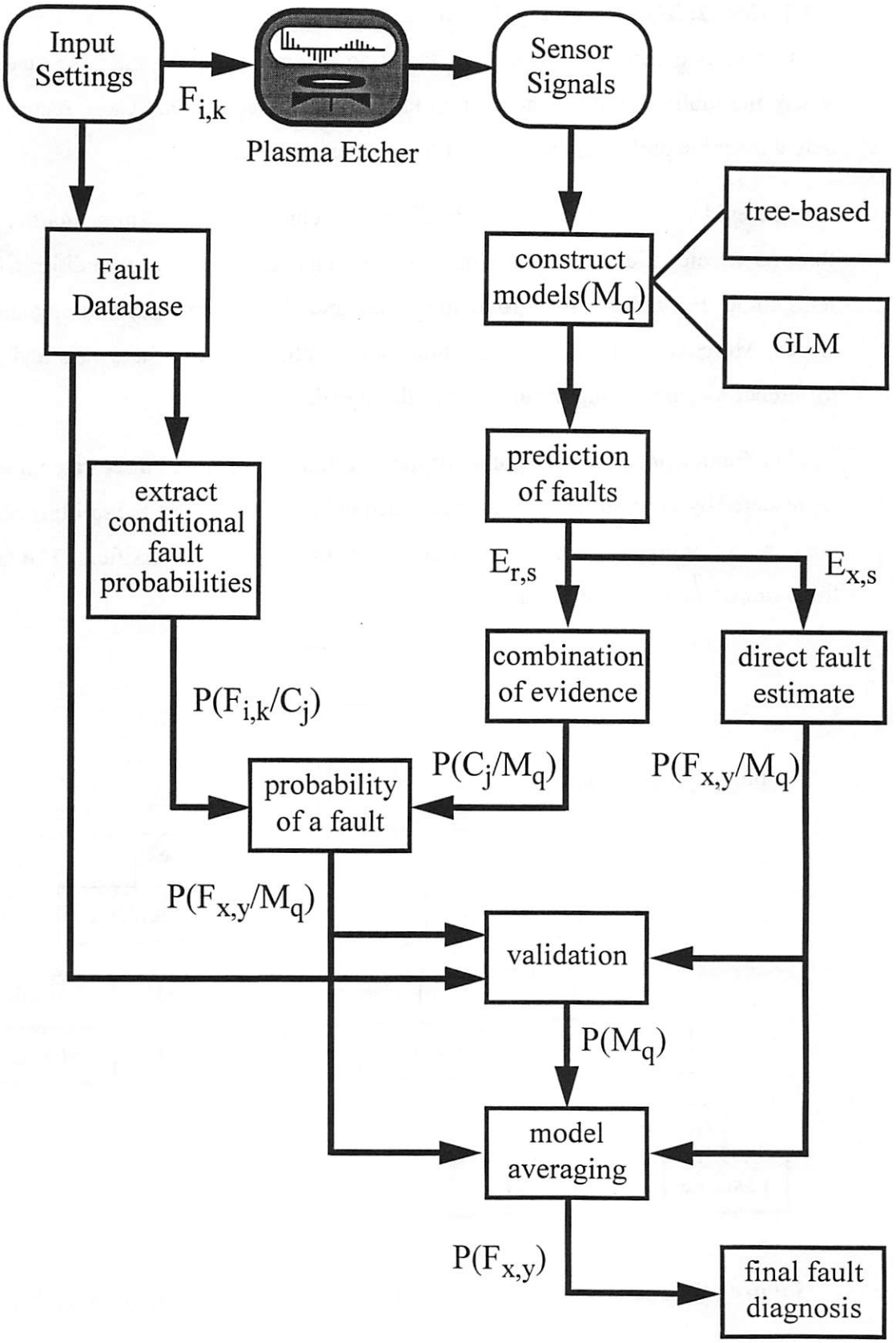


Figure 6-2. Flowchart for calculation of fault probabilities based on DOE data

6.2.2. Case 2: Manufacturing data for machine qualification

Our second case is built on manufacturing data collected for machine qualification, where the faults, diagnosis, and action taken are all documented. These records capture actual machine problems encountered by the manufacturer.

The evidence library, described in Chapter 3, contains qualification data that fall into three basic categories: (1) the baseline, representing normal operating conditions, (2) problems connected with gas line grounding issues, and (3) problems related to the match networks. Moreover, four types of machines are identified, due to hardware and software differences, complicating the analysis of the signals.

The framework for classification of the qualification data into three categories can be represented by a tree structure. This is depicted in Figure 6-3, where the splitting conditions are defined by predictive odds ratios extracted from Bayesian classifiers. The labels for these ratios are described in Table 6-7.

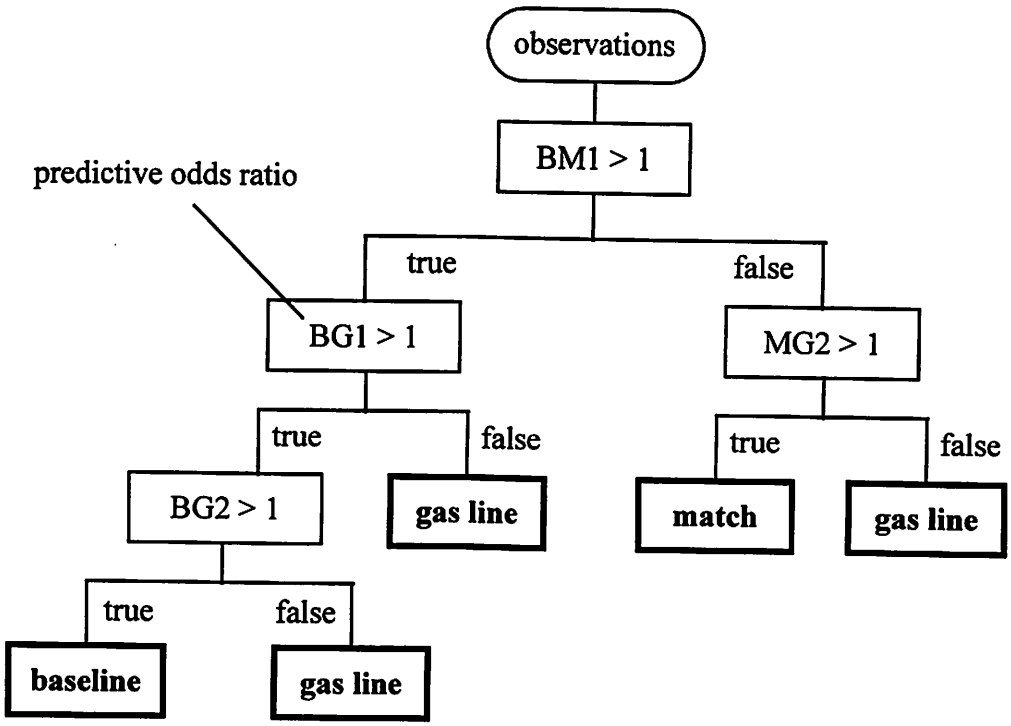


Figure 6-3. Classification tree using predictive odds ratios as splitting conditions

Testing for:	Bottom (RF) Match Signals	Top (TCP) Match Signals
baseline over match	BM1*	BM2
baseline over gas line	BG1*	BG2*
match over gasoline	MG1	MG2*

Table 6-7. Labels for predictive odds ratios for top and bottom match networks

* ratios used in tree structure for classification

Assuming that our observations come from populations that are normally distributed, in the case of unequal covariance matrices, we can calculate a predictive odds ratio for classifying an observation into one population over another. The details of this procedure will be made clear in a following section, but for now, suffice it to say that these ratios provide us with tests or measures for classifying an observation into one group as opposed to another. In this case, we have three groups (baseline, gas line, match), and four observation vectors, corresponding to the tune and load capacitor positions of the top and bottom match networks, respectively. We consider the top and bottom signals separately, leading to six predictive ratios. Thus, we can conduct pairwise comparisons to test (separately using the top and bottom match network signals) whether an observation belongs to: (1) baseline or gas line grounding (2) baseline or match network or (3) match network or gas line grounding. Table 6-7 summarizes the signals or observations, and the predictive odds ratios calculated to classify these observations into populations corresponding to fault categories.

6.2.3. Case 3: High speed data for RF match problems

The third and final case we consider involves isolating and recognizing features in the transient behavior in RF signals triggered by the onset of plasma ignition. The objective in this case is to identify cues relating to predictions of RF match problems, and conditions where the plasma will not ignite. Because the impedance of the plasma changes after ignition, the parameters of both RF match networks can also undergo drastic changes while attempting to adjust to the changing impedance. We focus on the load and tune positions as key variables to monitor, and note the change in the profile of the measured impedance. We simulate the adjustment of the match networks responding to a changing chamber state by varying the preset values for the positions of the load and tune capacitors.

The features we identify are structural, in the sense that there appears to be a pattern in the profile of the impedance signal over time, depending on the "fault" conditions determined by the preset values for the load and tune capacitors. Once these features are captured and linked to a fault condition, they can be considered as pieces of evidence whose combination can lead to a specific diagnosis. Hence, for this analysis, we apply the framework depicted in Figure 6-1, where a fault condition causes a combination of evidence, in this case, comprised of the presence or absence of a given feature found in the profile of the measured impedance.

6.3. Case 1: Models for Predicting Changing Input Conditions

Designed experiments are used to simulate conditions caused by miscalibrations in the equipment. The classification framework relies on models to extract the probability of a fault cause given observed monitored sensor signals. Techniques to predict the various different operating conditions utilize data collected from two types of plasma etchers- a Lam Rainbow 4400, and a Lam TCP 9600. Classification results are obtained using data from the DOEs to train and validate the system. In particular, we explore two different modeling techniques: (1) a simple decision tree structure is used to distinguish between three factor levels of each of the input settings and (2) generalized linear models are used to predict binary responses. In the latter case, the binary response is defined by grouping the three factor levels into two groups.

6.3.1. Monitored signals for the Lam Rainbow 4400

The monitored signals used are those suspected to be most sensitive to changes in the chamber state of the etcher. These signals are known as real-time tool signals and are collected while wafers are being processed at a rate of 1 Hz. The changes we wish to detect and classify in this section correspond to specific shifts in the input settings of the machine. The assumption is that abnormal machine behavior will manifest itself in a manner which can be simulated by a change in the input settings. For the Lam Rainbow 4400 DOE data, there are five input settings which are varied over three levels according to a central composite design; this is summarized in Table 6-8. The design includes 36 runs with 9 centerpoints and is meant to cover a range of different faulty operating conditions. The purpose

of the models is to predict these factor response variables based on the signatures of real-time tool data. The signatures are represented by the average value of each real-time signal over the main-etch period for each of the 36 wafers. For model training and validation, the data set is divided into two mutually exclusive sets by arbitrarily picking 12 runs out of the 36 to use as a validation set.

Response	High	Low	Medium
Pressure (mT)	480	370	425
RF Power (W)	315	235	275
Gas Ratio	0.48	0.42	0.45
Total Flow (sccm)	620	540	580
Gap Spacing (cm)	0.9	0.7	0.8

Table 6-8. Input settings for the Lam Rainbow 4400 plasma etcher

6.3.2. Monitored signals for the Lam TCP 9600

The designed experiment conducted on a TCP 9600 etcher during the J-88-E project is comprised of 56 runs varying five input variables covering a range of different faulty operating conditions. Unlike the previous experiment, the values for the input settings did not fall into three discrete groups, so we used a range, given by Table 6-9, around the center-point to determine three levels corresponding to values of high, low, and medium. As before, the signatures are represented by the average value of each real-time signal over the main-etch period for each of the 56 wafers. As in the previous case, for model construction, the 56 runs of the designed experiment are divided into two sets- a training set of 36 runs to build the models, and a validation set of 20 runs to test the performance of the models.

Response	High	Low	Medium
Pressure (mT)	15-20	7-15	11-15
Top TCP Power (W)	375 - 450	250 -325	325 - 375
Bottom RF Power (W)	137 - 150	110 -125	125 - 137
Gas Ratio	1.06 - 1.15	0.85 - 0.94	0.94 - 1.06
Total Flow (sccm)	155 - 170	130 -145	145 - 155

Table 6-9. Input settings for Lam TCP 9600 plasma etcher (discretized to three levels)

6.3.3. Signal selection

The probability of a high, low, or medium value for an input setting to a plasma etcher is determined using real-time tool signals collected from the plasma chamber as predictors. First, boxplots are used to view the distributions of the real-time tool signals as a function of each input setting. This determines a preliminary set of predictor variables to be used for modeling. Tables 6-10 and 6-11 summarize the real-time signals identified as potential predictors for the factor responses. These signals reflect changes in the machine state which are in turn affected by changes in the input settings.

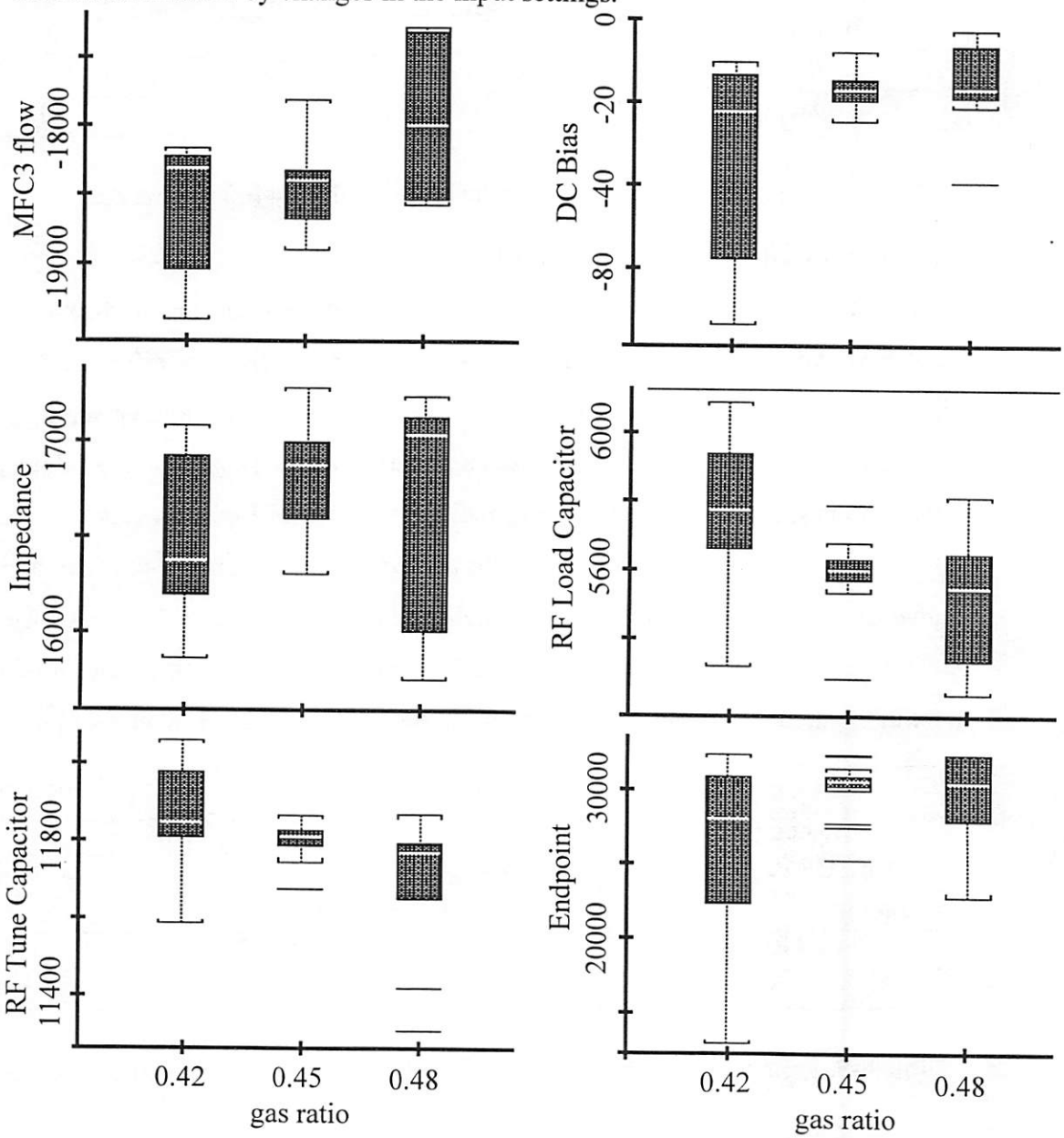


Figure 6-4. Boxplots for the gas ratio input setting using six real-time tool signals*

* These signals include a mass flow calibration (MFC3) measurement, DC bias, impedance, RF load capacitor position, RF tune capacitor position, and endpoint measurements.

The boxplots in Figure 6-4 indicate that the distribution of values is such that one can distinguish clearly between high and low levels of an input response. However, the medium value range appears to overlap with the boundaries of the high and low levels, which is not surprising. Also note that the high and low levels span a greater range of values than the medium level.

Response	Predictors
Pressure	DCBias, Power, Phase, Impedance, RFLoad
RF Power	DCBias, EndpointA, EndpointB
Gas Ratio	RFTune, RFLoad, MFC3, Impedance, DCBias, EndpointC
Total Flow	MFC3, MFC6, HeCFlow, Impedance, Pressure
Gap Spacing	RFTune, RFLoad, Phase, Impedance, Volt, DCBias, EndpointC, Pressure

Table 6-10. Predictor variables for input setting responses - Lam Rainbow 4400

Response	Predictors
Pressure	EndpointA, EndpointB, RFLoad, RF Impedance, TCP Tune
Top TCP Power	RF Impedance, DCBias, TCP Impedance, EndpointA
Bottom RF Power	TCP Tune, TCP Load, TCP Impedance, EndpointA
Gas Ratio	DCBias, RF Phase, EndpointA
Total Flow	Pressure, RF Impedance, TCP Tune, TCP Load, DCBias

Table 6-11. Predictor variables for input setting responses - Lam TCP 9600

The signal selection, model construction and validation are implemented using S-PLUS software in an S-PLUS environment [77].

6.3.4. Tree-Based Model Construction and Validation

6.3.4.1. Tree Construction and Representation

Classification trees for each factor response (input setting) are constructed from the training data using the preliminary set of predictors identified in Tables 6-10 and 6-11. These trees are represented graphically in a block diagram form, as depicted in Figure 6-5, with the root of the tree at the top and the leaves at the bottom. The split condition is used to label each node, and the final selection value to label terminal nodes or leaves. These

nodes have corresponding probability distributions which provide an estimate of the likelihood of each category. For instance, if the terminal node contains 10, 20, and 200 data points in categories A, B and C respectively, the corresponding probabilities would be calculated as P(A,B,C) = (10/230, 20/230, 200/230) = (0.045, 0.09, 0.865), so the model would favor category C with a probability of 0.865. Figure 6-6 depicts a tree model constructed for the factor response RF power using the Endpoint sensor signal as a predictor.

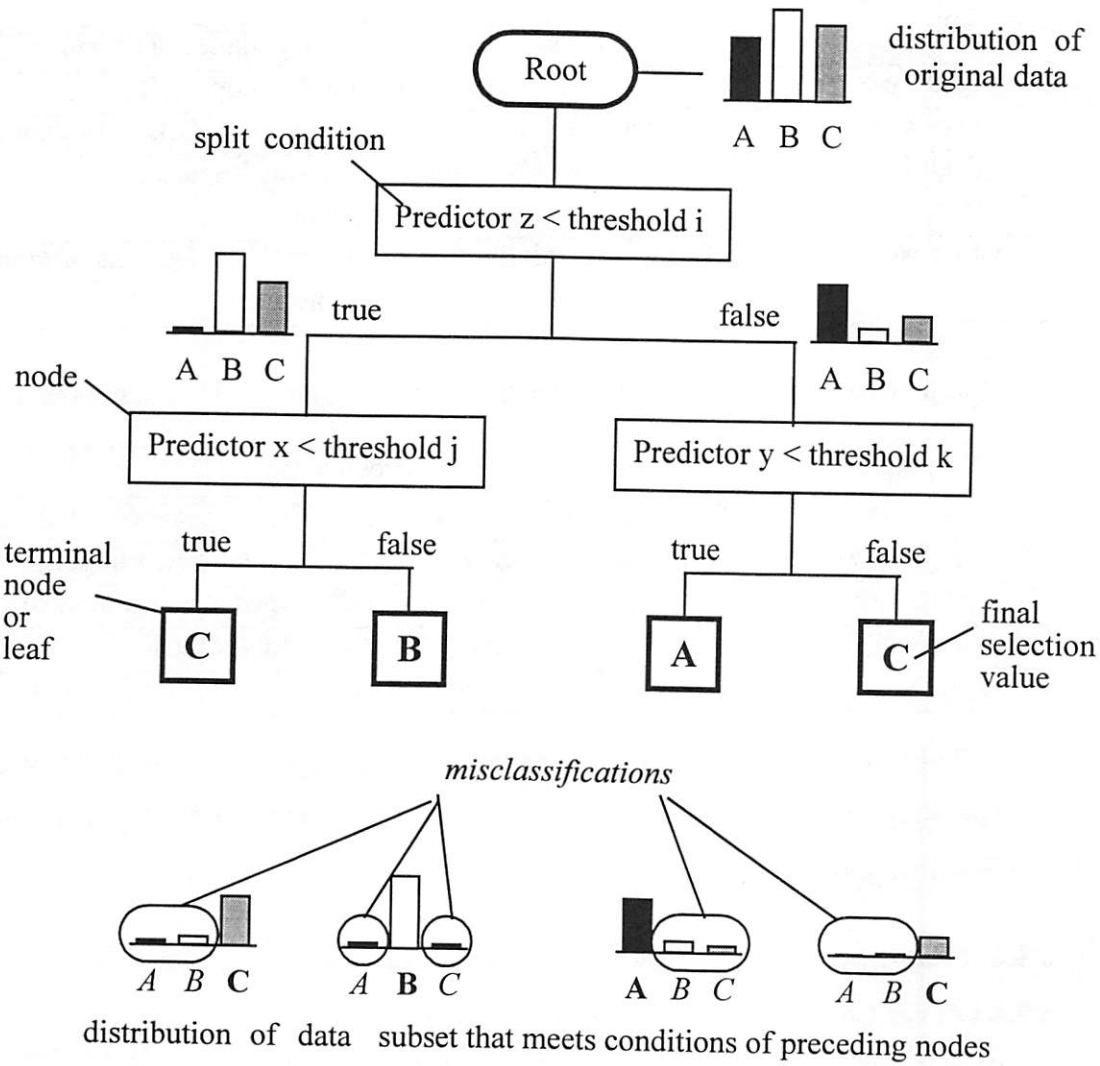


Figure 6-5. Graphical representation of a tree-based model to choose among categories A,B,C.

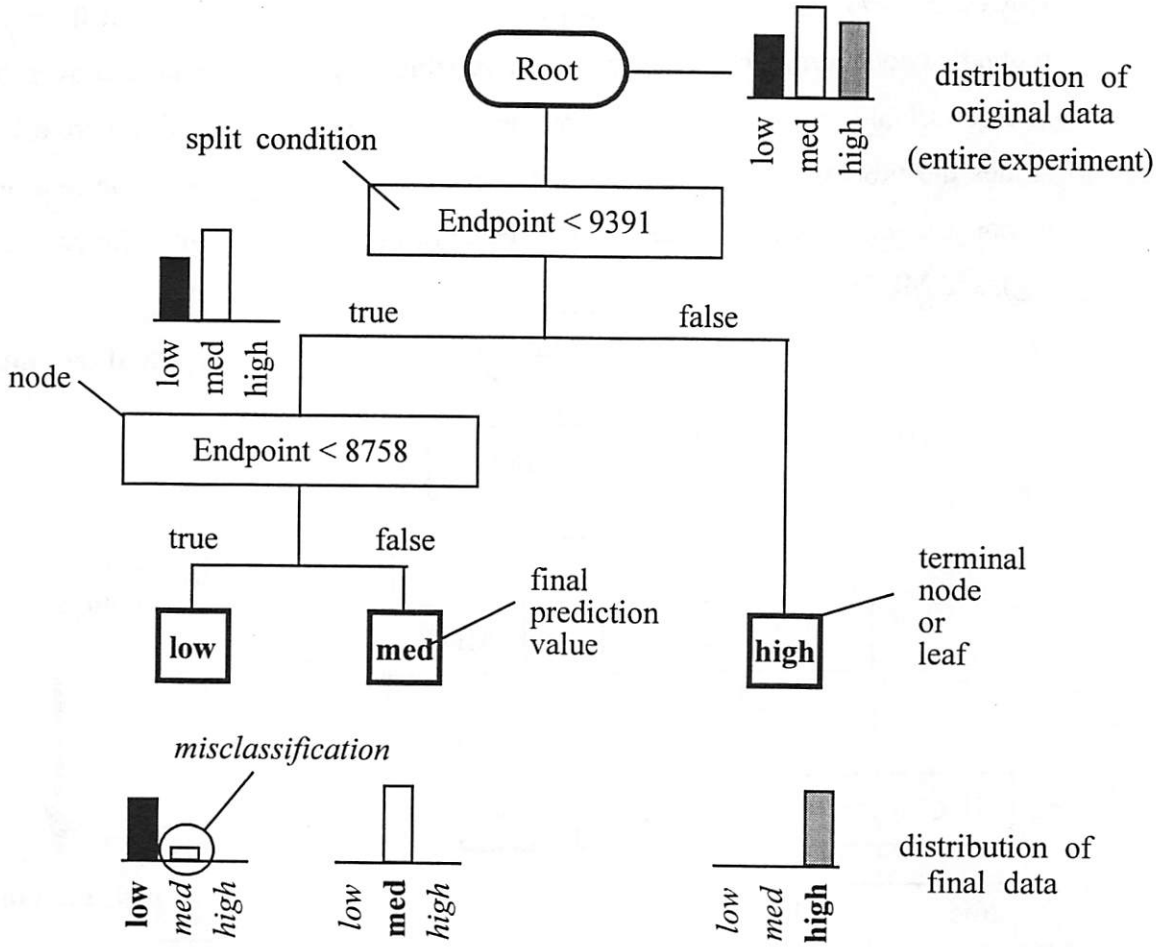


Figure 6-6. Tree model choosing among high, medium, and low RF power using Endpoint

6.3.4.2. Tree Simplification

In any methodology for data-based modeling, there is a chance that the model will be built such that it fits the training data too well, degrading its value as a predictor for other datasets. Pruning and shrinking are methods of simplifying trees; however, because the trees developed in this study are all relatively simple and easy to analyze, the use of these methods for simplification was not necessary. However, viewing these tree-based models as decision trees, it is evident that simplification can be achieved by "snipping" unnecessary nodes. In other words, nodes that do not contribute to improving the final prediction (i.e. decreasing the misclassification rate) can be removed from the tree, effectively merging these nodes to their respective "parent" nodes. By definition, because these nodes are redundant or unnecessary, removing them does not increase the misclassification rate.

However, removing them does create another advantage. Specifically, if the split at the redundant node introduces a new predictor variable, snipping that node removes the effect of that variable and thus reduces the predictor space to be partitioned. Figure 6-7 demonstrates the process of node removal for a tree constructed to predict the total gas flow response using the sensor signals MFC3 , HeCFlow (the flow of helium for backside cooling), and MFC6.

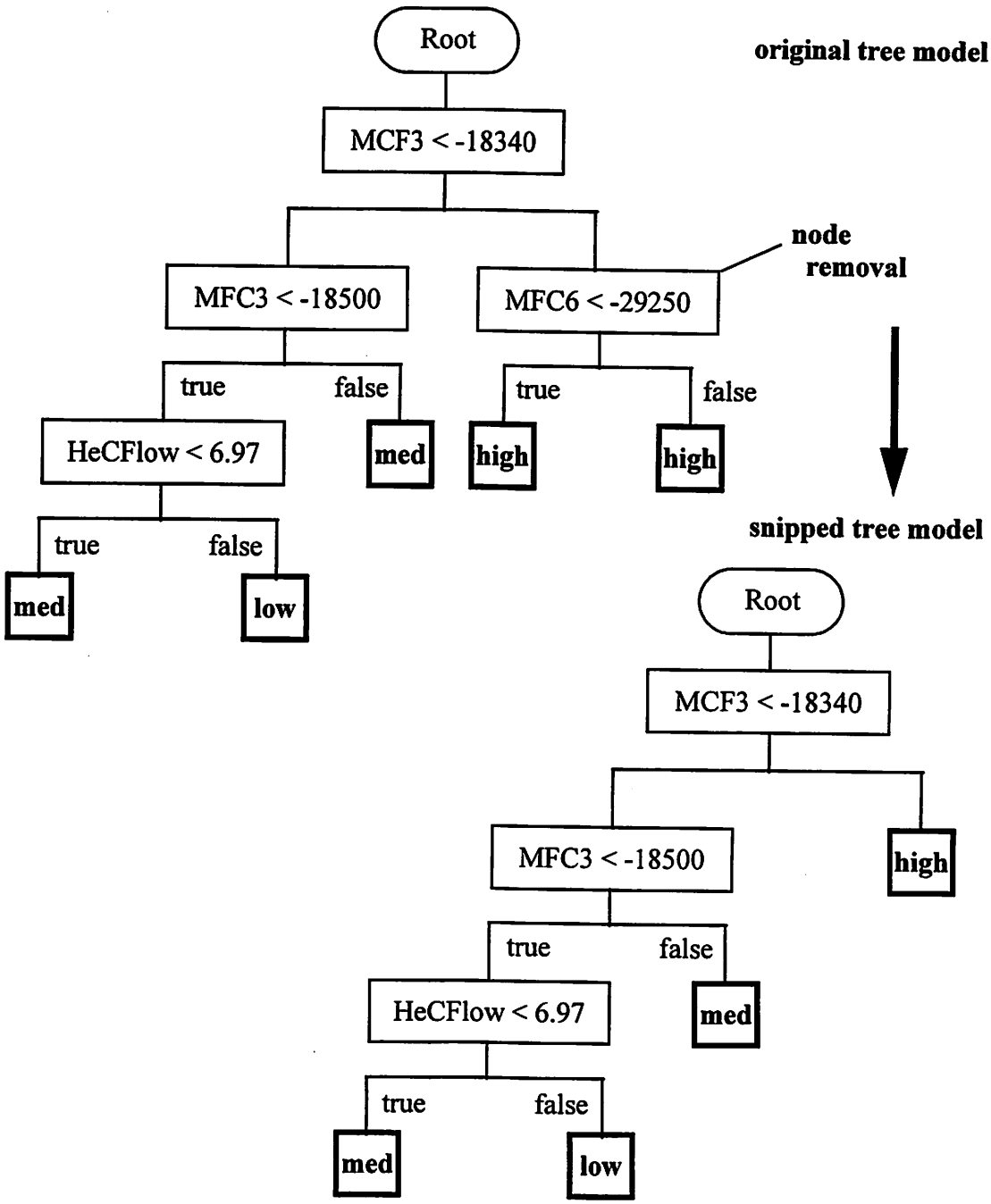


Figure 6-7. Node removal for a tree model used to predict the Total Gas Flow response

6.3.4.3. Coding of Trees

Original construction of the tree-based models is done using S-PLUS software. As described above, the models can be viewed graphically, but can also be represented as a collection of rules. For the two classification trees depicted in Figures 6-6 and 6-7, the equivalent rule-based models and corresponding probabilities for high, low and medium, calculated from the training data set are given by:

(i) *Rule-based model for classification of RF power response:*

If Endpoint is less than 9391, RF power is high - P(H,L,M) = (1,0,0)
Else, if Endpoint is less than 8758, RF power is low - P(H,L,M) = (0,0.8,0.2)
Else RF power is medium - P(H,L,M) = (1,0,0).

(ii) *Rule-based model for classification of Total Gas Flow response:*

If Endpoint is less than 9391, then RF power is high - P(H,L,M) = (0.7,0,0.3)
Else, if Endpoint is less than 8758, then RF power is low - P(H,L,M) = (0,1,0)
Else RF power is medium - P(H,L,M) = (0,0,1).

Once the threshold values are determined, these rules are implemented in a matlab environment, with a function to update the probabilities automatically given new observations. Appendices C and D contain the code for the tree-based models, while the results for extracting probabilities of categories during classification of the validation data for the two different etchers can be found in Appendices E1 and E2. These probability estimates are the values used as evidence, $E_{r,s}$, as described in section 6.2.

The tree-based models are constructed using a subset of the original data, also known as the *training set*. Once the parameters of the models have been determined, in this case the choice of predictors and threshold values defining the split conditions, the performance of the models is judged by how well they are able to classify new observations. These data are referred to as the *validation set*, and we can quantify performance in terms of misclassification rates derived from this data. Thus, for each model, we can keep a tally of how many observations are correctly classified. This information is saved in our database and serves two purposes: (1) by monitoring the performance of the models, we have a measure

of how often the models require updating and (2) the performance serves as a measure of goodness of the model, giving us information as to the likelihood of the model given the data. Hence, we have a method of accounting for the case where machine behavior has been altered so drastically that the models built from data in the past are no longer valid. Furthermore, we have a measure of how good the current model is given the data, and can use this measure to combine the current models with other classification results.

6.3.4.4. Summary of Tree-Based Models

Classification trees are instrumental in screening predictor variables, determining those with the strongest discriminatory ability in terms of predicting response factors. This analysis provides a meaningful breakdown of a complex multivariate set in a form from which conclusions may easily be drawn. In particular, tree-based models are shown to be effective in detecting changes in the input settings using only a small subset of the real-time tool signals. In most cases, the trees are reduced to operate on a space defined by only two predictor variables, without an increase in the misclassification rate. In addition, tree-based methods allow the combining of both numeric variables and factors, and can model factor response variables with more than two levels. Finally, tree-based models can be used as a tool for examining relationships among variables, providing valuable insight for decision making.

6.3.5. Generalized Linear Models (GLMs) for Classification

Using the same data sets for the two different types of plasma etch equipment described above, two sets of generalized linear models are built by encoding each factor response into a binary response. The first set is based on the high level as a "success" encoded with value "1", while the medium and low levels are grouped together as a "failure" and encoded with value "0". The second set reverses the high and low roles, with *low* being a "success" encoded with value "1", and *medium* and *high* together encoded as "0".

GLM models are constructed using the training data set to predict the probability of success for each factor response. As in the building of classification trees, the linear predictors for the models are chosen using the same set of preliminary variables identified for

each factor in Tables 6-10 and 6-11. For example, the form of the model fitted for predicting the RF power response is represented symbolically as:

$$\text{logit}(\mu) = \alpha + \beta^T x \tag{6.9}$$

where μ is the probability of the RF power response taking the value "high" for the first set of models. Recall from chapter 5 that the linear predictor is the logit link function

$$\eta = \log\left(\frac{\mu}{1-\mu}\right) \tag{6.10}$$

or

$$\mu = \frac{e^{\eta}}{1+e^{\eta}} \tag{6.11}$$

The values of the model coefficients α and β can be found in Appendix F, along with the linear prediction η, and corresponding probability values, μ.

A measure of goodness for the models is calculated using the formula:

$$D_{\mu_0} - D_{\mu} \sim \chi^2_{p-q} \tag{6.12}$$

In other words, the difference between the null and residual deviance is tested on the Chi-squared distributed with degree of freedom equal to the difference in the degrees (p-q) of the null and residual deviance respectively. All of the models are found to be significant according to this test. Model validation is conducted on the remaining set of runs not used in building the models.

To combine the results of the GLM models with those of the tree-based models, we need to convert these results into probabilities for the three levels of high, low and medium. Using the predictions of the probabilities of "high", denoted by μ_H, determined by the first set of GLMs, and the predictions of the probabilities of "low" from the second set, denoted by μ_L, if the sum of the two, given by:

$$P(high) + P(low) = \mu_H + \mu_L \tag{6.13}$$

is greater than one, then

$$P(high) = \frac{\mu_H}{\mu_H + \mu_L} \tag{6.14}$$

and

$$P(low) = \frac{\mu_L}{\mu_H + \mu_L} \tag{6.15}$$

Otherwise, we take $P(high) = \mu_H$ and $P(low) = \mu_L$. To determine the probability of a medium value, we use:

$$P(med) = \mu_M = 1 - (\mu_H + \mu_L) \tag{6.16}$$

The probability estimates given by μ_H, μ_L and μ_M, for each of the five input settings are the values used as evidence, $E_{r,s}$, as described in section 6.2. The values extracted from the GLMs can be found in Appendix F.

6.3.6. Modeling Results and Combinations of Evidence

The modeling results for the decision tree and GLM approaches before model averaging are summarized in Tables 6-12 and 6-13.

Setting	Tree (train)	Tree (validate)	GLM (train)	GLM (validate)
Pressure	18 / 6 *	7 / 5	24 / 0	8 / 4
RF Power	23 / 1	11 / 1	14 / 10	7 / 5
Gas Ratio	18 / 6	7 / 5	16 / 8	7 / 5
Total Gas Flow	21 / 3	6 / 6	16 / 8	4 / 8
Gap Spacing	22 / 2	10 / 2	16 / 8	3 / 9

Table 6-12. Classification Results for Lam Rainbow 4400 (correct/ incorrect)* - Direct Prediction of Models using Training Set of 24 runs, Validation Set of 12 runs

Setting	Tree (train)	Tree (validate)	GLM (train)	GLM (validate)
Pressure	31 / 5 *	14 / 6	36 / 0	16 / 4
Top TCP Power	33 / 3	13 / 7	36 / 0	17 / 3
Bottom RF Power	20 / 16	7 / 13	25 / 11	7 / 13
Gas Ratio	31 / 5	9 / 11	36 / 0	10 / 10
Total Gas Flow	26 / 10	9 / 11	24 / 12	5 / 15

Table 6-13. Classification Results for Lam TCP 9600 (correct/ incorrect)* - Direct Prediction of Models using Training Set of 36 runs, Validation Set of 20 runs

The tables display the number of observations correctly classified in the two groups of data collected from the Lam Rainbow 4400, and the Lam TCP 9600, corresponding to the training and validation sets respectively for each model type.

Based on the predictions of the probability values, denoted by $E_{r,s}$, provided by the tree-based and GLM models, we can calculate the probability of a given combination using Equation 6.3. The conditional probability of a fault from the database is extracted using Equation 6.5. Equation 6.8 is then used to calculate the probability of a fault from a combination of evidence. This procedure is conducted separately for the tree-based modeling results, and for the values taken from the GLMs, and the classification results are summarized in Tables 6-14 and 6-15.

Setting	Tree (train)	Tree (validate)	GLM (train)	GLM (validate)
Pressure	18 / 6*	6 / 6	18 / 6	8 / 4
RF Power	16 / 8	8 / 4	17 / 7	7 / 5
Gas Ratio	21 / 3	7 / 5	18 / 6	7 / 5
Total Gas Flow	19 / 5	9 / 3	15 / 9	8 / 4
Gap Spacing	18 / 6	7 / 5	17 / 7	8 / 4

Table 6-14. Classification Results for Lam Rainbow 4400 (correct/ incorrect)*- Based on Evidence Combination using Training Set of 24 runs, Validation Set of 12 runs

Setting	Tree (train)	Tree (validate)	GLM (train)	GLM (validate)
Pressure	24 / 12	8 / 12	25 / 11	7 / 13
Top TCP Power	28 / 8	9 / 11	25 / 11	12 / 8
Bottom RF Power	23 / 13	8 / 12	27 / 9	6 / 14
Gas Ratio	26 / 10	8 / 12	30 / 6	6 / 14
Total Gas Flow	22 / 14	11 / 9	24 / 12	9 / 11

Table 6-15. Classification Results for Lam TCP 9600 (correct/ incorrect)*- Based on Evidence Combination using Training Set of 36 runs, Validation Set of 20 runs

Consequently, we end up with four different estimates for each fault group, namely, two direct estimates, and two estimates using the combinations of evidence, from the tree-based models and the GLMs respectively. This is depicted in the flowchart of Figure 6-2.

Model averaging allows us to combine the four estimates extracted from the different models. Here we take Equation 5-29 to calculate the probability of a specific fault (given the data), denoted by $F_{x,y}$,, from the conditional probability of the fault given the model (and the data), $P(F_{x,y}/data, M_q)$, and the prior probability of the model (given the data), $P(M_q/data)$.

$$P(F_{x,y}/data) = \sum_i P(M_q/data)P(F_{x,y}/data, M_q) \qquad (6.17)$$

The conditional probabilities of the faults given the models (and the data), $P(F_{x,y}/data, M_q)$, are the estimates taken from the four models described above. Hence, we have that $q = \{1, \ldots, 4\}$. The probability or likelihood of a model given the sample, $P(M_q/data)$, is based on the performance of the models. Specifically, the results of Tables 6-12 - 6-15 summarizing the number of "hits", or correct classifications, are used to calculate the weights for the models. Thus, the prior probability of a model, M_1, is given by:

$$P(M_1/data) = \frac{hits_{M_1}}{\sum_q hits_{M_q}} \tag{6.18}$$

Tables 6-16 and 6-17 summarize the results of model averaging for (1) combining the direct estimates with those using combinations of evidence for each model, and (2) combining the tree-based and GLM results to form one final estimate.

Setting	Tree *1 (train)	Tree *1 (validate)	GLM *2 (train)	GLM *2 (validate)	Tree *3 GLM (train)	Tree *3 GLM (validate)
Pressure	23 / 1*	9 / 3	24 / 0	10 / 2	24 / 0	11 / 1
RF Power	24 / 0	12 / 0	17 / 7	8 / 4	24 / 0	12 / 0
Gas Ratio	24 / 0	10 / 2	18 / 6	9 / 3	24 / 0	12 / 0
Total Gas Flow	24 / 0	10 / 2	17 / 7	6 / 6	24 / 0	11 / 1
Gap Spacing	24 / 0	11 / 1	18 / 6	8 / 4	24 / 0	12 / 0

Table 6-16. Classification Results for Lam Rainbow 4400 (correct/ incorrect)* - (1) Tree Combination, (2) GLM Combination and (3) Tree/GLM Combination

Setting	Tree *1 (train)	Tree *1 (validate)	GLM *2 (train)	GLM *2 (validate)	Tree *3 GLM (train)	Tree *3 GLM (validate)
Pressure	35 / 1	17 / 3	36 / 0	16 / 4	36 / 0	19 / 1
Top TCP Power	35 / 1	17 / 3	36 / 0	19 / 1	36 / 0	19 / 1
Bottom RF Power	25 / 11	11 / 9	32 / 4	10 / 10	32 / 4	14 / 6
Gas Ratio	31 / 5	10 / 10	36 / 0	10 / 10	36 / 0	11 / 9
Total Gas Flow	30 / 6	16 / 4	31 / 5	10 / 10	35 / 1	17 / 3

Table 6-17. Classification Results for Lam TCP 9600 (correct / incorrect)* - (1) Tree Combination, (2) GLM Combination and (3) Tree/GLM Combination

The final results display almost perfect classification for all fault groups in the Lam 4400 data, comprised of 24 training samples, and 12 validation runs. Moreover, excellent results were obtained for predictions of Pressure, Top Power and Total Gas Flow for both training sets (36 runs total), and validation sets (20 runs total) for the Lam 9600 data. For

the other two fault groups, Bottom Power and Gas Ratio, the models did well on the Lam 9600 training data, but not as well on the validation sets.

6.3.7. Application Example

As an example, let us take a wafer from the validation set collected from the Lam TCP 9600. This is "Wafer 30" out of a total of 56 runs. First, the sensor signals collected from Wafer 30 are used by the tree-based models and GLMs to calculate a direct estimate of each input response, and a corresponding probability. These are summarized in Table 6-18 under the headings "Tree Direct" and "GLM Direct" respectively. Next, we extract a fault probability estimate based on the combination of evidence for each model. These probabilities are listed under "Tree Combo" and "GLM Combo", corresponding to the tree-based model and GLM results. Model weights, shown in Table 6-19, are calculated using Equation 6-18, and the two estimates for each model type are combined through model averaging. Note that the weights are calculated separately for each input response. Thus, to combine the models for the "Pressure" input response for the tree-based modeling results, we use Equation 6.17:

$$P(F_{x,y}/data) = \sum_i P(M_q/data)P(F_{x,y}/data, M_q)$$

$$P(pressure/data) = P(Tdirect/data) \cdot P(pressure/Tdirect) + P(Tcombo/data) \cdot P(pressure/Tcombo) \quad \textbf{(6.19)}$$

This leads to a combined model estimate for the "Pressure" input response of:

$$P\left(T2\begin{bmatrix} high \\ low \\ med \end{bmatrix}/data\right) = 0.4706 \cdot \begin{bmatrix} 0 \\ 0.0556 \\ 0.9444 \end{bmatrix} + 0.5294 \cdot \begin{bmatrix} 0.2331 \\ 0.2322 \\ 0.5347 \end{bmatrix} = \begin{bmatrix} 0.1234 \\ 0.1491 \\ 0.7275 \end{bmatrix}$$

Repeating this procedure for each input response leads to the results in Table 6-18, under the headings "T2" and "G2", for the combined tree-based models and GLMs respectively. Finally, to combine the tree-based modeling results with those from the GLMs, we use Equation 6.17 again to obtain :

$$P\left(pressure\begin{bmatrix}high\\low\\med\end{bmatrix}/data\right) = 0.5294\cdot\begin{bmatrix}0.1234\\0.1491\\0.7275\end{bmatrix} + 0.4706\cdot\begin{bmatrix}0.1756\\0.4553\\0.3691\end{bmatrix} = \begin{bmatrix}0.1480\\0.2932\\0.5588\end{bmatrix}$$

Input Response	Fault Label	Tree Direct	Tree Combo	GLM Direct	GLM Combo	Tree T2	GLM G2	**Final**
High Pressure	$F_{1,1}$	0	0.2331	0	0.3732	0.1234	0.1756	0.1480
Low Pressure	$F_{1,2}$	0.0556	0.2322	0.4919	**0.4140**	0.1491	**0.4553**	0.2932
Med Pressure	$F_{1,3}$	**0.9444**	**0.5347**	**0.5081**	0.2128	**0.7275**	0.3691	**0.5588**
High TCP	$F_{2,1}$	0	0.1578	**1.0000**	0.2430	0.0888	**0.5992**	**0.3440**
Low TCP	$F_{2,2}$	**1.0000**	0.1605	0	0.2430	**0.5278**	0.1286	0.3282
Med TCP	$F_{2,3}$	0	**0.6817**	0.0000	**0.5140**	0.3835	0.2722	0.3278
High RF	$F_{3,1}$	0	0.2115	**0.3720**	**0.3495**	0.1190	**0.3600**	0.2324
Low RF	$F_{3,2}$	0.0909	0.2730	0.3295	**0.3495**	0.1933	0.3402	0.2624
Med RF	$F_{3,3}$	**0.9091**	**0.5155**	0.2984	0.3009	**0.6877**	0.2998	**0.5051**
High Ratio	$F_{4,1}$	0.2308	**0.4432**	0	0.2908	0.3307	0.1454	0.2381
Low Ratio	$F_{4,2}$	0.1538	0.2181	0	0.3233	0.1841	0.1616	0.1729
Med Ratio	$F_{4,3}$	**0.6154**	0.3387	**1.0000**	**0.3859**	**0.4852**	**0.6930**	**0.5891**
High Flow	$F_{5,1}$	0.1250	0.1869	**0.6287**	**0.3403**	0.1621	**0.4845**	0.3233
Low Flow	$F_{5,2}$	**0.5000**	0.1002	0.1096	**0.3403**	0.2601	0.2250	0.2425
Med Flow	$F_{5,3}$	0.3750	**0.7129**	0.2617	0.3194	**0.5777**	0.2905	**0.4341**

Table 6-18. Fault probabilities for different modeling techniques - Wafer 30

Input Response	Tdir	Tcom	Gdir	Gcom	T2	G2
Pressure	0.4706	0.5294	0.5294	0.4706	0.5294	0.4706
TCP	0.4375	0.5625	0.4706	0.5294	0.5000	0.5000
RF	0.4375	0.5625	0.4667	0.5333	0.5294	0.4706
Gas Ratio	0.5294	0.4706	0.5000	0.5000	0.5000	0.5000
Total	0.4000	0.6000	0.5000	0.5000	0.5000	0.5000

Table 6-19. Model weights for different modeling techniques - Wafer 30

For this example, Wafer 30 happened to be a baseline wafer, which means that it was processed under "normal" conditions corresponding to "medium" levels for each input response. The final diagnosis predicts all responses correctly, with the exception of the TCP (Top) Power, which was diagnosed with probabilities distributed almost evenly among the three levels.

6.4. Case 2: Analysis of Manufacturing Data for Machine Qualification

6.4.1. Covariance Analysis

In Chapter 5, we discuss the importance of finding features in data, and in particular, identifying those which contribute most to distinguishing between different machines and different fault groups. This aspect is crucial in the analysis of the machine qualification data, as variability in the data sets arises not only due to different fault causes, but also because of machine differences.

The examination of the covariance structure within the data serves many purposes. First, if a common covariance structure is found to exist, in particular, for observations taken:

(1) within the same machine type, and the same fault group

(2) among different machine types, but within the same fault group

(3) among different fault groups, but within the same machine type

then we can treat the covariance matrix as a feature that provides pertinent information to distinguish between groups. Even in the case where we do not find any commonalities, the analysis is still of great value in delineating what assumptions we can reasonably make about the distribution of the data. This in turn will influence which classification method we choose to apply.

6.4.1.1. Testing the equality of several covariance matrices

In Chapter 5, we describe the procedure for testing the equality of several covariance matrices. We apply this procedure to the machine qualification data, assuming k populations, and observations with p attributes. The null hypothesis, given by Equation 5.30:

$$H_0 : \Sigma_1 = \Sigma_2 = \ldots = \Sigma_k$$

of the equality of the covariance matrices of k p-dimensional multinormal populations can be tested against the alternative of general positive definite matrices using a modified generalized likelihood-ratio statistic.

The qualification data can be divided into twelve sets, corresponding to different machines. The baseline data, labelled "b1" to "b3", are from three machines of type 1; data diagnosed as a "gas line grounding problem" are collected from machine types 1 and 3; data diagnosed as a "match network problem" are collected from machine types 1, 2, and 4. As shown in Table 6-20, common hardware (labeled as A and B, respectively) is shared by machine types (1,2) and (3,4), while machine types (1,3) and (2,4) use similar software (labeled as C and D, respectively).

Type	Shared		Baseline			Gas Line Grounding					Match Network			
	Hardware	Software	b1	b2	b3	g1	g2	g3	g4	g5	m1	m2	m3	m4
1	A	C	X	X	X	X	X	X					X	
2	A	D									X	X		
3	B	C							X	X				
4	B	D												X

Table 6-20. Qualification data by fault group and machine type (hardware/software differences)

The first case we examine is to test data taken within the same machine type, and the same fault group. Specifically, we take four wafers from each machine (corresponding to a single type and fault group), using a sample set of twenty data points per wafer. Thus, we can consider each wafer to form a population, where the number of populations is $k = 4$, and using Equations 5.31 and 5.32, we can calculate maximum likelihood estimates for the

sample mean, $\bar{x}(j)$, and covariance, S_j, for sample sizes of twenty. Because we are using equal sample sizes ($\nu_j = \nu = 20$) for all $j = 1, \ldots, k$, we can apply Equation 5.34 to calculate the pooled estimate of the common covariance matrix, S. Hence, we have all the variables necessary to calculate the *Box m* statistic, m_{test}, using Equation 5.37:

$$m_{test} = \nu\left(k\ln|S| - \sum_{j=1}^{k} \ln|S_j|\right)$$

where $\nu = 20, k = 4$, and S_j and S are calculated as described above.

Table 6-21 summarizes the *Box m* test statistics computed for four types of machines (with hardware and software differences), and three fault groups. The machines are listed by fault group and type along the first two columns. The total number of variables used is denoted by "p", with the specific sensor variables (as listed in the heading) marked with an "X". The final four columns summarize the results for the *Box m* test.

The "*Box m* test" column contains the information necessary to determine if there is enough evidence to reject the null hypothesis, H_0, of equal covariance structures among the populations. In particular, we list the number of variables used in the computation under the heading "p", and the calculated *Box m* test statistic using Equation 5.37 under "m_{test}". As mentioned in Chapter 5, in the case of equal sample sizes, tables of the critical values for m_{test}, have already been tabulated, particularly for small k (number of populations) and p (number of variables). We list these critical values under the heading "*crit*". Note that this value changes depending on p, that is, depending on the number of variables used in the calculations. If $m_{test} \geq crit$, we reject the null hypothesis and conclude that the populations being tested do not share a common covariance structure. The conclusions are summarized under the final column which specifies if m_{test} is greater (yes = "Y") than the critical value and we reject the null hypothesis, or whether it falls under the critical value (no = "N").

Machine		Bottom (RF) Match					Top (TCP) Match					Clamp		*Box m* test			
Fault	Type	pw	ld	tn	ph	im	pw	ld	tn	ph	im	pr	fl	p	m_{test}	crit	>
b1	1	X	X	X		X		X	X					6	74.67	70.17	*
		X	X	X		X								4	25.48	48.47	N
b2	1	X	X	X				X	X					5	53.3	70.17	N
							X	X	X					3	28.07	31.13	N
b3	1							X	X					2	196.5	17.77	*Y*
g1	1					X				X	X	X		4	45.21	48.47	N
						X				X	X		X	4	45.94	48.47	N
g2	1				X	X				X		X	X	5	58.70	70.17	N
g3	1		X	X	X	X						X	X	6	54.05	70.17	N
					X	X				X	X	X	X	6	54.47	70.17	N
					X	X			X		X	X	X	6	60.14	70.17	N
g4	3	X	X	X	X	X								5	52.71	70.17	N
			X	X	X	X						X		5	60.81	70.17	N
			X		X	X						X	X	5	66.16	70.17	N
					X	X				X		X	X	5	60.56	70.17	N
					X	X		X		X	X		X	6	67.84	70.17	N
g5	3			X	X	X				X			X	5	37.66	48.47	N
		X		X	X	X	X						X	6	57.72	48.47	*Y*
m1	2		X	X	X	X								4	45.07	48.47	N
m2	2		X	X	X	X								4	45.07	48.47	N
m3	1		X	X	X	X								4	163.0	48.47	*Y*
m4	4		X	X										2	155.1	17.77	*Y*

Table 6-21. *Box m* Test results: Four machine types and three fault groups

The sensor variables are grouped according to origin and function. Reading from left to right, the labels for the top and bottom match networks correspond to the power, load, tune, phase and impedance. The variables listed under the heading "Clamp" refer to the clamp pressure and clamp flow, respectively.

Table 6-21 demonstrates some common covariance structure (or at least, no evidence to reject the hypothesis of a common covariance) within the same machine type and fault group, especially using the bottom (RF) match network variables. Note also that the high-

est critical value from the tables is for $p = 5$, and consequently, we use this value with m_{test} calculations involving six variables as well, although the actual critical value for $p = 6$ would presumably be a higher one.

The second case we consider is to test data taken among different machine types (labeled as *1-4*, with differences in hardware and software), but within the same fault group. In this case, we take two wafers from each machine and use either groups of two, three or four machines from the same fault group. Thus, as in the previous case, we can consider each wafer to form a population, where the number of populations is $k = 4$ (two machines contributing two wafers each), $k = 6$ (three machines contributing two wafers each), or $k = 8$ (four machines contributing two wafers each). As before, we use equal sample sizes of twenty points per wafer, and calculate maximum likelihood estimates for the sample mean, $\bar{x}(j)$, covariance, S_j, and pooled estimate of the common covariance matrix, S, using Equations 5.31, 5.32, and 5.34 respectively. Applying Equation 5.37 to compute the *Box m* test statistics yields the results summarized in Table 6-22.

Machines		Bottom (RF) Match					Top (TCP)		Clamp	*Box m* test				
Fault group	Types	pw	ld	tn	ph	im	pw	ph	fl	p	k	m_{test}	crit	>
g3 g4	1, 3	X	X	X	X	X				5	4	64.85	70.17	N
					X	X		X	X	4	4	35.92	48.47	N
g2 g3 g4	1, 1, 3				X	X		X	X	4	6	54.64	74.25	N
g1 g3 g4	1, 1, 3					X		X		2	6	21.88	26.16	N
g3 g4 g5	1, 3, 3	X	X		X	X				4	6	61.20	74.25	N
					X	X	X			3	6	42.58	46.96	N
g1 g2 g3 g4	1,1,1,3				X			X		2	8	34.11	34.14	N

Table 6-22. *Box m* Test results: Across two machine types, within the same fault group (gas line grounding problems)

We include only successful trials in this case, where $m_{test} < crit$, and we can conclude that there is no evidence to reject the null hypothesis. Also, because the sensor signals "TCP tune," "TCP load," and "TCP impedance" of the top match network, as well as the

"clamp pressure" are not used in any of the successful trials, we have eliminated them from the table. Moreover, note that the values of the critical values for m_{test} depend on both p and k. We also performed trials using wafers from different types of machines in the other two fault groups corresponding to the baseline and match network problems respectively. However, in both cases, the trials were unsuccessful, and we found no evidence to support the hypothesis of equal covariance structures in these groups.

Finally, we consider the third case of testing wafers taken from different fault groups, but within the same machine type. Not surprisingly, we found no successful trials, and no evidence to support the hypothesis of equal covariance structures in these groups. However, we did conduct a few successful trials using different machines of the same type, within the same fault group. The results here are distinct from case 1, in that the wafers are taken from separate machines. In case 1, the wafers were taken from the same machine for testing. Table 6-23 summarizes the results for the trials conducted on different machines of the same type and fault group.

Machines		Bottom (RF) Match					Top (TCP)				*Box m* test				
Fault	Types	pw	ld	tn	ph	im	ld	tn	ph	im	p	k	m_{test}	crit	>
b1 b2	1, 1		X	X							2	4	83.97	17.77	*Y*
							X	X			2	4	75.84	17.77	*Y*
g1 g2	1, 1				X	X			X	X	4	4	28.60	48.47	N
g1 g2 g3	1, 1, 1				X	X			X		3	6	36.04	46.96	N
g4 g5	3, 3	X	X	X	X	X					5	4	57.79	70.17	N
				X	X	X			X		4	4	47.68	48.47	N
m1 m2	2, 2				X		X				2	4	9.35	17.77	N
					X				X		2	4	24.15	17.77	*Y*

Table 6-23. *Box m* Test results: Within two machine types, within the same fault group (gas line grounding problems)

6.4.1.2. Summary of Covariance Analysis

In this section, we test the hypothesis of equal covariance structures for data taken from (1) within the same machine type, and the same fault group, (2) among different machine types, but within the same fault group, and (3) among different fault groups, but within the

same machine type. In the first and second cases, we found that in some trials, for certain sensor variables, no evidence to reject the null hypothesis of equal covariance matrices. However, although we might conclude that wafers taken from the same machine might share a common covariance structure in some cases for a limited set of variables, we find this much less plausible once we take wafers from different machines, even if they are still diagnosed in the same fault group. Consequently, the covariance matrix cannot be used reliably as a feature to distinguish between fault groups. Moreover, we cannot assume a common covariance structure to exist in any of the three cases we examined.

6.4.2. Building Bayesian Classifiers

Building on the results of our covariance analysis, let z: p x 1 denote an observation from one of the fault populations $\pi_j = N(\theta_j, \Sigma_j), j = 1, \ldots, k$, where the parameters (θ_j, Σ_j) are unknown, and we do not assume equal covariance matrices. The observation vector, z, is actually comprised of wafer average values of the variables, based on 20 points taken within the main etch step. In this case, our variables are the tune and load capacitor positions (p=2). We consider these separately for the top (TCP) and bottom (RF) match, so that we are working in two 2-dimensional spaces. The values of the variables appear small because we subtract the sample average (listed in Table 6-24) from each signal. Load and tune capacitor positions are typically represented on a scale of 0-32000 points.

Fault Group: Variable	Baseline	Gas Line	Match
RF Load	10533	10099	10056
RF Tune	8363	8667	8589
TCP Load	18962	18987	18093
TCP Tune	24864	23485	23342

Table 6-24. Original sample averages used to de-mean qualification data

6.4.2.1. Generating Gaussian fault populations using Maximum Likelihood

Because we have three categories corresponding to the baseline, gas line, and match problems, this defines three fault populations (k=3). We form a training data set for the Bayesian classifiers by applying Equations 5.31 and 5.32 to calculate maximum likelihood

estimates, (MLE's) for the sample mean and covariance matrix, (θ_j, Σ_j), for the jth population based on the demeaned qualification data.

Using a C-program to generate two-dimensional gaussian data, and the MLEs for each fault category, we form data sets, x(j): 50 x 2, for each of the three fault groups. These data sets comprise the fault populations used in the Bayesian classifiers. Figures 6-8 and 6-9 show the distribution of these populations for the two cases corresponding to the top and bottom match networks respectively. Note that, for each fault group, even in cases where the population means are similar, the variances are noticeably different. This is also apparent in Table 6-25, which lists the MLE values used to generate the training data sets.

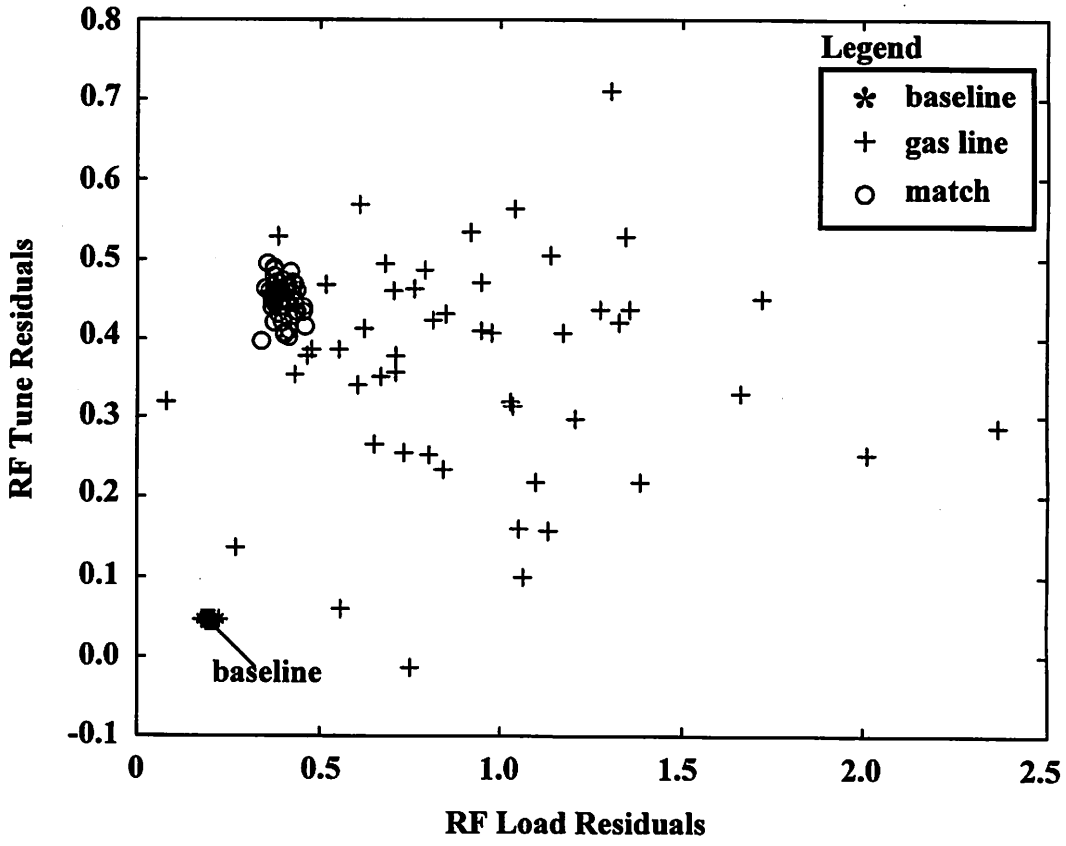


Figure 6-8. Bottom RF Match network "load" versus "tune" capacitor position residuals: distribution of fault populations

Match Location	Fault Group:	Baseline	Gas Line	Match
	Variable	$(\bar{x}_1, S_1)$	$(\bar{x}_2, S_2)$	$(\bar{x}_3, S_3)$
Bottom Match	RF Load	(0.1984, 0.0105)	(0.8749, 0.5031)	(0.4017, 0.0313)
	RF Tune	(0.0472, 0.0009)	(0.3725, 0.1289)	(0.4457, 0.0242)
Top Match	TCP Load	(1.3609, 0.2383)	(1.045, 0.3591)	(1.0068, 0.2887)
	TCP Tune	(0.3767, 0.0681)	(1.1356, 0.1286)	(1.1306, 0.5655)

Table 6-25. MLE's of demeaned sample data, $(\bar{x}(j), S_j)$, used to generate training data

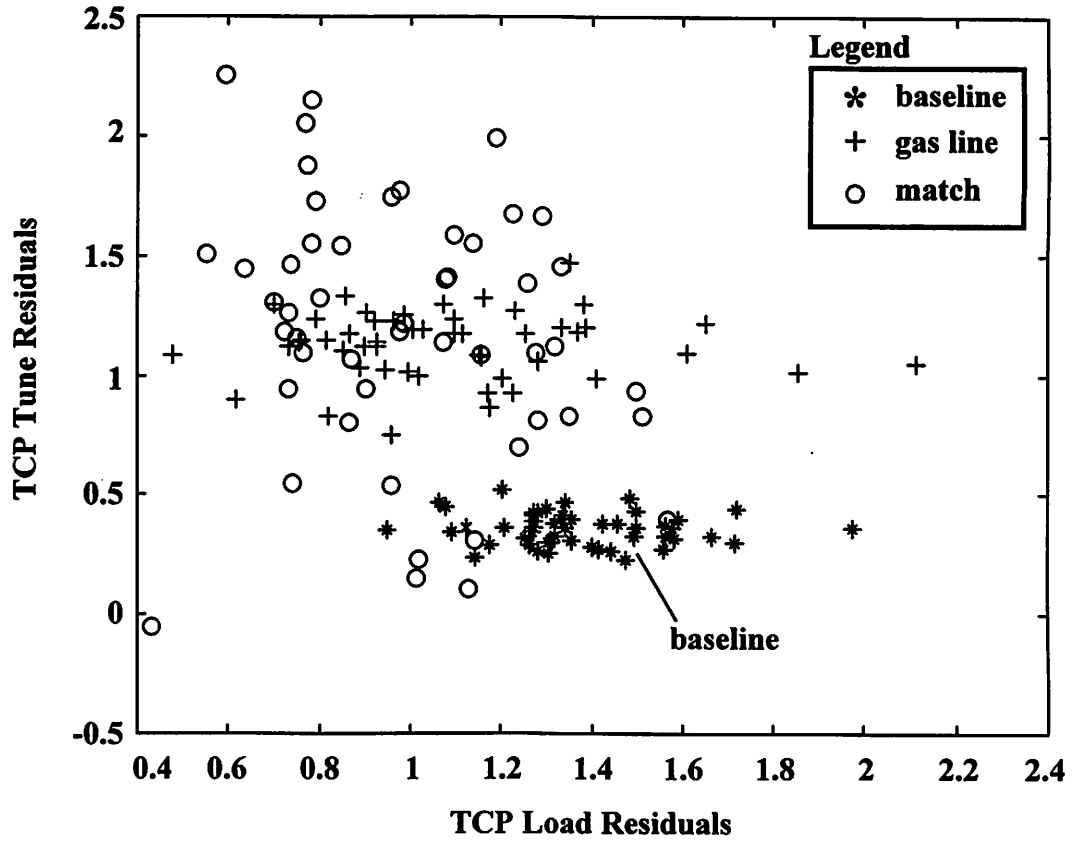


Figure 6-9. Top TCP Match network "load" versus "tune" capacitor position residuals: distribution of fault populations

6.4.2.2. Calculating Predictive Odds Ratios

Denoting the fault populations corresponding to baseline, gas line and match problems as π_1, π_2 and π_3, respectively, we now have all we need to calculate the predictive odds ratio for classifying z into π_i over π_k. Using Equation 5.47, we can make pairwise comparisons between fault groups by taking the ratio of the corresponding multivariate Student t-densities for the fault populations.

For instance, if we want to calculate the predictive odds ratio of classifying z into π_1 over π_2, that is, to find the odds that z was taken from a baseline machine versus one with a gas line problem, we use:

$$\frac{p(z/data, j=1)}{p(z/data, j=2)} = L_{12}\frac{\left[1+\frac{N_2}{N_2^2-1}(z-\bar{x}_2)'S_2^{-1}(z-\bar{x}_2)\right]^{N_2/2}}{\left[1+\frac{N_1}{N_1^2-1}(z-\bar{x}_1)'S_1^{-1}(z-\bar{x}_1)\right]^{N_1/2}} = BG2 \qquad (6.20)$$

where L_{12} is a constant given by Equation 5.48. For equal prior probabilities, $p_1 = p_2$, and equal sample sizes, $N_1 = N_2$, Equation 5.48 simplifies to:

$$L_{12} = \left(\frac{|S_2|}{|S_1|}\right)^{1/2} \qquad (6.21)$$

Note that the label "BG2" is used for the predictive odds ratio for baseline over gas line, using the top match signals, TCP tune and TCP load. These labels are listed in Table 6-7.

6.4.2.3. Validation data by Machine Type

As mentioned previously, the predictive odds ratios are based on calculations using generated data to represent the fault populations, π_j. However, the observation vectors, z, are taken from actual machines experiencing these problems. Thus, our validation set consists of qualification data summarized in Table 6-20.

6.4.2.4. Calculating Probabilities for Fault Classification

Using the predictive odds ratios calculated as described above, we compute probabilities with respect to fault groups for 228 observations (nineteen wafers processed on each of the twelve machines). The predictive odds ratio, in the form $r/1$ is easily converted to a probability by normalizing:

$$r/1 = \frac{r}{r+1} / \frac{1}{r+1} \tag{6.22}$$

Hence, if BG2 = r, in our example above, the odds of favoring the baseline over gas line are r : 1. If we want to represent this as a probability, then P(z = baseline) is $\frac{r}{r+1}$, while the P(z = gas line) is $\frac{1}{r+1}$.

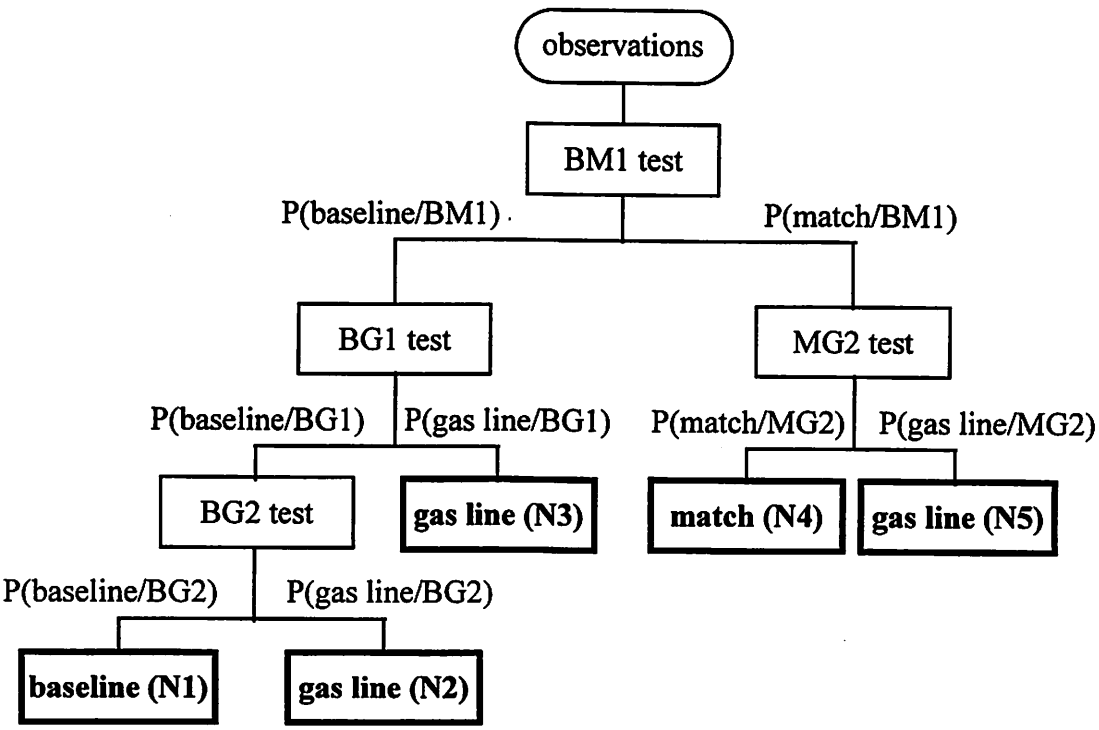


Figure 6-10. Classification tree replacing predictive odds ratios with probabilities

Returning to the tree structure of Figure 6-3, we can replace the predictive odds ratio tests with probabilities for classifying an observation into one group over another. The

probability that z belongs to a particular node is given by the product of the probabilities at the splits along the path to the node. We depict this in Figure 6-10.

Table 6-26 summarizes the calculations for the probability of the terminal nodes, representing fault diagnoses.

Diagnosis	Node	Probability
baseline	N1	P(baseline/BG2) * P(baseline/BG1) * P(baseline/BM1)
gas line	N2	P(gas line/BG2) * P(baseline/BG1) * P(baseline/BM1)
gas line	N3	P(gas line/BG1) * P(baseline/BM1)
match	N4	P(match/MG2) * P(match/BM1)
gas line	N5	P(gas line/MG2) * P(match/BM1)

Table 6-26. Probability calculations for terminal nodes in the classification tree of Figure 6-10

6.4.2.5. Results of Bayesian Classifiers

A total of 228 observations, comprised of 57 baseline (from three machines, ***b1-b3***), 95 gas line (from five machines, ***g1-g5***) and 76 match problems (from four machines, ***r1-r4***) are diagnosed using the Bayesian classifiers. After calculating the six predictive odds ratios listed in Table 6-7, the probabilities of the terminal nodes representing fault diagnoses are calculated according to the formulas in Table 6-26. Table 6-27 summarizes the classification results.

Fault Diagnosis	Node	Baseline			Gas Line Grounding					Match Network			
		b1	b2	b3	g1	g2	g3	g4	g5	m1	m2	m3	m4
baseline	N1	19	19	19			19*						
gas line	N2												
gas line	N3				18			19	19				1*
match	N4					19*				19	19	19	18
gas line	N5				1				0				

Table 6-27. Classification results by machine (total of 19 wafers for each case)

*indicates number of misclassified wafers

Note that the shaded regions in Table 6-27 represent areas of correct classification. Our results show that all observations are correctly classified, with the exception of all wafers processed on machines ***g2*** and ***g3***, and one wafer from machine ***m4***. All wafers from ***g2*** are classified as "match problems", and from ***g3*** as "baseline. Thus, although these two machines were diagnosed with gas line grounding problems, the behavior exhibited in these signals more closely resembles that of machines having match problems and normal behavior, respectively.

Diagnosed from fault:	label	baseline	gas line	gas line	match	gas line
		N1	N2	N3	N4	N5
Baseline Machine	b1	0.7299	0.0376	0.1902	0.0409	0.0013
	b2	0.7970	0.0433	0.0984	0.0411	0.0202
	b3	0.4383	0.1527	0.1890	0.0868	0.1332
Gas Line Grounding Problem	g1	0.0468	0.0169	0.5531	0.2148	0.1683
	g2	0.2701	0.0814	0.2485	0.3934	0.0065
	g3	0.5721	0.0502	0.3175	0.0576	0.0025
	g4	0.0256	0.0245	0.5438	0.0997	0.3064
	g5	0.1559	0.1899	0.3202	0.1132	0.2208
Match Network Problem	m1	0.0761	0.0144	0.2365	0.4858	0.1873
	m2	0.0218	0.0026	0.0942	0.6502	0.2312
	m3	0.0197	0.0008	0.1335	0.8069	0.0391
	m4	0.2752	0.0834	0.2752	0.3317	0.0345

Table 6-28. Average fault node probabilities (over nineteen wafers for each machine)

In general, machine to machine variability will be greater than wafer to wafer variability, where the wafers are processed by the same machine. We see this effect above, noting that the results are somewhat binary in nature, that is, either all wafers from a machine are correctly diagnosed, or all of them are misclassified. It is not surprising then, that the probabilities calculated for observations also tend to cluster around certain values depending on machine. Thus, we summarize the calculations of probabilities for each node in Table 6-28, which averages the probabilities of nineteen wafer observations for each machine. The

final diagnosis is taken as the category with the highest probability (bold border entries in Table 6-28).

6.4.2.6. Application Example

As an example, let us take "Wafer 19" processed by machine *g1*, diagnosed as a "gas line grounding" problem by qualification engineers. Using Equations 5.47 and 6.21, along with our training data sets for each fault population, we calculate the six predictive odds ratios and corresponding probabilities. These are summarized in Table 6-29.

Label	BG1	BM1	MG1	BG2	BM2	MG2
Test	base/gas	base/match	match/gas	base/gas	base/match	match/gas
ratio = r	0.1162	0.8218	0.1414	0.1770	4.1338	0.0428
prob = p	0.1041	0.4511	0.1239	0.1504	0.8052	0.0411
1 - p	0.8959	0.5489	0.8761	0.8496	0.1948	0.9589

Table 6-29. Predictive odds ratios and corresponding probabilities - Wafer 19

The probabilities, denoted by "p", represent the probability of the first variable, while "1-p" is the probability of the second variable in the ratio. Thus, for the test "BG1", the predictive odds ratio is 0.1162, the probability of the first variable is P(z = baseline) = $\frac{r}{r+1} = \frac{0.1162}{0.1162+1} = 0.1041$, and consequently the probability of the second variable is P(z = gas line) = $1 - p = 0.8959$.

Diagnosis	Node	Probability
baseline	N1	P(base/BG2) * P(base/BG1) * P(base/BM1) = 0.1504*0.1041*0.4511 = 0.0071
gas line	N2	P(gas/BG2) * P(base/BG1) * P(base/BM1) = 0.8496*0.1041*0.4511 = 0.0399
gas line	N3	P(gas/BG1) * P(base/BM1) = 0.8959*0.4511 = 0.4041
match	N4	P(match/MG2) * P(match/BM1) = 0.0411*0.5489 = 0.0226
gas line	N5	P(gas/MG2) * P(match/BM1) = 0.9589*0.5489 = 0.5263

Table 6-30. Probability calculations for terminal nodes in classification tree - Wafer 19

Finally, we calculate the final probabilities for each terminal fault node as in Table 6-26. These results are listed in Table 6-30 for "Wafer 19" processed by machine *g1*. Hence, the final probabilities for each node are:

$$P(N1,N2,N3,N4,N5) = [0.0071,0.0399,0.4041,0.0226,0.5263]$$

and we classify the observation as being in *N5*, diagnosed as a "gas line grounding" problem, with a probability of 0.5263.

6.5. Case 3: Analysis of High Speed Data

The performance of a diagnostic system based on sensor data depends to a large extent on the selection and extraction of features that reliably fingerprint a failure mode. Previous work has focused on using statistics taken from the stable portion of a plasma etch, including using average values, sample variances, and time series prediction to characterize the signal behavior. However, it has become increasingly clear that in order to identify the signature of a machine fault, attention must be focused on more subtle characteristics of the signal, not necessarily captured by taking average values over stable portions of the etch. For the analysis of the transient behavior of the high speed sensor data, it is necessary to extract features from the signals that will then form the basis for classification.

Figure 6-11 displays the impedance signal resulting from the nine fault conditions listed in Table 3-3. Note that the impedance signals for the first five fault categories are shown in Figure 3-6, along with the corresponding tune/load positions. However, for completeness, we include these signals here as well.

Looking at the signals in Figure 6-11, we observe the presence of patterns in the profile of the impedance signal over time, depending on the "fault" conditions determined by the preset values for the load and tune capacitors. Hence, the features we identify are structural, and if we can capture and link these to a fault condition, they can be considered as pieces of evidence whose combination can lead to a specific diagnosis. More importantly, we intend to use these preset conditions as a training source, a "baseline" for comparison against real machine failures involving the binding of the tune and load capacitors.

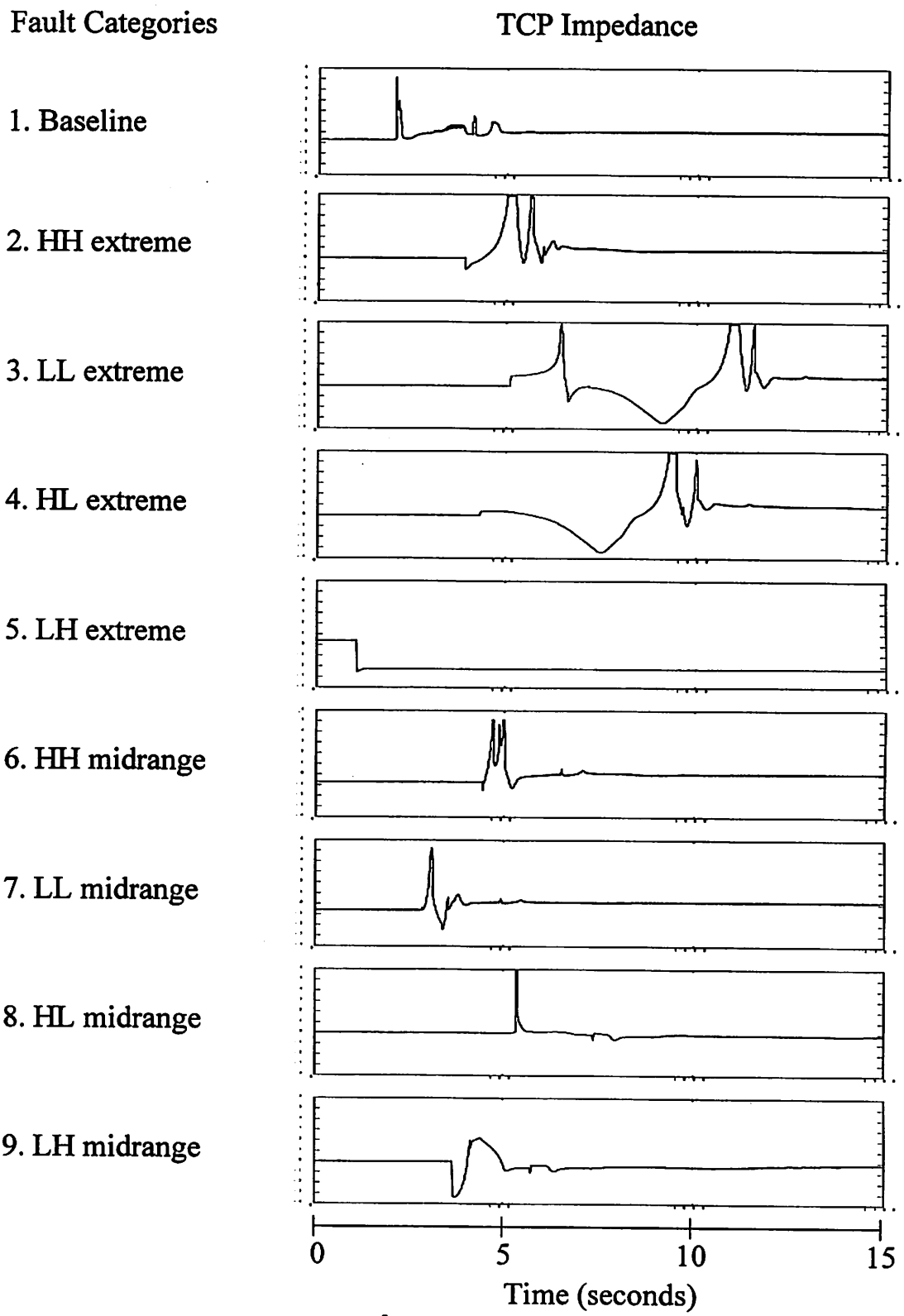


Figure 6-11. Impedance signal[5] corresponding to fault categories in Table 3-3

[5] Signals are plotted on identical scales.

6.5.1. Pattern Templates for Matched Filters

The decision-theoretic approach relies on numerical features and statistical classification techniques such as clustering. The success of statistical classification techniques is highly dependent on the feature set extracted from the data. Because the transient behavior displays a pattern, it is necessary to have a method of identifying the pattern, and having some quantitative measure of how well it matches a representative "template". One approach is to use each example where the pattern occurs as a "template" or model, and to see how closely other samples match the given template.

First, the template is determined from the data by using a windowing function to isolate the pattern as shown in Figure 6-12. This particular pattern appears to be present in three distinct impedance signals corresponding to fault conditions 2 through 4, as listed in Table 3-3, and pictured in Figure 6-11. The pattern in the window is "flipped" in time, as in Figure 6-13, and this is used as a template, acting as a matched filter for all of the other data sets. By taking the convolution of this template with the signal, the result is a measure of goodness of fit (using appropriate normalization factors). In this way, we can quantify how well the pattern in a signal matches a given template, and where it is located with respect to some reference point. We take the reference point to be the onset of RF power in the bottom match, and the quantity measuring goodness of fit to be the maximum value of the convolution (normalized).

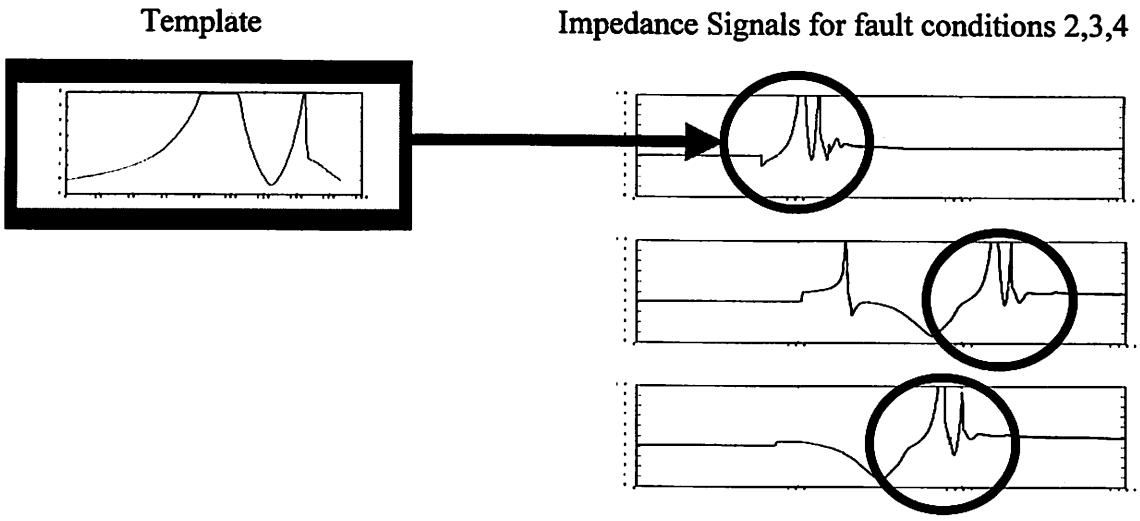


Figure 6-12. Windowing function for a pattern

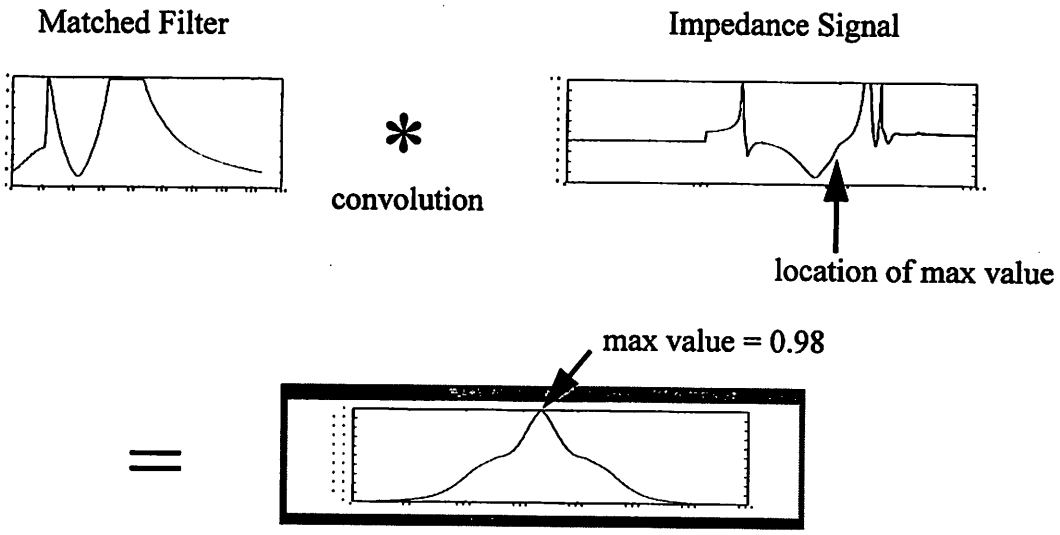


Figure 6-13. Forming a template for the matched filter

Figure 6-14 demonstrates how we measure the position or location of the pattern with respect to the onset of the bottom RF power. We emphasize that the positions of the tune and load capacitors that we are altering belong to the top (TCP) match network, and that the bottom RF power turns on after a fixed delay of 1.5 seconds following the top (TCP) power. We also establish the convention that the position takes a positive (+) value if the location of the pattern occurs before the onset of bottom RF power, and a negative (-) value if it occurs afterwards. Hence, the zero value for position corresponds to the moment when the bottom RF power is turned on.

Preliminary examination of the distribution of the features - the maximum normalized convolution and position of the pattern (shown in Figures 6-15 and 6-16 for the first five fault categories using the pattern in Figure 6-12), shows that the combination of both enables us to adequately distinguish one fault category from another. In other words, even if one pattern is found to be present in several cases, as in the example depicted by Figure 6-12, the position of the pattern varies by fault group, and hence, this becomes a crucial identifying feature. Thus, while the convolution values suggest a profile match for conditions 2, 3, and 4, we can still distinguish amongst categories by looking at the position of the pattern. Finally, note that in Figure 6-16, the position is measured according to the convention described by Figure 6-14, where 100 units is the equivalent of one second.

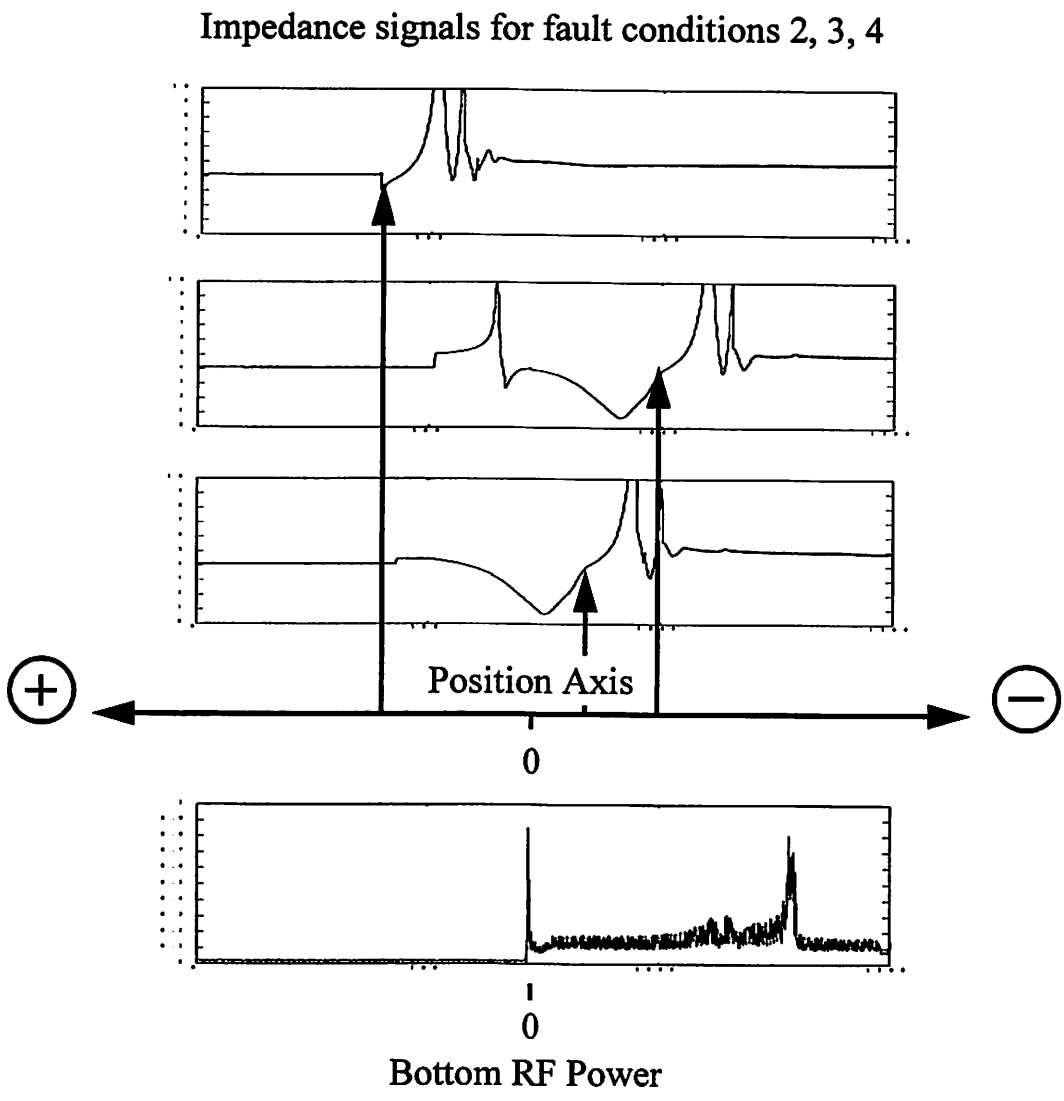


Figure 6-14. Location of pattern defined with respect to onset of (bottom match) power

As stated previously, the test pattern in the example of Figure 6-12 is found to be present in fault categories 2, 3, and 4. However, Figures 6-15 and 6-16 show that the convolution for categories 1 and 5, where the test pattern is absent, results in a fairly large number. This is because there will always be some overlap when testing a pattern against an observation. Consequently, a close profile match must be determined via a strict standard. In other words, we will consider the pattern to match the observation only for relatively high values of the convolution. The details of how we will implement this will be clearer as we develop a procedure for diagnosis.

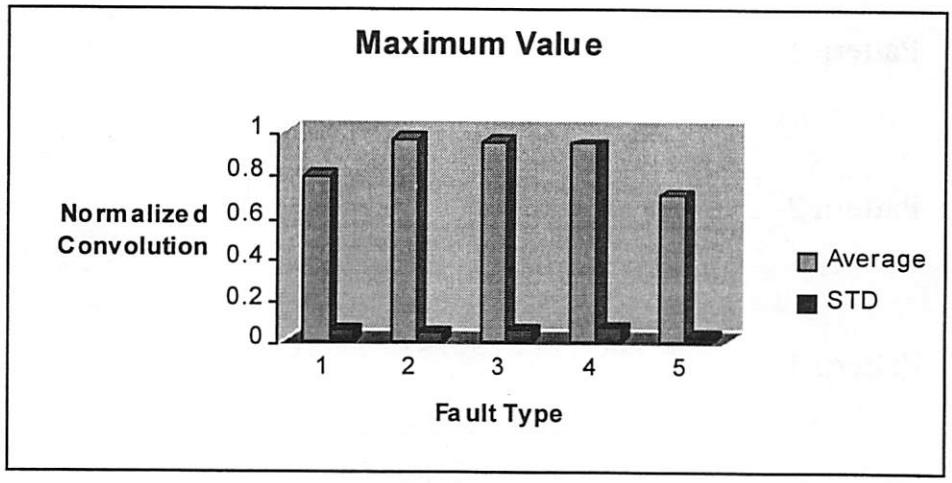


Figure 6-15. Distribution of numerical features by fault type - Normalized Convolution

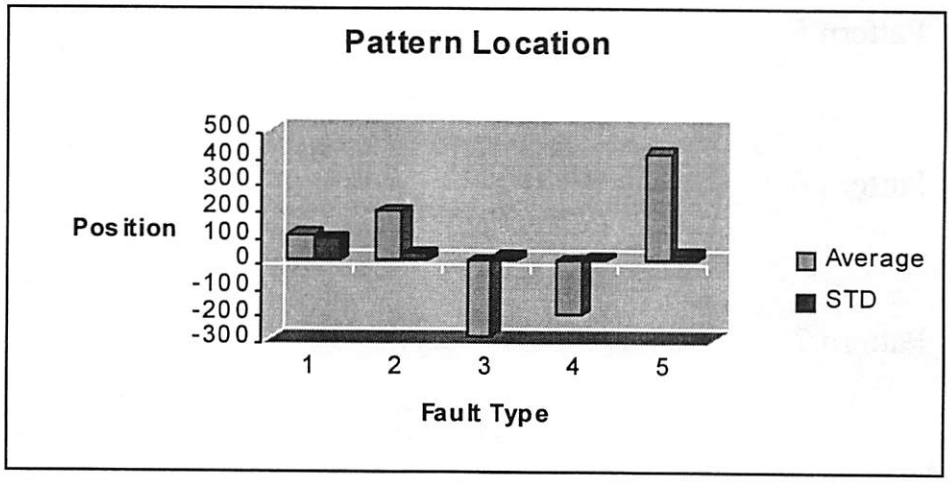


Figure 6-16. Distribution of numerical features by fault type - Pattern Location

From the impedance signals depicted in Figure 6-11, we extract nine distinct test patterns. The profiles of these patterns are shown in Figure 6-17.

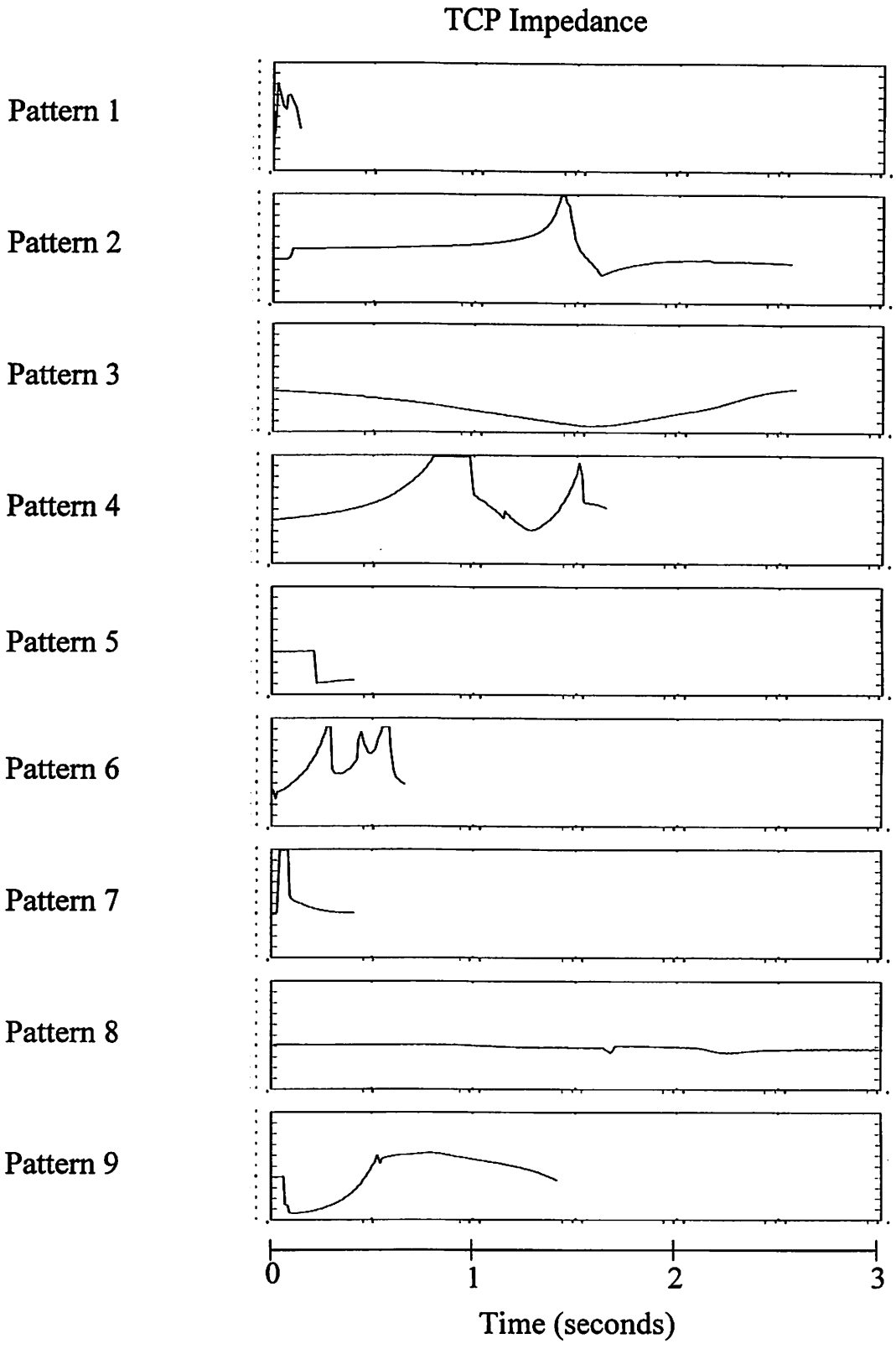


Figure 6-17. Test patterns/features extracted from the TCP Impedance signal [6]

[6] Signals are plotted on identical scales.

6.5.2. Probability Assessments for Determining Goodness of Fit

Using the matching filter approach described in the previous section, we test these patterns against our database of signals corresponding to the nine fault categories listed in Table 3-3. Consequently, we obtain values for the maximum normalized convolution and position of the patterns, and compute averages over fault categories for each test pattern as we did in the example described by Figures 6-15 and 6-16. The distribution of values obtained for the features (convolution and position) over different fault groups forms the basis of a training set. In addition, we create rules to convert these values into some measure of likelihood of how well the observation matches the profile shape and position. This procedure employs the Bayesian approach to assess the probability of an event based on a degree of belief. The process of measuring a degree of belief, for instance, using a probability wheel [57], is referred to as a probability assessment. In this case, the assessment is based on actual measurements; however, the conversion (mapping) of these measurements to probability values relies on human judgment, based on experience.

Table 6-31 summarizes the conversion rules used to map the values of the features (convolution and position) into probabilities of goodness of fit.

Convolution		Position of Pattern	
Measured Value Range	Probability	Measured Value Range	Probability
greater than 0.98	1	average position $\pm\sigma_p$	1
0.9 to 0.98	0.8	average position $\pm 2\sigma_p$	0.8
0.7 to 0.9	0.6	average position $\pm 3\sigma_p$	0.6
less than 0.7	0	else	0

Table 6-31. Conversion mapping for assessing probabilities of goodness of fit from measured values

The value of the maximum normalized convolution gives a measure of how well the observation fits the test pattern. Specifically, we wish to infer the presence or absence of the test pattern in the observation. As shown in Table 6-31, we determine that a value for the maximum normalized convolution that is greater than 0.98 is equivalent to a perfect fit (probability = 1), while any value less than 0.7 is considered not a fit (probability = 0). A "good" fit is assigned a probability assessment of 0.8, while a "fair" fit takes a probability

value of 0.6. Note that, because there will always be some amount of overlap for any test pattern against an observation signal, in general, the convolution values will be high. Thus, using the normalized convolution directly as a probability of goodness of fit will result in artificially elevated values. Consequently, the probability assessments must be skewed accordingly.

Similarly, we desire a measure of how well the test pattern identified in a given observation matches the position of the pattern as determined by our training data. In this case, we use the average values calculated for the positions of the patterns in each fault group. We say that if the pattern in the observation is found to lie within one standard deviation of this average, it is a perfect fit (probability = 1), while if it falls outside three standard deviations, there is no fit (probability = 0). A "good" fit encompasses the range of two standard deviations around the average; a "fair" fit lies within three standard deviations.

Once we have ascertained the threshold values for the measurements that determine how well the test pattern fits an observation (in terms of its shape and location), we use the mapping defined in Table 6.31 to transform the values for features computed from our training set database. Using the calculated probabilities for the training data, listed in Appendix G, we have information to determine the presence or absence of a given feature, as well as its location relative to a reference point. Moreover, we consider evidence of matching a pattern's shape (via convolution) and matching the pattern's position as independent pieces. Thus, to find the probability of matching both the shape and position of a pattern to a given observation, we take the product of the two individual probabilities. Figure 6-18 represents the analysis in the form of a flowchart. In addition, Table 6-32 summarizes the classification results for the training data in Appendix G, and shows the features found to be linked to each fault category. Note that three different patterns are found in the fault condition, "LL extreme", and that two patterns are detected in "HL extreme". Because the presence or absence of each feature is treated as an independent piece of evidence, the probability of the combination (indicating the presence of more than one feature) is given simply by the product of the individual probabilities of each feature. The next step is to determine whether this procedure can be useful in identifying real machine problems caused by the binding of the tune and load capacitors in the top match network.

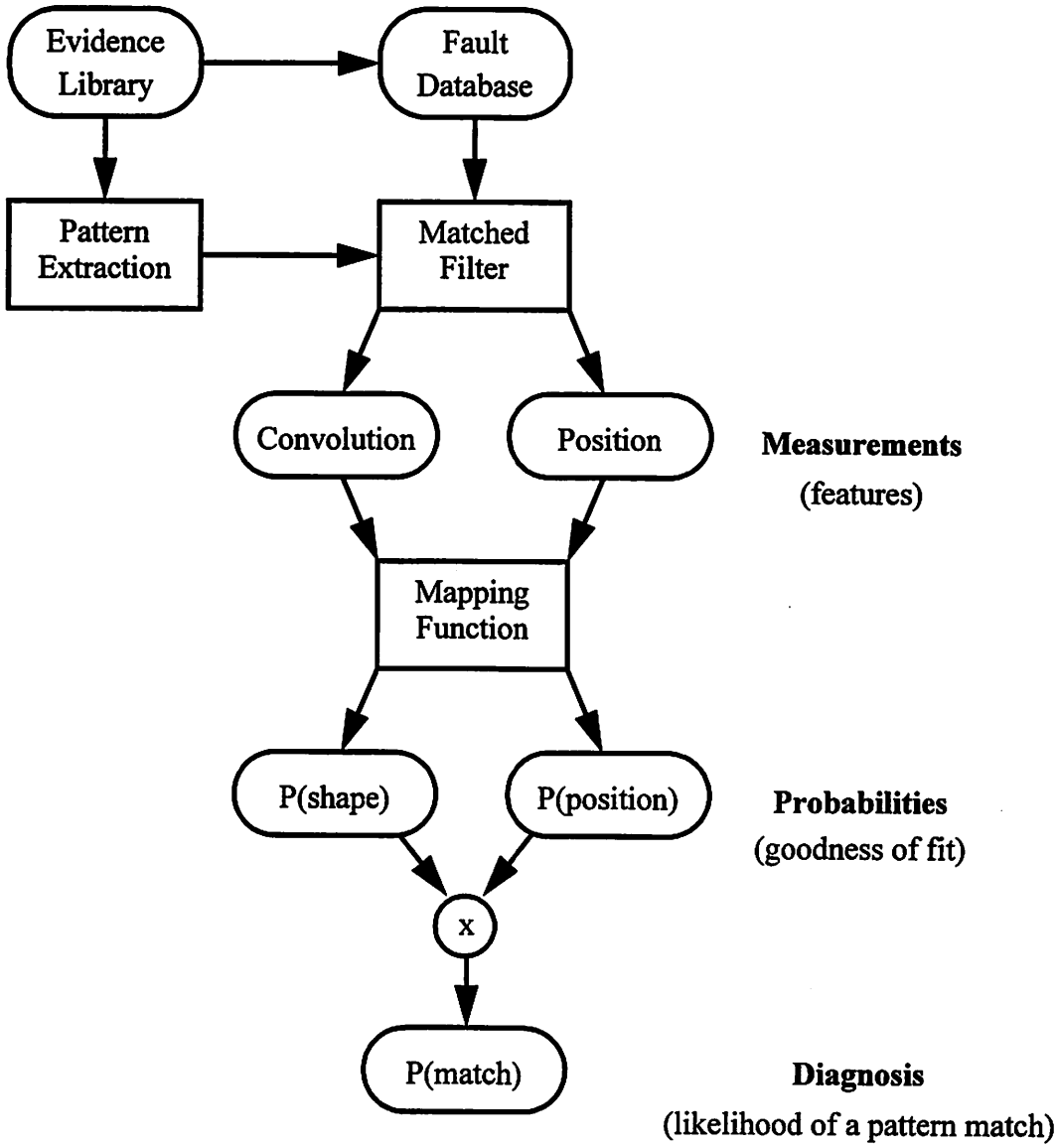


Figure 6-18. Flowchart outlining steps toward final diagnosis

Table 6-32 includes two columns for pattern 3, and three for pattern 4, corresponding to the fact that the same pattern is found in several different positions depending on the fault category. Moreover, the table lists the average probability of matching each pattern for the nine fault groups. In addition, we also include the percent of observations from each fault category successfully linked to a pattern (where the probability of a match is greater than 0.5) in Appendix G.

Fault Category		Test Patterns											
#	Description	1	2	3-1	3-2	4-1	4-2	4-3	5	6	7	8	9
1	baseline	.42					.21		.64				.32
2	HH extreme					.48			.8				
3	LL extreme		1	.8			.9	.24		.54			
4	HL extreme				.87		.62	.74		.54	.36	.48	
5	LH extreme								1				
6	HH midrange		.6						.48	1	.36	.21	
7	LL midrange					.53			1	.24			.37
8	HL midrange	.42									.93	1	
9	LH midrange					.47			1	.36			.74

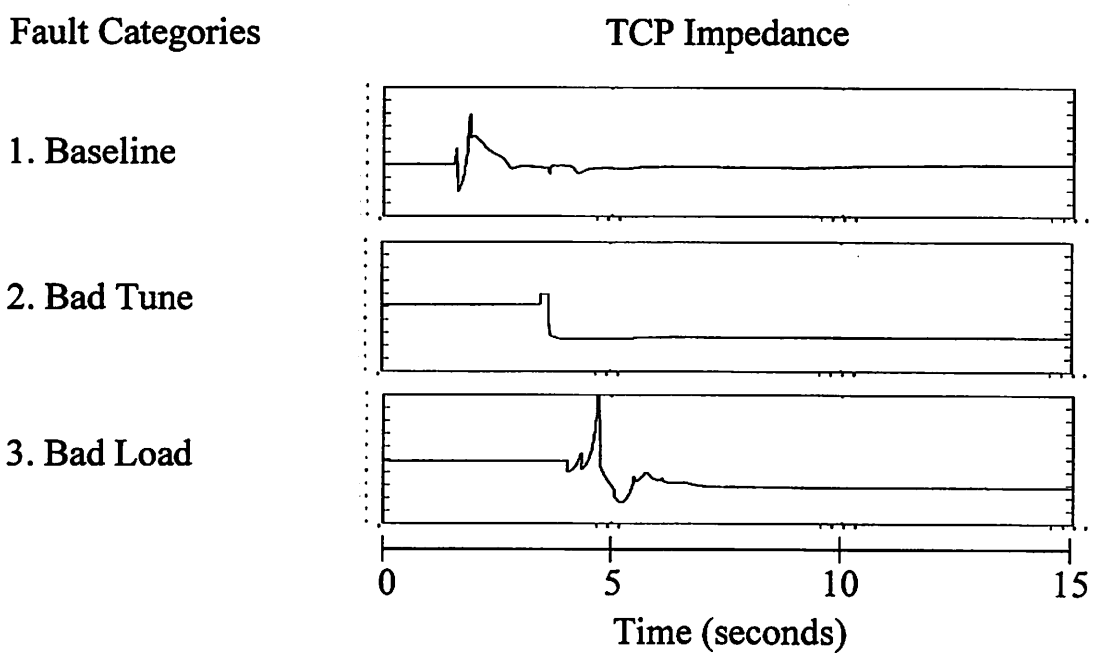
Table 6-32. Average probability of linking a pattern to a fault category

6.5.3. Diagnosing Faulty Capacitors in the Match Network

To test the utility of our diagnostic system trained using data taken from preset conditions, we physically disable the tune and load capacitors one at a time, and measure the resulting TCP impedance signals. By loosening the connection between the capacitors and their driving motors, we immobilize them, simulating a "binding" condition that commonly occurs in production when the capacitors require replacement. Figure 6-19 displays the impedance profiles produced by a baseline condition, a disabled tune capacitor, and a disabled load capacitor, respectively.

Using the patterns extracted from our training data (stored in the evidence library), and the mapping defined by Table 6-31, we test the failure data (stored in the fault base) for the known patterns, according to the procedure described by Figure 6-18. The results of this analysis are tabulated in Appendix G.

Table 5 in Appendix G shows that we are able to classify our baseline examples by testing for pattern 7, and that we find that this pattern matches the four baseline observations with probabilities {0.64, 0.59, 0.64, 0.8}, respectively. Moreover, pattern 8 matches three out of four baseline examples, with probabilities of {0.36, 0.36, 0, 0.6}.



Pattern 3-2 proves useful in identifying the signals resulting from a bad load capacitor, yielding probabilities of fit {0.48, 0.8, 0.8} for the three bad load examples, although in this case, one of the baseline observations also fits this pattern with a probability of 0.64. Finally, the bad tune signals are linked to pattern 5, with probabilities of fit {0.6, 0.6, 0.6} for the three observations taken with an immobile tune capacitor.

Thus, by using the preset conditions to simulate faults, we are able to set up a diagnostic procedure that proves useful in identifying real machine failures where the tune and load capacitors are unable to adjust to changing plasma conditions. This is accomplished by identifying, capturing and matching patterns in the profile of the transient behavior in the TCP impedance signals. In particular, the results of pattern matching show that a faulty load capacitor exhibits behavior similar to "HL extreme", or in other words, to a condition where the tune capacitor is high while the load is low. In contrast, a faulty tune capacitor is found to behave like the condition "HH extreme", where both tune and load capacitors are too high. These inferences are made due to the patterns that these different fault conditions are found to have in common.

6.6. Summary

In this chapter, we implement various modeling techniques and approaches in order to successfully classify failure data arising from different sources. Moreover, we introduce a framework that allows us to integrate the models by assuming that a fault condition causes a combination of different symptoms embodied by pieces of evidence. We consider three cases: (1) miscalibrations in the equipment simulated through DOEs, (2) machine qualification data, and (3) high speed data capturing transient behavior.

In the first case, the fault hypotheses are defined as incorrect input settings, and we find that by using a subset of the monitored sensor signals, tree-based modeling techniques can be combined with GLMs for prediction of failure modes corresponding to the changes in operating conditions. Although the models directly estimate the probabilities of each value for every fault variable, we can also obtain an estimate of these probabilities for a particular fault variable based on combinations of predictions for the remaining fault variables. Our final diagnosis of fault conditions is a result of model averaging over the different techniques, tree-based and GLMs, as well as over the direct predictions and those resulting from combinations of the remaining fault variables. Our system achieves a high success rate of fault classification for DOEs conducted on two different types of plasma etchers.

In the second case, three fault categories are identified in the qualification data: (1) the baseline, representing normal operating conditions, (2) problems connected with gas line grounding issues, and (3) problems related to the match networks. Moreover, four types of machines are identified, due to hardware and software differences, complicating the analysis of the signals. We first test the hypothesis of equal covariance structures for data taken from (1) within the same machine type, and the same fault group, (2) among different machine types, but within the same fault group, and (3) among different fault groups, but within the same machine type. Although in few cases we find no evidence to reject the null hypothesis of equal covariance matrices, we also find high machine to machine variability and hence, cannot assume a common covariance structure to exist in any of the three cases we examined. This analysis, which leads to the assumption of unequal covariance matrices, also guides us towards using a Bayesian approach, rather than sampling theory to solve this problem. Specifically, we find that predictive odds ratios extracted from Bayesian classi-

fiers can be used as splitting conditions in a tree-like classification structure. We conduct pairwise comparisons to test one fault condition over another, and are able to successfully classify all observations with the exception of two cases from machines with gas line grounding problems.

Finally, the objective in the third case is to identify cues relating to predictions of RF match problems, and conditions where the plasma will not ignite. We pay particular attention to the load and tune positions as key variables to monitor, and note the change in the profile of the measured impedance. By designing an experiment that varies the preset tune and load positions, we identify, capture, and test for structural patterns occurring in the resulting transient behavior in the impedance signal. We find that, not only are we able to link these patterns to the preset conditions, but we can also find them present in real failures where we have immobilized the tune and load capacitors to simulate a binding condition. Hence, this procedure provides a method of diagnosing the problem of a faulty tune or load capacitor that may require immediate attention or replacement.

7 Conclusions and Future Work

7.1. Thesis Summary

The utility of monitoring and process control in semiconductor manufacturing will be fully realized only if, upon the detection of a fault, relevant inferences can be drawn as to the current state of the machine. Such a system promises to be invaluable to the operator, especially as a trouble-shooting tool to find problems early, thus preventing the propagation of faults and further damage to the machine. The problem can then be resolved before it ever affects the final product.

This thesis presented the development of a decision support tool to enhance a human operator's ability to effectively monitor and diagnose problematic behavior in the course of operating a critical semiconductor manufacturing process.

First, we extended the scope and power of fault detection and monitoring procedures for the plasma etch process through the study and analysis of models to account for long term trends. Specifically, trends that are only visible over several lots in marathon runs were characterized through data transformations and linear modeling techniques. By filtering the known effects of machine aging, these models facilitate the integration of optical emission data with other sensor signals, resulting in a fault detection system that is robust over time. Moreover, the long term models are consistent with physical equations describing the window clouding effect on the measured data. Repeatability of these results over several preventative maintenance (PM) cycles suggested that a simple linear adaptive model may be used to effectively predict the behavior of a cycle, even after a change of the machine state as drastic as that produced by a PM event. Hence, the construction of new models would not be required every time the chamber or window is cleaned.

We confronted the next task of fault diagnosis by utilizing a toolbox of different modeling techniques and methods of dealing with uncertainty to exploit the characteristics of the different datasets. Our classification framework was based on the assumption that a fault will cause a combination of evidence represented by features extracted from the data. In particular, we focused on three types of data acquired from various sources, where our objectives are distinctly different in each case.

In the first case, our models were built using data from designed experiments meant to simulate a change in operating conditions. Using model averaging techniques, we combined the predictions yielded by tree-based models and GLMs. In addition, we also incorporated direct estimates of the probabilities of each value for every fault variable, with estimates based on combinations of predictions for the remaining fault variables. This procedure greatly enhanced the performance of our diagnostic system over the use of any stand-alone model. In particular, we successfully classified changing input conditions using validation sets collected from two types of plasma etch equipment.

In contrast, our objective in the second case was to classify observations into one of three states found to exist in machine qualification data: (1) the baseline, representing normal operating conditions, (2) problems connected with gas line grounding issues, and (3) problems related to the match networks. We found that, because the data were collected from four different machine types, it was unclear what we could assume in terms of characterizing the signals across machines and fault groups. An analysis of covariance suggested the use of a Bayesian approach, as opposed to sampling theory under the assumption of unequal covariance matrices for different fault populations. Predictive odds ratios, extracted from Bayesian classifiers, proved to be powerful discriminators in pairwise comparisons to test for one fault condition over another. The use of these ratios as splitting conditions in a tree-like classification structure rendered a successful diagnosis of all observations in the fault base, with the exception of two cases from machines with gas line grounding problems.

Identification and isolation of cues relating to predictions of RF match problems, and conditions where the plasma will not ignite was the focus of the third case, using data col-

lected at an increased sample rate resulting in a higher signal resolution. In particular, we observed the change in the profile of the measured impedance in response to changing load and tune capacitor positions. Structural patterns in the profile of the transient were identified, captured, and tested against observations collected while varying preset load and tune positions. In our analysis, we linked these patterns to the preset conditions by devising a method to quantify how well the pattern matched a given observation signal. In addition, we used a mapping function to assign a probability of matching a feature in an observation. This provided a test bed for classifying real machine problems resulting from tune and load capacitors that are "bound" and unable to adjust to the changing impedance. Hence, our system is capable of successfully diagnosing the problem of a faulty tune or load capacitor, possibly requiring immediate attention or replacement.

7.2. Future Directions

The focus of this work has been to use sensor signals as a source of information to infer the machine state. One possible extension would be to predict how these changes are reflected in the final wafer product. Parameters of interest that are used to measure the wafer state include etch rate, uniformity and selectivity. Here again, the main difficulty lies in obtaining access to complete datasets, including well documented examples of specific machine problems detected and diagnosed from sensor signals, along with the final wafer measurements (assuming a production environment).

Our study utilizes a decision-theoretic approach in an empirical analysis of sensor signals based on assumptions of their time-dependent behavior and statistical distribution. A disadvantage of this approach is that we work solely with classification models, grouping observations into categories based on their respective traits. A more powerful model would be a generative one, which is able to produce the output we expect under certain conditions. For instance, in our analysis of the profile of the transient in the impedance signal, we attempt to find patterns and match them to our observations. Based on experience (learned from our training dataset) we are able to characterize the behavior of a faulty tune or load capacitor. However, it would be more useful if, rather than simply identifying and matching patterns, we could actually generate them. The syntactic approach offers an attractive

alternative in this case, providing both data classification and generation. Results using this approach for characterizing and classifying sensor signals from plasma etch equipment can be found in [73].

Along the same lines, focusing on the generative aspect of modeling, another future area to consider is to examine the signals in the frequency domain. Specifically, because the electrical and mechanical machine parts generate periodic signals at different frequencies, the chamber can be viewed as a filter, and one can monitor the harmonics generated at the output to infer the chamber state.

7.3. Concluding Remarks

We have presented a unified framework for data fusion that combines evidence from multiple sensors for the purpose of diagnosing machine faults, that often arise from diverse operating conditions. While the advent of multiple sensors has widened the scope of monitoring, it has simultaneously brought new areas of complexity to the manufacturing environment. It is reasonable to expect that advances in technology will only further accelerate this trend in the future. Hence, the development of paradigms to effectively manage this complexity in the form of comprehensive models becomes especially critical.

Current computerized decision support tools available to engineers operating highly complex interrelated systems are not keeping pace with factory complexity, often resulting in data overload. Much research remains to be done to develop useful metrics for tracking complexity, for tracking the yield rate, and for converting data into knowledge. The result should be easily interpreted, and facilitate quick response and immediate action. Every contribution to improve in situ process control, the integration of off-line metrology with insitu data, and management of complexity through extraction and diagnosis of data from cluster tools, is a step towards better run to run control and real time closed-loop control. It is hoped that through effective diagnosis of the machine state, we are one step closer to implementing fault-tolerant supervisory control, and in reducing the cost of ownership of state of the art, semiconductor manufacturing equipment.

Bibliography

[1] "The National Technology Roadmap for Semiconductors, Technology Needs", SIA Semiconductor Industry Association, 1997 Edition.

[2] R. R. Schaller, "Moore's Law: Past, Present, and Future", IEEE Spectrum, June 1997, p. 53.

[3] R. Degner, "The Frugal Fab: July 1996", Special Report, Electronic Business Today, Cahners Publishing, CO, 1996.

[4] P. Singer, ". Singer, "The Many Challenges of Oxide Etching", *Semiconductor International*, Cahners Publishing, CO, June 1997, pp 109-114.

[5] P. Singer, "Plasma Etch: A Matter of Fine-Tuning", *Semiconductor International*, Cahners Publishing, CO, Dec 1995, pp 65-68.

[6] P. Singer, "New Frontiers in Plasma Etching", *Semiconductor International*, Cahners Publishing, CO, July 1996, pp 152-164.

[7] "Trends, Products & Profile: Major Market Trends", Lam Research Corporation Home Page, http://www.lamrc.com.

[8] M.J. Kushner, "Advances in plasma equipment modeling", *Semiconductor International*, Cahners Publishing, CO, June 1996, pp 135-142.

[9] J.S. Ogle, "Method and Apparatus for Producing Magnetically-Coupled Planar Plasma", U.S. Patent No. 4,948,458, Aug. 14, 1990.

[10] D.C. Montgomery, *Introduction to Statistical Quality Control*, 2nd ed., John Wiley & Sons, 1991.

[11] S.F. Lee, E.D. Boskin, H.C. Liu, E. Wen, C.J. Spanos, "RTSPC: A Software Utility for Real-Time SPC and Tool Data Analysis", IEEE Transactions on Semiconductor Manufacturing, vol.8, no. 1, Feb. 1995, pp. 17-25.

[12] A. Pankratz, *Forecasting with Univariate Box-Jenkins Models*, John Wiley & Sons, 1983.

[13] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Inc.,1995.

[14] H.E. Stephanou and A.P. sage, "Perspectives on imperfect information processing," IEEE Systems, Man and Cybernetics, Vol. SMC-17, pp. 780-798, Sept./Oct. 1987.

[15] R.R. Murphy, "Dempster-Shafer theory for sensor fusion in autonomous mobile robots", IEEE Transactions on Robotics and Automation, Vol. 14, No. 2, April 1998, pp. 197-206.

[16] J. Pearl, "*Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*", San Mateo, CA: Morgan Kaufmann, 1988.

[17] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola and L.K. Saul, "An Introduction to Variational Methods for Graphical Models," in *Learning in Graphical Models*, M.I. Jordan, Ed., Kluwer Academic Publishers, 1998.

[18] E. Horvitz and M. Barry, "Display of Information for Time-Critical Decision Marking," Proceedings of Eleventh Conference on Uncertainty in Artificial Intelligence, Montreal, August 1995, pp. 296-305.

[19] F. Nadi, A.M. Agogino, and D.A. Hodges, "Use of influence diagrams and neural networks in modeling semiconductor manufacturing processes", IEEE Transactions on Semiconductor Manufacturing, Vol. 4, No. 1, February 1991, pp. 52-58.

[20] A. Rege and A.M. Agogino, "Topological Framework for Representing and Solving Probabilistic Inference Problems in Expert Systems," IEEE Systems, Man and Cybernetics, Vol. 18(3), May 1988, pp. 402-414.

[21] T. Miltonberger, D. Morgan, and G. Orr, "Multisensor object recognition from 3D models," in Proceedings of SPIE: Sensor Fusion I, 1988, pp.161-169.

[22] P.S. Chatterjee and T.L. Huntsberger, "Comparison of techniques for sensor fusion under uncertain conditions," in Proceedings of SPIE: Sensor Fusion, Vol. 1003, 1998, pp. 194-199.

[23] G.D. Hager, *Task-Directed Sensor Fusion and Planning*, Norwell, MA: Kluwer Academic Publishers, 1990.

[24] S. Leang, "A control and diagnostic system for the photolithography process sequence", PhD Thesis, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, December 1995.

[25] S. Alag, "A Bayesian decision-theoretic framework for real-time monitoring and diagnosis of complex systems: theory and application," Ph.D. Thesis, University of California, Berkeley, Department of Mechanical Engineering, 1996.

[26] M. Beckerman and E.M. Oblow, "Treatment of systematic errors in the processing of wide-angle sonar sensor data for robotics navigation," IEEE Transactions in Robotics and Automation, Vol. 6, pp. 137-145, April 1990.

[27] R.C. Luo, M. Lin and R.S. Scherp, "The issues and approaches of a robot multi-sensor integration," in Proceedings of IEEE Robotics and Automation Conference, Raleigh, NC, March 30- April 3, 1987, pp. 1941-1946.

[28] Y. Nakamura and Y. Xu, "Geometrical fusion method for multi-sensor robotic systems," in Proceedings fo the 1989 IEEE International Conference on Robotics and Automation, 1989, Vol. 1, pp. 668-673.

[29] G. Shafer and R. Logan, "Implementing Dempster's rule for hierarchical evidence", Artificial Intelligence, Vol. 33, No. 3, November 1987, pp. 271-298.

[30] G. Shafer, *A Mathematical Theory of Evidence*, Princeton, NJ: Princeton University Press, 1976.

[31] N.H. Chang, "Monitoring, maintenance and diagnosis in a computer-integrated environment for semiconductor manufacturing", PhD Thesis, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences, July 1990.

[32] J. Chao, K. Shao and L. Jang, " Uncertain information fusion using belief measure and its application to signal classification," in Proceedings of the 1996 IEEE/SICE/ RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, 1996, pp. 151-157.

[33] Z. Luo and D. Li, " Multi-source information integration in intelligent systems using the plausibility measure," in Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, NV, October 2-5, 1994, pp. 403-409.

[34] P.L. Bolger, "Shafer-Dempster reasoning with applications to multi-sensor target identification systems," IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-17, November/December 1987, pp. 968-977.

[35] J. M. Keller and G. Hobson, "Uncertainty management in a rule-based automatic target recognizer," in Proceedings of SPIE Applications of Artificial Intelligence VII, Orlando, FL, March 28-30, 1989, Vol. 1095, pp. 126-137.

[36] R.J. Safranek, S. Gottschlich, and A.C. Kak, "Evidence accumulation using binary frames of discernment for verification vision," IEEE Transactions in Robotics and Automation, Vol. 4., August 1990, pp. 405-417.

[37] H.E. Kyburg, "Bayesian and non-Bayesian evidential updating," *Artificial Intelligence*, Vol. 31, No. 3, March 1987, pp. 271-293.

[38] J.F. Lemmer, "Confidence faction, empiricism and the Dempster-Shafer theory of evidence," in *Uncertainty in Artificial Intelligence*, L.N. Kanal and J.F. Lemmer, Eds. Amsterdam, The Netherlands: Elsevier, 1986, pp. 117-125.

[39] J.T. Nutter, "Uncertainty and probability," in the 10th International Joint Conference for Artificial Intelligence, Vol.1, 1987, pp.373-379.

[40] P. Smets, "The combination of evidence in the transferrable belief model," IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1990, Vol. 12, pp. 447-458.

[41] D.M. Buede, "A target identification comparison of Bayesian and Dempster-Shafer multisensor fusion", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, Vol. 27, No. 5, September 1997, pp. 569-577.

[42] L.A. Zadeh, "A simple view of the Dempster-Shafer theory of evidence and its implication for the rule of combination," AI Magazine, 1986, pp. 85-90.

[43] M. Laviolette, J.W. Seaman, Jr., J.D. Barrett, and W.H. Woodall, " A probabilistic and statistical view of fuzzy methods" (with discussion), Technometrics, Vol. 37, No. 3, August 1995, pp 249-287.

[44] T. Terano, K. Asai, and M. Sugeno (eds.), Fuzzy Systems Theory and Its Applications, Academic Press Inc., 1992.

[45] K. Goebel, "Management of uncertainty for sensor validation, sensor fusion, and diagnosis using soft computing techniques," Ph.D. Thesis, University of California, Berkeley, 1996.

[46] B. Hussien, A. Ismael, and M. Bender, "Evidence combination using fuzzy linguistic terms in a dynamic, multisensor environment," in Proceedings of the 1994 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems, Las Vegas, NV, October 2-5, 1994, pp. 387-394.

[47] M.I. Jordan (ed.), *Learning in Graphical Models*, Kluwer Academic Press, 1998.

[48] K. Murphy, "A brief introduction to graphical models and Bayesian networks", World wide web site, URL is http://http.cs.berkeley.edu/~murphyk/Bayes/bayes.html, 1999.

[49] A.P. Dawid, "Conditional independence in statistical theory" (with discussion), Journal of the Royal Statistical Society, Series B, 41, 1979.

[50] W.L. Buntine, "Operations for learning with graphical models", Journal of Artificial Intelligence Research 2, 1994, pp. 159-225.

[51] R.D. Shachter, "Evaluating influence diagrams", Operations Research, Vol. 34, No.6, November-December 1986, pp. 871-882.

[52] A.M. Agogino, A. Rege, "IDES: Influence Diagram based Expert System," Mathematical Modeling, Vol. 8, 1987, pp. 227-233.

[53] A.M. Agogino, R. Guha, S. Russell, "Sensor Fusion Using Influence Diagrams and Reasoning by Analogy: Application to Milling Machine Monitoring and Control," Artificial Intelligence in Engineering: Diagnosis and Learning, Computational Mechanics Publications, Southampton, England, 1988, pp. 333-357.

[54] A.M. Agogino, S. Srinivas, K.M. Schneider, "Multiple Sensor Expert System for Diagnostic Reasoning, Monitoring and Control of Mechanical Systems," Mechanical Systems and Signal Processing, Vol. 2, No.1, 1988.

[55] Y. Kim, W.H. Wood III, A.M. Agogino, "Sensor systems for an on-line diagnostic expert system in fossil power plants", ICHMT 2nd International Forum on Expert Systems and Computer Simulation in Energy Engineering, March, 1992.

[56] D. Heckerman, "Bayesian networks for data mining", Data Mining and Knowledge Discovery 1, Kluwer Academic Publishers, 1997, pp. 79-119.

[57] D. Heckerman, "A tutorial on learning with Bayesian networks", Microsoft Research tech. report, MSR-TR-95-06, 1996.

[58] R. Howard and J. Matheson, "Influence diagrams," in *Readings on the Principles and Applications of Decision Analysis*, R. Howard and J. Matheson (Eds.). Strategic Decisions Group, Menlo Park, CA, Vol. II, pp. 721-762.

[59] S. Olmsted, "On representing and solving decision problems," Ph.D. Thesis, , Stanford University, Department of Engineering-Economic Systems, 1983.

[60] R. Shachter, "Probabilistic inference and influence diagrams," Operations Research, Vol. 36, 1988, pp. 589-604.

[61] J. Pearl, "Fusion, propagation, and structuring in belief networks," Artificial Intelligence, Vol. 29, 1986, pp. 241-288.

[62] S. Lauritzen and D. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," J. Royal Statistical Society B, Vol. 50, 1988, pp. 157-224.

[63] F. Jensen, S. Lauritzen, and K. Olesen, "Bayesian updating in recursive graphical models by local computations," Computational Statisticals Quarterly, Vol. 4, 1990, pp. 269-282.

[64] P. Dawid, "Applications of a general propagation algorithm for probabilistic expert systems," Statistics and Computing, Vol. 2, 1992, pp. 25-36.

[65] W.L. Buntine, "A guide to the literature on learning probabilistic networks from data", IEEE Transactions on Knowledge and Data Engineering 8, pp. 195-210.

[66] Z. Ghahramani and M.I. Jordan, "Learning from incomplete data", A.I. Memo No. 1509, C.B.C.L. Paper No.108, M.I.T. Artificial Intelligence Laboratory and the Center for Biological and Computational Learning, December 1994.

[67] R.T. Cox, "Probability, frequency and reasonable expectation," American Journal of Physics, Vol. 14, 1946, pp. 1-13.

[68] S. James Press, *Applied Multivariate Analysis*, Holt, Rinehart and Winston, Inc., 1972, pp. 369-386.

[69] G.E.P. Box, "A general distribution theory for a class of likelihood ratio criteria," Biometrika, Vol. 36, 1949, pp. 317-346.

[70] B.P. Korin, "On testing the equality of k covariance matrices," Biometrika, Vol. 56, 1969, pp. 216-218.

[71] E.S. Pearson and H.O. Hartley, *Biometrika Tables for Statisticians*, Vol. 2, Cambridge University Press, Cambridge, 1972.

[72] A.K. Gupta and J. Tang, "Distribution of likelihood ratio statistic for testing equality of covariance matrices of multivariate Gaussian models," Biometrika, Vol. 71, 1984, pp. 555-559.

[73] D.W. Zhao, "A syntactic method for analyzing plasma etching signals," Memorandum No. UCB/ERL M99/41, 1999.

[74] D.F. Morrison, *Multivariate Statistical Methods*, 3rd ed., McGraw-Hill Publishing Company, 1990, pp. 291-309.

[75] B.G. Tabachnick, L.S. Fidell, *Using Multivariate Statistics*, Harper & Row, Publishers, NY, 1983, pp. 68-72.

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

[77] W.N. Venables and B.D. Ropley, *Modern Applied Statistics with S-plus*, Springer-Verlag, NY, 1993.

Appendix A

List of Symbols

a_t the prediction error

$g(\mu)$ the logit link function for GLMs

m modification of the likelihood ratio

m_{test} the *Box m* statistic

m_1 in Dempster-Shafer theory, the basic probability mass distribution (BPMD) derived from the multivalued mapping, $\Gamma_1: E_1 \rightarrow \Theta$

n number of observations (samples)

p number of variables

s^2 sample variance - estimate of σ^2

t test statistic for the univariate t-test

w_t the differenced data

x_t the original data series

y observation

$\bar{y}$ sample average - estimate of μ

z the thickness of the deposited material

$a \wedge b$ the min of a and b

$a \vee b$ the max of a and b

A subset of X

$\underset{\sim}{A}$ fuzzy subset of X

ARIMA (p,d,q) autoregressive integrated moving average model, where p is the auto-regressive order, d is the integration order, and q is the moving average order.

C_j the combination of evidence

D_μ the deviance

D_{μ_0} the null deviance

E_j evidence label

F_i fault label

H_0 null hypothesis

H_1 alternative hypothesis

I intensity of the plasma

L_{ij} a constant used in the calculation of the predictive odds ratio

M_q a model indexed by q

$N_p(\mu,\Sigma)$ the multivariate normal distribution

P plausibility - the degree of belief not directly in contradiction of a specific element

$P(C_j)$ the probability of matching observations to the combination of evidence C_j

$P(C_j/F_i)$ the conditional probability of a combination given a fault

$P(E_r)$ probability of a particular piece of evidence

$P(E_j/F_i)$ the class conditional probability of the evidence given the fault

$P(F_i)$ the prior probability

$P(F_i/C_j)$ the relative frequency of the fault given a combination of evidence

$P(F_i/E_j)$ the posterior probability

S supportability - the degree of belief directly supporting a specific element

S_j the unbiased estimator of Σ_j

T^2 Hotelling's T^2 statistic - mutltivariate t-test

X the whole set

α the exponential decay constant - related to the absorption properties of the material

η the linear predictor for GLMs using the logit link function

θ the true parameters specifying the distribution of a population

θ_k the moving average parameters

μ the population mean

μ_0 a given value for μ

$\mu_{\tilde{A}}$ the membership function of set $\tilde{A}$ in X

μ_H GLM probability estimate of "high"

μ_L GLM probability estimate of "low"

μ_M GLM probability estimate of "medium"

ν sample size

ν_j degrees of freedom

π_j distribution of a population j, usually given by $N(\theta_j, \Sigma_j)$

σ^2 the population variance

ϕ_k the autoregressive parameters

Γ in Dempster-Shafer theory, a multivalued mapping function that maps the elements in the evidence space to the fault space

ζ no fault

Θ the frame of discernment - fault space in Dempster-Shafer theory

Σ covariance matrix

χ_A the characteristic function of set A in X

χ^2_{p-q} the Chi-squared distribution with degree of freedom (p-q)

2^n the power set - given n elements, the set of all possible sets

{0,1}the set of zero and one

[0,1]the real-number interval from zero to one

$\varnothing$ the empty set

Appendix B

J-88-E Project - Designed Experiments for Lam TCP 9600

Experiment 30

N	Trial	PR	Top	RF_{BOT}	BCl_3	Cl_2	Ratio	Flow	Purpose	Lot
1	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	A
2	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	A
3	48	20.0	450.0	150.0	79.1	90.9	1.15	170.0	CV	C
4	18	16.3	307.3	138.5	76.8	81.7	1.06	158.5	DOE Block1	A
5	64	9.0	425.0	125.0	71.1	63.9	0.90	135.0	Verification	B
6	10	10.7	307.3	138.5	68.5	72.9	1.06	141.5	DOE Block1	A
7	37	7.0	450.0	150.0	60.5	69.5	1.15	130.0	CV	C
8	8	16.3	307.3	121.5	68.5	72.9	1.06	141.5	DOE Block1	A
9	62	18.0	275.0	142.0	79.5	75.5	0.95	155.0	Verification	B
10	7	10.7	392.7	138.5	73.1	68.4	0.94	141.5	DOE Block1	A
11	45	7.0	450.0	150.0	91.9	78.1	0.85	170.0	CV	C
12	6	16.3	307.3	138.5	73.1	68.4	0.94	141.5	DOE Block1	A
13	17	16.3	392.7	121.5	76.8	81.7	1.06	158.5	DOE Block1	A
14	40	20.0	450.0	150.0	70.3	59.7	0.85	130.0	CV	C
15	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	A
16	53	9.0	275.0	142.0	64.3	70.7	1.10	135.0	Verification	A
17	15	16.3	392.7	138.5	81.9	76.6	0.94	158.5	DOE Block1	A

Table 7-1.

N	Trial	PR	Top	RF_{BOT}	BCl_3	Cl_2	Ratio	Flow	Purpose	Lot
18	13	10.7	392.7	121.5	81.9	76.6	0.94	158.5	DOE Block1	A
19	34	10.0	450.0	110.0	79.1	90.9	1.15	170.0	CV	C
20	14	10.7	307.3	138.5	81.9	76.6	0.94	158.5	DOE Block1	A
21	63	12.0	425.0	137.0	76.3	68.7	0.90	145.0	Verification	B
22	9	10.7	392.7	121.5	68.5	72.9	1.06	141.5	DOE Block1	A
23	43	18.0	450.0	110.0	60.5	69.5	1.15	130.0	CV	C
24	19	10.7	392.7	138.5	76.8	81.7	1.06	158.5	DOE Block1	A
25	4	10.7	307.3	121.5	73.1	68.4	0.94	141.5	DOE Block1	A
26	47	7.0	450.0	110.0	70.3	59.7	0.85	130.0	CV	C
27	11	16.3	392.7	138.5	68.5	72.9	1.06	141.5	DOE Block1	A
28	12	16.3	307.3	121.5	81.9	76.6	0.94	158.5	DOE Block1	A
29	35	20.0	450.0	110.0	91.9	78.1	0.85	170.0	CV	C
30	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	A
31	16	10.7	307.3	121.5	76.8	81.7	1.06	158.5	DOE Block1	A
32	61	15.0	375.0	125.0	86.8	78.2	0.90	165.0	Verification	B
33	5	16.3	392.7	121.5	73.1	68.4	0.94	141.5	DOE Block1	A
34	11	16.3	392.7	138.5	68.5	72.9	1.06	141.5	DOE Block1	A
BREAK: SWITCH LOTS										
35	1	12.0	350.0	132.0	75.0	75.0	1.00	150.0	DOE Block1	B
36	52	13.5	350.0	130.0	75.0	75.0	1.00	150.0	CV	C
37	54	15.0	375.0	132.0	80.5	84.5	1.05	165.0	Verification	A
38	24	13.5	450.0	130.0	75.0	75.0	1.00	150.0	DOE Block2	B
39	49	17.0	350.0	130.0	70.3	59.7	0.85	130.0	CV	C
40	50	13.5	350.0	110.0	85.0	85.0	1.00	170.0	CV	C
41	29	13.5	350.0	130.0	65.0	65.0	1.00	130.0	DOE Block2	B
42	55	15.0	275.0	132.0	78.6	86.4	1.10	165.0	Verification	A
43	56	18.0	425.0	125.0	70.7	74.3	1.05	145.0	Verification	B
44	28	13.5	350.0	130.0	69.8	80.2	1.15	150.0	DOE Block2	B
45	31	8.0	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block2	B
46	36	10.0	250.0	150.0	79.1	90.9	1.15	170.0	CV	C

Table 7-2.

N	Trial	PR	Top	RF_{BOT}	BCl_3	Cl_2	Ratio	Flow	Purpose	Lot
47	33	16.0	250.0	150.0	60.5	69.5	1.15	130.0	CV	C
48	30	13.5	350.0	130.0	85.0	85.0	1.00	170.0	DOE Block2	B
49	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	B
50	57	12.0	375.0	137.0	70.7	74.3	1.05	145.0	Verification	B
51	58	12.0	325.0	132.0	73.8	81.2	1.10	155.0	Verification	B
52	23	13.5	250.0	130.0	75.0	75.0	1.00	150.0	DOE Block2	B
53	26	13.5	350.0	150.0	75.0	75.0	1.00	150.0	DOE Block2	B
54	39	20.0	250.0	150.0	91.9	78.1	0.85	170.0	CV	C
55	42	7.0	250.0	150.0	70.3	59.7	0.85	130.0	CV	C
56	27	13.5	350.0	125.0	81.1	68.9	0.85	150.0	DOE Block2	B
57	32	18.0	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block2	B
58	51	13.5	250.0	130.0	69.8	80.2	1.15	150.0	CV	C
59	44	20.0	250.0	110.0	79.1	90.9	1.15	170.0	CV	C
60	25	13.5	350.0	110.0	75.0	75.0	1.00	150.0	DOE Block2	B
61	59	15.0	325.0	142.0	69.2	65.8	0.95	135.0	Verification	B
62	31	8.0	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block2	B
63	38	7.0	250.0	110.0	60.5	69.5	1.15	130.0	CV	C
64	41	15.0	250.0	110.0	70.3	59.7	0.85	130.0	CV	C
65	1	13.5	350.0	130.0	75.0	75.0	1.00	150.0	DOE Block1	B
66	46	10.0	250.0	110.0	91.9	78.1	0.85	170.0	CV	C
67	60	9.0	325.0	137.0	79.5	75.5	0.95	155.0	Verification	B
68	43	15.0	450.0	110.0	60.5	69.5	1.15	130.0	CV	C
69	43	13.0	450.0	110.0	60.5	69.5	1.15	130.0	CV	C
70	45	10.0	450.0	150.0	91.9	78.1	0.85	170.0	CV	C

Table 7-3.

Wafers= 1484

"TIGHT flow constraints

(not enough flow at higher PR OR too much at lower PR, for the CV corner points)

RF ON = 5660 min

"Had to change some of the DOE points, on the fly, to accomodate PR/flow constraints

"First half run 3/5/96, second on 3/6/96 - 2 dummies before DOE

Appendix C

S-PLUS Output for Tree-Based Models

Lam Rainbow 4400 DOE Data

```
## Prediction for Pressure:
Classification tree:
tree(formula = Pressure ~ Impedance + MeasPower + Phase + RFCoil + MFC6 +
     DCBias, data = lamstat.train)
Variables actually used in tree construction:
[1] "RFCoil"    "Impedance"
Number of terminal nodes:  5
Residual mean deviance:  1.095 = 21.89 / 20
Misclassification error rate: 0.1667 = 4 / 24
summary(press.train.tree)
1) root 24 23.5700 medium ( 0.25000 0.25000 0.5000 )
  2) RFCoil<5799.94 22 20.7500 medium ( 0.29030 0.12900 0.5806 )
    4) RFCoil<5541.59 4 6.1210 high ( 0.50000 0.37500 0.1250 ) *
    5) RFCoil>5541.59 18 16.0000 medium ( 0.21740 0.04348 0.7391 )
     10) Impedance<17014 16 12.8300 medium ( 0.05882 0.05882 0.8824 )
       20) Impedance<16530.5 8 8.4810 medium ( 0.20000 0.20000 0.6000 ) *
       21) Impedance>16530.5 8 0.0000 medium ( 0.00000 0.00000 1.0000 ) *
     11) Impedance>17014 2 0.8109 high ( 0.66670 0.00000 0.3333 ) *
  3) RFCoil>5799.94 2 0.0000 low ( 0.00000 1.00000 0.0000 ) *

## Prediction for RFpower:
Classification tree:
tree(formula = RFpower ~ Volt + DCBias + EndpointA + EndpointB, data =
     lamstat.train)
Variables actually used in tree construction:
```

```
[1] "EndpointA"
Number of terminal nodes:  3
Residual mean deviance:  0.2383 = 5.004 / 21
Misclassification error rate: 0.04167 = 1 / 24
summary(rfpow.train.tree)
1) root 24 49.150 medium ( 0.375 0.1667 0.4583 )
  2) EndpointA<9390.72 15 17.400 medium ( 0.000 0.2667 0.7333 )
    4) EndpointA<8758.09 5  5.004 low ( 0.000 0.8000 0.2000 ) *
    5) EndpointA>8758.09 10  0.000 medium ( 0.000 0.0000 1.0000 ) *
  3) EndpointA>9390.72 9  0.000 high ( 1.000 0.0000 0.0000 ) *

## Prediction for Ratio:
Classification tree:
tree(formula = Ratio ~ Impedance + RFCoil + MFC3 + DCBias + RFTune + EndpointC,
     data = lamstat.train)
Variables actually used in tree construction:
[1] "RFTune" "RFCoil"
Number of terminal nodes:  3
Residual mean deviance:  1.305 = 27.41 / 21
Misclassification error rate: 0.25 = 6 / 24
summary(ratio.train.tree)
1) root 24 49.15 medium ( 0.3750 0.1667 0.4583 )
  2) RFTune<11797.7 9 12.31 high ( 0.7778 0.1111 0.1111 ) *
  3) RFTune>11797.7 15 25.83 medium ( 0.1333 0.2000 0.6667 )
    6) RFCoil<5655.53 8  0.00 medium ( 0.0000 0.0000 1.0000 ) *
    7) RFCoil>5655.53 7 15.11 low ( 0.2857 0.4286 0.2857 ) *

## Prediction for Total:
Classification tree:
tree(formula = Total ~ HeCFlow + MeasPressure + Impedance + MFC6 + MFC3 + Phase,
     data = lamstat.train)
Variables actually used in tree construction:
[1] "MFC3"        "MeasPressure"
Number of terminal nodes:  4
Residual mean deviance:  0.5867 = 11.73 / 20
Misclassification error rate: 0.125 = 3 / 24
summary(total.train.tree)
 1) root 24 51.050 medium ( 0.2917 0.25 0.4583 )
```

```
  2) MFC3<-18497.9 6  0.000 low ( 0.0000 1.00 0.0000 ) *
  3) MFC3>-18497.9 18 24.060 medium ( 0.3889 0.00 0.6111 )
    6) MFC3<-18337.6 8  0.000 medium ( 0.0000 0.00 1.0000 ) *
    7) MFC3>-18337.6 10 12.220 high ( 0.7000 0.00 0.3000 )
     14) MeasPressure<436.625 5  6.730 high ( 0.6000 0.00 0.4000 ) *
     15) MeasPressure>436.625 5  5.004 high ( 0.8000 0.00 0.2000 ) *

Classification tree:
snip.tree(tree = total.train.tree, nodes = 7)
Variables actually used in tree construction:
[1] "MFC3"
Number of terminal nodes:  3
Residual mean deviance:  0.5818 = 12.22 / 21
Misclassification error rate: 0.125 = 3 / 24
summary(total.snip.tree)
1) root 24 51.05 medium ( 0.2917 0.25 0.4583 )
  2) MFC3<-18497.9 6  0.00 low ( 0.0000 1.00 0.0000 ) *
  3) MFC3>-18497.9 18 24.06 medium ( 0.3889 0.00 0.6111 )
    6) MFC3<-18337.6 8  0.00 medium ( 0.0000 0.00 1.0000 ) *
    7) MFC3>-18337.6 10 12.22 high ( 0.7000 0.00 0.3000 ) *

## Prediction for Gap:
Classification tree:
tree(formula = Gap ~ MeasPressure + Volt + DCBias + Impedance + Phase + RFCoil +
      RFTune + EndpointC, data = lamstat.train)
Variables actually used in tree construction:
[1] "EndpointC"    "MeasPressure"
Number of terminal nodes:  4
Residual mean deviance:  0.5373 = 10.75 / 20
Misclassification error rate: 0.08333 = 2 / 24
summary(gap.train.tree)
 1) root 24 51.050 medium ( 0.29170 0.25000 0.4583 )
   2) EndpointC<32275.1 19 31.890 medium ( 0.36840 0.05263 0.5789 )
     4) EndpointC<29231.1 7  5.742 high ( 0.85710 0.14290 0.0000 ) *
     5) EndpointC>29231.1 12  6.884 medium ( 0.08333 0.00000 0.9167 )
      10) MeasPressure<435.812 5  5.004 medium ( 0.20000 0.00000 0.8000 ) *
      11) MeasPressure>435.812 7  0.000 medium ( 0.00000 0.00000 1.0000 ) *
   3) EndpointC>32275.1 5  0.000 low ( 0.00000 1.00000 0.0000 ) *
```

Classification tree:
snip.tree(tree = gap.train.tree, nodes = 5)
Variables actually used in tree construction:
[1] "EndpointC"
Number of terminal nodes: 3
Residual mean deviance: 0.6012 = 12.63 / 21
Misclassification error rate: 0.08333 = 2 / 24
summary(gap.snip.tree)
1) root 24 51.050 medium (0.29170 0.25000 0.4583)
  2) EndpointC<32275.1 19 31.890 medium (0.36840 0.05263 0.5789)
    4) EndpointC<29231.1 7 5.742 high (0.85710 0.14290 0.0000) *
    5) EndpointC>29231.1 12 6.884 medium (0.08333 0.00000 0.9167) *
  3) EndpointC>32275.1 5 0.000 low (0.00000 1.00000 0.0000) *

Lam TCP 9600 DOE Data

Prediction for Pressure:
Classification tree:
tree(formula = Pressure ~ endA + endB + rfcoil + X564tcptun + rfimp, data =
    avg.30.train)
Variables actually used in tree construction:
[1] "rfcoil" "X564tcptun"
Number of terminal nodes: 5
Residual mean deviance: 0.5878 = 18.22 / 31
Misclassification error rate: 0.1389 = 5 / 36
summary(press.train.ftree)
> > > > node), split, n, deviance, yval, (yprob)
    * denotes terminal node
 1) root 36 76.080 medium (0.2500 0.2778 0.4722)
  2) rfcoil<8892.08 5 0.000 low (0.0000 1.0000 0.0000) *
  3) rfcoil>8892.08 31 60.930 medium (0.2903 0.1613 0.5484)
    6) X564tcptun<19504.6 18 19.070 medium (0.2222 0.0000 0.7778)
     12) X564tcptun<19145.3 6 7.638 high (0.6667 0.0000 0.3333) *
     13) X564tcptun>19145.3 12 0.000 medium (0.0000 0.0000 1.0000) *
    7) X564tcptun>19504.6 13 27.910 high (0.3846 0.3846 0.2308)
     14) rfcoil<9360.76 5 0.000 low (0.0000 1.0000 0.0000) *
     15) rfcoil>9360.76 8 10.590 high (0.6250 0.0000 0.3750) *

Prediction for Top RF Power:

Classification tree:
tree(formula = Top.Power ~ rfpow1 + rfimp + rfmatchdc + endA + tcpimp, data =
avg.30.train)
Variables actually used in tree construction:
[1] "endA" "rfmatchdc" "rfpow1"
Number of terminal nodes: 4
Residual mean deviance: 0.4065 = 13.01 / 32
Misclassification error rate: 0.08333 = 3 / 36
summary(Top.train.ftree)
> > > > node), split, n, deviance, yval, (yprob)
* denotes terminal node
1) root 36 76.810 medium (0.2222 0.3611 0.4167)
2) endA<332.281 16 15.440 low (0.0000 0.8125 0.1875)
4) rfmatchdc<986.887 5 6.730 medium (0.0000 0.4000 0.6000) *
5) rfmatchdc>986.887 11 0.000 low (0.0000 1.0000 0.0000) *
3) endA>332.281 20 26.920 medium (0.4000 0.0000 0.6000)
6) rfpow1<24.5282 9 6.279 high (0.8889 0.0000 0.1111) *
7) rfpow1>24.5282 11 0.000 medium (0.0000 0.0000 1.0000) *

Prediction for Bottom RF Power:
Classification tree:
tree(formula = RFBot.Power ~ X564tcptun + tcpimp + endA + X578tcploadcap, data
= avg.30.train)
Variables actually used in tree construction:
[1] "X564tcptun" "endA" "X578tcploadcap"
Number of terminal nodes: 5
Residual mean deviance: 1.634 = 50.64 / 31
Misclassification error rate: 0.3611 = 13 / 36
summary(RFBot.train.ftree)
> > > > node), split, n, deviance, yval, (yprob)
* denotes terminal node
1) root 36 76.99 medium (0.3056 0.2500 0.4444)
2) X564tcptun<19504.6 23 43.15 medium (0.2174 0.1739 0.6087)
4) endA<369.805 8 0.00 medium (0.0000 0.0000 1.0000) *
5) endA>369.805 15 32.56 medium (0.3333 0.2667 0.4000)

```
    10) X578tcploadcap<28037.1 6 10.41 medium ( 0.1667 0.1667 0.6667 ) *
    11) X578tcploadcap>28037.1 9 19.10 high ( 0.4444 0.3333 0.2222 ) *
  3) X564tcptun>19504.6 13 26.32 high ( 0.4615 0.3846 0.1538 )
    6) endA<212.416 5 10.55 low ( 0.2000 0.4000 0.4000 ) *
    7) endA>212.416 8 10.59 high ( 0.6250 0.3750 0.0000 ) *

## Prediction for Gas Ratio:
Classification tree:
tree(formula = Gas.Ratio ~ rfmatchdc + rfphase + endA, data = avg.30.train)
Number of terminal nodes:  4
Residual mean deviance:  0.6604 = 21.13 / 32
Misclassification error rate: 0.1389 = 5 / 36
summary(ratio.train.ftree)
> > > > > > node), split, n, deviance, yval, (yprob)
      * denotes terminal node
 1) root 36 48.72 medium ( 0.13890 0.08333 0.7778 )
   2) endA<212.416 5  6.73 high ( 0.60000 0.40000 0.0000 ) *
   3) endA>212.416 31 23.53 medium ( 0.06452 0.03226 0.9032 )
     6) rfphase<-377.023 18  0.00 medium ( 0.00000 0.00000 1.0000 ) *
     7) rfphase>-377.023 13 17.86 medium ( 0.15380 0.07692 0.7692 )
      14) rfmatchdc<986.869 5  0.00 medium ( 0.00000 0.00000 1.0000 ) *
      15) rfmatchdc>986.869 8 14.40 medium ( 0.25000 0.12500 0.6250 ) *

## Prediction for Total Gas Flow:
Classification tree:
tree(formula = TotalGasFlow ~ ChamPress + rfimp + rfmatchdc + X564tcptun +
      X578tcploadcap, data = avg.30.train)
Variables actually used in tree construction:
[1] "rfimp"         "ChamPress"     "X578tcploadcap" "rfmatchdc"
Number of terminal nodes:  7
Residual mean deviance:  1.317 = 38.2 / 29
Misclassification error rate: 0.25 = 9 / 36
summary(total.train.ftree)
> > > > node), split, n, deviance, yval, (yprob)
      * denotes terminal node
 1) root 36 75.640 medium ( 0.3889 0.1944 0.4167 )
   2) rfimp<16339 10  6.502 medium ( 0.1000 0.0000 0.9000 )
     4) ChamPress<1335.93 5  0.000 medium ( 0.0000 0.0000 1.0000 ) *
```

5) ChamPress>1335.93 5 5.004 medium (0.2000 0.0000 0.8000) *
3) rfimp>16339 26 53.990 high (0.5000 0.2692 0.2308)
6) X578tcploadcap<28020.6 16 35.030 low (0.3125 0.3750 0.3125)
12) rfimp<16452.4 6 7.638 low (0.0000 0.6667 0.3333) *
13) rfimp>16452.4 10 20.590 high (0.5000 0.2000 0.3000)
26) rfmatchdc<986.893 5 5.004 high (0.8000 0.0000 0.2000) *
27) rfmatchdc>986.893 5 10.550 low (0.2000 0.4000 0.4000) *
7) X578tcploadcap>28020.6 10 12.780 high (0.8000 0.1000 0.1000)
14) ChamPress<1543.76 5 5.004 high (0.8000 0.0000 0.2000) *
15) ChamPress>1543.76 5 5.004 high (0.8000 0.2000 0.0000) *

Appendix D

Matlab Code for Tree-Based Models

I. Lam Rainbow 4400 DOE Data:

pressuretree.m

```
%% classification tree for pressure using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
%% supply wafer average data in data, index selects wafers
%% rows in data are observations
%% save updated probabilities as observations
%% return probability of high,low,medium diagnosis for observation

function [probs,newpmat] = pressuretree(data,index,diagbase)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 1;  % code for pressure, indexes diag5 (with wafer tag)
init_1 = [0.5 0.375 0.125];
init_2 = [0.2174 0.04348 0.7391];
init_3 = [0 1 0];
```

```
%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newpmat = [newp1 newp2 newp3];

%% number of training samples
newn1 = 10;
newn2 = 7;
newn3 = 7;

thres_1 = 5541.59;
thres_2 = 5799.94;

rfcoil = 6;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,rfcoil) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,rfcoil) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagbase,input);
         else
            p = newp3;
            [newp3,newn3] = update(p,newn3,j,diagbase,input);
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3];
   probs = [probs; p];
end
```

powertree.m

```
%% classification tree for rfbottom power using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
```

```
function [probs,newpmat] = powertree(data,index,diagbase)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 4;  % code for power, indexes diag5 (with wafer tag)
init_1 = [0 0.8 0.2];
init_2 = [0 0 1];
init_3 = [1 0 0];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newpmat = [newp1 newp2 newp3];

%% number of training samples
newn1 = 5;
newn2 = 10;
newn3 = 9;

thres_1 = 8758.09;
thres_2 = 9390.72;

endA = 1;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,endA) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,endA) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagbase,input);
```

```
        else
           p = newp3;
           [newp3,newn3] = update(p,newn3,j,diagbase,input);
        end
     end
  end
  newpmat = [newpmat; newp1 newp2 newp3];
  probs = [probs; p];
end
```

ratiotree.m

```
%% classification tree for gas ratio using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation (validation data)
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = ratiotree(data,index,diagbase)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 7;  % code for ratio, indexes diag5 (with wafer tag)
init_1 = [7/9 1/9 1/9];
init_2 = [0 0 1];
init_3 = [0.2857 0.4286 0.2857];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newpmat = [newp1 newp2 newp3];

%% number of training samples
```

```
newn1 = 9;
newn2 = 8;
newn3 = 7;

thres_1 = 11797.7;
thres_2 = 5655.53;

rftune = 7;
rfcoil = 6;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,rftune) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,rfcoil) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagbase,input);
         else
            p = newp3;
            [newp3,newn3] = update(p,newn3,j,diagbase,input);
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3];
   probs = [probs; p];
end
```

totaltree.m

```
%% classification tree for total gas flow using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = totaltree(data,index,diagbase)
```

```
%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 10; % code for total, indexes diag5 (with wafer tag)
init_1 = [0 1 0];
init_2 = [0 0 1];
init_3 = [0.7 0 0.3];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newpmat = [newp1 newp2 newp3];

%% number of training samples
newn1 = 6;
newn2 = 8;
newn3 = 10;

thres_1 = -18497.9;
thres_2 = -18337.6;

MFC3 = 13;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,MFC3) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,MFC3) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagbase,input);
         else
            p = newp3;
            [newp3,newn3] = update(p,newn3,j,diagbase,input);
         end
```

```
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3];
   probs = [probs; p];
end
```

gaptree.m

```
%% classification tree for gap spacing using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = gaptree(data,index,diagbase)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 13; % code for total, indexes diag5 (with wafer tag)
init_1 = [0.8571 0.1429 0];
init_2 = [0.08333 0 0.9167];
init_3 = [0 1 0];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newpmat = [newp1 newp2 newp3];

%% number of training samples
newn1 = 7;
newn2 = 12;
```

```
newn3 = 5;

thres_1 = 29231.1;
thres_2 = 32275.1;

endC = 15;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,endC) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,endC) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagbase,input);
         else
            p = newp3;
            [newp3,newn3] = update(p,newn3,j,diagbase,input);
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3];
   probs = [probs; p];
end
```

II. Lam TCP 9600 DOE Data:

pressuretree.m

```
%% classification tree for pressure using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = pressuretree(data,index,diagbase)

%% will return probabilities- initialize:
```

```
probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 1;  % code for pressure, indexes diag5 (with wafer tag)
init_1 = [0 1 0];
init_2 = [2/3 0 1/3];
init_3 = [0 0 1];
init_4 = [0 1 0];
init_5 = [0.625 0 0.375];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newp4 = init_4;
newp5 = init_5;
newpmat = [newp1 newp2 newp3 newp4 newp5];

%% number of training samples
newn1 = 5;
newn2 = 6;
newn3 = 12;
newn4 = 5;
newn5 = 8;

thres_1 = 8892.08;
thres_2 = 19504.6;
thres_3 = 19145.3;
thres_4 = 9360.76;

rfcoil = 6;
tcptune = 12;

for i = [index]
   for j = [1:size(diagbase,1)]  % number of rows (observations)
      if diagbase(j,1) == i
         if data(j,rfcoil) < thres_1
            p = newp1;
```

```
            [newp1,newn1] = update(p,newn1,j,diagbase,input);
         elseif data(j,tcptune) < thres_2
            if data(j,tcptune) < thres_3
               p = newp2;
               [newp2,newn2] = update(p,newn2,j,diagbase,input);
            else
               p = newp3;
               [newp3,newn3] = update(p,newn3,j,diagbase,input);
            end
         elseif data(j,rfcoil) < thres_4
            p = newp4;
            [newp4,newn4] = update(p,newn4,j,diagbase,input);
         else
            p = newp5;
            [newp5,newn5] = update(p,newn5,j,diagbase,input);
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3 newp4 newp5];
   probs = [probs; p];
end
```

toptree.m

```
%% classification tree for top power using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation (validation data)
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = toptree(data,index,diagdata)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:
```

```
input = 4;  % code for top, indexes diag5 (with wafer tag)
init_1 = [0 0.4 0.6];
init_2 = [0 1 0];
init_3 = [0.8889 0 0.1111];
init_4 = [0 0 1];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newp4 = init_4;
newpmat = [newp1 newp2 newp3 newp4];

%% number of training samples
newn1 = 5;
newn2 = 11;
newn3 = 9;
newn4 = 11;

thres_1 = 332.281;
thres_2 = 986.887;
thres_3 = 24.5282;

endA = 1;
rfmatchdc = 5;
rfpow = 4;

for i = [index]
   for j = [1:size(diagdata,1)]  % number of rows (observations)
      if diagdata(j,1) == i
         if data(j,endA) < thres_1
            if data(j,rfmatchdc) < thres_2
               p = newp1;
               [newp1,newn1] = update(p,newn1,j,diagdata,input);
            else
               p = newp2;
               [newp2,newn2] = update(p,newn2,j,diagdata,input);
            end
         elseif data(j,rfpow) < thres_3
            p = newp3;
```

```
            [newp3,newn3] = update(p,newn3,j,diagdata,input);
         else
            p = newp4;
            [newp4,newn4] = update(p,newn4,j,diagdata,input)
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3 newp4];
   probs = [probs; p];
end
```

rfbottree.m

```
%% classification tree for rfbottom power using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = rfbottree(data,index,diagdata)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 7;  % code for rfbot, indexes diag5 (with wafer tag)
init_1 = [0 0 1];
init_2 = [1/6 1/6 2/3];
init_3 = [0.4444 0.3333 0.2222];
init_4 = [0.2 0.4 0.4];
init_5 = [0.625 0.375 0];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
```

```
newp4 = init_4;
newp5 = init_5;
newpmat = [newp1 newp2 newp3 newp4 newp5];

%% number of training samples
newn1 = 8;
newn2 = 6;
newn3 = 9;
newn4 = 5;
newn5 = 8;

thres_1 = 19504.6;
thres_2 = 369.805;
thres_3 = 28037.1;
thres_4 = 212.416;

endA = 1;
tcpload = 11;
tcptune = 12;


for i = [index]
   for j = [1:size(diagdata,1)]  % number of rows (observations)
      if diagdata(j,1) == i
         if data(j,tcptune) < thres_1
            if data(j,endA) < thres_2
               p = newp1;
               [newp1,newn1] = update(p,newn1,j,diagdata,input);
            elseif data(j,tcpload) < thres_3
               p = newp2;
               [newp2,newn2] = update(p,newn2,j,diagdata,input);
            else
               p = newp3;
               [newp3,newn3] = update(p,newn3,j,diagdata,input);
            end
         elseif data(j,endA) < thres_4
            p = newp4;
            [newp4,newn4] = update(p,newn4,j,diagdata,input);
         else
            p = newp5;
            [newp5,newn5] = update(p,newn5,j,diagdata,input);
```

```
        end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3 newp4 newp5];
   probs = [probs; p];
end
```

ratiotree.m

```
%% classification tree for gas ratio using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation (validation data)
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = ratiotree(data,index,diagdata)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 10; % code for ratio, indexes diag5 (with wafer tag)
init_1 = [0.6 0.4 0];
init_2 = [0 0 1];
init_3 = [0 0 1];
init_4 = [0.25 0.125 0.625];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newp4 = init_4;
newpmat = [newp1 newp2 newp3 newp4];

%% number of training samples
newn1 = 5;
```

```
newn2 = 18;
newn3 = 5;
newn4 = 8;

thres_1 = 212.416;
thres_2 = -377.023;
thres_3 = 986.869;

endA = 1;
rfmatchdc = 5;
rfphase = 8;

for i = [index]
   for j = [1:size(diagdata,1)]  % number of rows (observations)
      if diagdata(j,1) == i
         if data(j,endA) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagdata,input);
         elseif data(j,rfphase) < thres_2
            p = newp2;
            [newp2,newn2] = update(p,newn2,j,diagdata,input);
         elseif data(j,rfmatchdc) < thres_3
            p = newp3;
            [newp3,newn3] = update(p,newn3,j,diagdata,input);
         else
            p = newp4;
            [newp4,newn4] = update(p,newn4,j,diagdata,input)
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3 newp4];
   probs = [probs; p];
end
```

totaltree.m

```
%% classification tree for total gas flow using lamstation signals
%% rules are from tree-based models built using S-plus
%% function returns probabilities of high, low and medium
%% then updates probabilities by calling update.m
%% requires a diagnosis database of actual fault classification
%% index needs to correspond to wafer observation
```

```
%% supply wafer average data in data, index selects wafers

function [probs,newpmat] = totaltree(data,index,diagdata)

%% will return probabilities- initialize:

probs = [];
p = [];

%% specify initial probabilities based on training the tree models:

input = 13; % code for total, indexes diag5 (with wafer tag)
init_1 = [0.1 0. 0.9];
init_2 = [0 2/3 1/3];
init_3 = [0.8 0 0.2];
init_4 = [0.2 0.4 0.4];
init_5 = [0.8 0.1 0.1];

%% initial probabilities
newp1 = init_1;
newp2 = init_2;
newp3 = init_3;
newp4 = init_4;
newp5 = init_5;
newpmat = [newp1 newp2 newp3 newp4 newp5];

%% number of training samples
newn1 = 10;
newn2 = 6;
newn3 = 5;
newn4 = 5;
newn5 = 10;

thres_1 = 16339;
thres_2 = 28020.6;
thres_3 = 16452.4;
thres_4 = 986.893;

press = 3;
rfmatchdc = 5;
```

```
rfimp = 9;
tcpload = 11;

for i = [index]
   for j = [1:size(diagdata,1)]  % number of rows (observations)
      if diagdata(j,1) == i
         if data(j,rfimp) < thres_1
            p = newp1;
            [newp1,newn1] = update(p,newn1,j,diagdata,input);
         elseif data(j,tcpload) < thres_2
            if data(j,rfimp) < thres_3
               p = newp2;
               [newp2,newn2] = update(p,newn2,j,diagdata,input);
            elseif data(j,rfmatchdc) < thres_4
               p = newp3;
               [newp3,newn3] = update(p,newn3,j,diagdata,input);
            else
               p = newp4;
               [newp4,newn4] = update(p,newn4,j,diagdata,input);
            end
         else
            p = newp5;
            [newp5,newn5] = update(p,newn5,j,diagdata,input);
         end
      end
   end
   newpmat = [newpmat; newp1 newp2 newp3 newp4 newp5];
   probs = [probs; p];
end
```

update.m

```
%% updates probabilities of tree-based models
%% uses diagbase - a database of the actual fault
%% indices have to correspond to wafer observations in data
%% specify input in tree program - to pick the right columns

function [newp,newn] = update(p,n,index,diagdata,input)

for i = [1:length(p)]
   if diagdata(index,input+i) == 0
      newp(i) = n*p(i)/(n+1);
```

```
    else
       newp(i) = (n*p(i)+1)/(n+1);
    end
    newn = n+1;
end
```

Appendix E1

Lam Rainbow 4400 DOE Data Classification Results

- Extracted Probabilities for Evidence Variables

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
Tree Model Prediction of Pressure	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$
Tree Model Prediction of RF Power	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$
Tree Model Prediction of Gas Ratio	$E_{3,1}$	$E_{3,2}$	$E_{3,3}$
Tree Model Prediction of Total Gas Flow	$E_{4,1}$	$E_{4,2}$	$E_{4,3}$
Tree Model Prediction of Gap Spacing	$E_{5,1}$	$E_{5,2}$	$E_{5,3}$

Table 1. Evidence Labels for Tree Model Prediction of Input Responses

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
Tree-Based Prediction of Pressure	$P(F_{1,1}/M_2)$	$P(F_{1,2}/M_2)$	$P(F_{1,3}/M_2)$
Tree-Based Prediction of RF Power	$P(F_{2,1}/M_2)$	$P(F_{2,2}/M_2)$	$P(F_{2,3}/M_2)$
Tree-Based Prediction of Gas Ratio	$P(F_{3,1}/M_2)$	$P(F_{3,2}/M_2)$	$P(F_{3,3}/M_2)$
Tree-Based Prediction of Total Gas Flow	$P(F_{4,1}/M_2)$	$P(F_{4,2}/M_2)$	$P(F_{4,3}/M_2)$
Tree-Based Prediction of Gap Spacing	$P(F_{5,1}/M_2)$	$P(F_{5,2}/M_2)$	$P(F_{5,3}/M_2)$

Table 2. Probabilities for Tree Model Prediction Based on Combinations of Evidence

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
GLM Prediction of Pressure	$E_{6,1}$	$E_{6,2}$	$E_{6,3}$
GLM Prediction of RF Power	$E_{7,1}$	$E_{7,2}$	$E_{7,3}$
GLM Prediction of Gas Ratio	$E_{8,1}$	$E_{8,2}$	$E_{8,3}$
GLM Prediction of Total Gas Flow	$E_{9,1}$	$E_{9,2}$	$E_{9,3}$
GLM Prediction of Gap Spacing	$E_{10,1}$	$E_{10,2}$	$E_{10,3}$

Table 3. Evidence Labels for GLM Prediction of Input Responses

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
GLM-Based Prediction of Pressure	$P(F_{1,1}/M_4)$	$P(F_{1,2}/M_4)$	$P(F_{1,3}/M_4)$
GLM-Based Prediction of RF Power	$P(F_{2,1}/M_4)$	$P(F_{2,2}/M_4)$	$P(F_{2,3}/M_4)$
GLM-Based Prediction of Gas Ratio	$P(F_{3,1}/M_4)$	$P(F_{3,2}/M_4)$	$P(F_{3,3}/M_4)$
GLM-Based Prediction of Total Gas Flow	$P(F_{4,1}/M_4)$	$P(F_{4,2}/M_4)$	$P(F_{4,3}/M_4)$
GLM-Based Prediction of Gap Spacing	$P(F_{5,1}/M_4)$	$P(F_{5,2}/M_4)$	$P(F_{5,3}/M_4)$

Table 4. Probabilities for GLM Prediction Based on Combinations of Evidence

Model Averaged Result	High (s=1)	Medium (s=2)	Low (s=3)
Tree/GLM Prediction of Pressure	$P(F_{1,1})$	$P(F_{1,2})$	$P(F_{1,3})$
Tree/GLM Prediction of RF Power	$P(F_{2,1})$	$P(F_{2,2})$	$P(F_{2,3})$
Tree/GLM Prediction of Gas Ratio	$P(F_{3,1})$	$P(F_{3,2})$	$P(F_{3,3})$
Tree/GLM Prediction of Total Gas Flow	$P(F_{4,1})$	$P(F_{4,2})$	$P(F_{4,3})$
Tree/GLM Prediction of Gap Spacing	$P(F_{5,1})$	$P(F_{5,2})$	$P(F_{5,3})$

Table 5. Final Fault Probabilities for Combined Model Prediction of Input Responses

Wafer	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$
1	0	1.0000	0	0	0.8000	0.2000
2	0.2174	0.0435	0.7391	1.0000	0	0
3	0.1902	0.1630	0.6467	0	0.8333	0.1667
4	0.5000	0.3750	0.1250	0	0.8571	0.1429
5	0.4545	0.4318	0.1136	0	0	1.0000
6	0.1691	0.1449	0.6860	0	0	1.0000
7	0.1522	0.1304	0.7174	0	0.8750	0.1250
8	0.1383	0.1186	0.7431	0.9000	0	0.1000
9	0.2101	0.1087	0.6811	0	0	1.0000
10	0.1940	0.1003	0.7057	0	0	1.0000
11	0.1801	0.0932	0.7267	0	0	1.0000
12	0.1681	0.0870	0.7449	0.9091	0	0.0909

Table 6. Tree Model Prediction of Input Responses Pressure and RF Power

Wafer	$E_{3,1}$	$E_{3,2}$	$E_{3,3}$	$E_{4,1}$	$E_{4,2}$	$E_{4,3}$
1	0.2857	0.4286	0.2857	0.7000	0	0.3000
2	0	0	1.0000	0	1.0000	0
3	0.7778	0.1111	0.1111	0	0.8571	0.1429
4	0.7000	0.1000	0.2000	0.7273	0	0.2727
5	0.7273	0.0909	0.1818	0	0.7500	0.2500
6	0.2500	0.5000	0.2500	0	0.6667	0.3333
7	0.6667	0.0833	0.2500	0	0.6000	0.4000
8	0.2222	0.5556	0.2222	0	0.5455	0.4545
9	0.6154	0.0769	0.3077	0	0.5833	0.4167
10	0.5714	0.0714	0.3571	0	0	1.0000
11	0	0	1.0000	0	0.5385	0.4615
12	0.5333	0.0667	0.4000	0.7500	0	0.2500

Table 7. Tree Model Prediction of Input Responses Gas Ratio and Total Gas Flow

Wafer	$E_{5,1}$	$E_{5,2}$	$E_{5,3}$
1	0.8571	0.1429	0
2	0.0833	0	0.9167
3	0.8750	0.1250	0
4	0	1.0000	0
5	0	1.0000	0
6	0.0769	0	0.9231
7	0.7777	0.1111	0.1111
8	0.7000	0.1000	0.2000
9	0.0714	0	0.9286
10	0.0667	0	0.9334
11	0.0625	0	0.9375
12	0.7272	0.0909	0.1818

Table 8. Tree Model Prediction of Input Response Gap Spacing

Wafer	$P(F_{1,1}/M_2)$	$P(F_{1,2}/M_2)$	$P(F_{1,3}/M_2)$	$P(F_{2,1}/M_2)$	$P(F_{2,2}/M_2)$	$P(F_{2,3}/M_2)$
1	0.3657	0.3657	0.2686	0.3809	0.4524	0.1667
2	0.3333	0.3333	0.3333	0.1075	0.1075	0.7850
3	0.4135	0.4135	0.1731	0.3005	0.4477	0.2519
4	0.4035	0.4035	0.1931	0.2250	0.6068	0.1682
5	0.3182	0.3182	0.3636	0.6539	0.1705	0.1756
6	0.1843	0.1843	0.6313	0.2267	0.2271	0.5462
7	0.3693	0.3693	0.2613	0.2835	0.3485	0.3680
8	0.3527	0.3564	0.2909	0.3155	0.2809	0.4036
9	0.1566	0.1566	0.6869	0.1976	0.2115	0.5908
10	0.0446	0.0446	0.9108	0.0951	0.1147	0.7902
11	0.0506	0.0506	0.8989	0.0860	0.1122	0.8017
12	0.4113	0.3232	0.2655	0.3318	0.2761	0.3920

Table 9. Fault Probabilities for Pressure and RF Power extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$P(F_{3,1}/M_2)$	$P(F_{3,2}/M_2)$	$P(F_{3,3}/M_2)$	$P(F_{4,1}/M_2)$	$P(F_{4,2}/M_2)$	$P(F_{4,3}/M_2)$
1	0.2267	0.6266	0.1467	0.4857	0.3551	0.1592
2	0.3261	0.3478	0.3261	0.1075	0.1075	0.7850
3	0.4754	0.2546	0.2700	0.3007	0.4438	0.2554
4	0.6953	0.1499	0.1548	0.6821	0.1571	0.1607
5	0.3239	0.3239	0.3523	0.3264	0.3264	0.3471
6	0.1034	0.1034	0.7932	0.1081	0.1081	0.7838
7	0.3869	0.2715	0.3417	0.2976	0.3915	0.3109
8	0.2779	0.3536	0.3685	0.3148	0.3611	0.3241
9	0.0986	0.0986	0.8029	0.1048	0.1048	0.7903
10	0.0664	0.0664	0.8672	0.0969	0.0969	0.8063
11	0.0884	0.0884	0.8232	0.0676	0.0676	0.8648
12	0.3933	0.2827	0.3240	0.3587	0.2899	0.3514

Table 10. Fault Probabilities for Gas Ratio and Total Gas Flow extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$P(F_{5,1}/M_2)$	$P(F_{5,2}/M_2)$	$P(F_{5,3}/M_2)$
1	0.4343	0.3543	0.2114
2	0.3333	0.3333	0.3333
3	0.4526	0.2563	0.2910
4	0.2358	0.5631	0.2011
5	0.3057	0.3057	0.3885
6	0.2155	0.2155	0.5690
7	0.3615	0.2626	0.3758
8	0.3337	0.2917	0.3745
9	0.1923	0.1923	0.6154
10	0.0860	0.0860	0.8280
11	0.0795	0.0795	0.8409
12	0.3562	0.2750	0.3688

Table 11. Fault Probabilities for Gap Spacing extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$E_{6,1}$	$E_{6,2}$	$E_{6,3}$	$E_{7,1}$	$E_{7,2}$	$E_{7,3}$
1	0	1.0000	0.0000	1.0000	0	0.0000
2	0	1.0000	0.0000	0	1.0000	0.0000
3	0	0.4448	0.5552	0	0	1.0000
4	0	1.0000	0.0000	0	0	1.0000
5	0	1.0000	0.0000	0	0	1.0000
6	0	1.0000	0.0000	0	0	1.0000
7	0	0	1.0000	0	0	1.0000
8	0	0	1.0000	0.5000	0.5000	0
9	0	0.9947	0.0053	0	0	1.0000
10	0	0.3058	0.6942	0	0	1.0000
11	0	0	1.0000	0	0	1.0000
12	1.0000	0	0.0000	1.0000	0	0.0000

Table 12. GLM Prediction of Input Responses Pressure and RF Power

Wafer	$E_{8,1}$	$E_{8,2}$	$E_{8,3}$	$E_{9,1}$	$E_{9,2}$	$E_{9,3}$
1	0	1.0000	0.0000	0.4163	0	0.5837
2	0	0	1.0000	0.4823	0	0.5177
3	0	1.0000	0.0000	0.6846	0	0.3154
4	0	1.0000	0.0000	0.4867	0	0.5133
5	0	0	1.0000	0.6691	0	0.3309
6	0	0	1.0000	0.5102	0	0.4898
7	0	1.0000	0.0000	0.2290	0	0.7710
8	1.0000	0	0.0000	0.3572	0.1244	0.5183
9	0	0	1.0000	0.5230	0	0.4770
10	0	0	1.0000	0.4727	0	0.5273
11	0	0	1.0000	0.4471	0	0.5529
12	1.0000	0	0.0000	0.5850	0.0133	0.4016

Table 13. GLM Prediction of Input Responses Gas Ratio and Total Gas Flow

Wafer	$E_{10,1}$	$E_{10,2}$	$E_{10,3}$
1	0	0	1.0000
2	0	1.0000	0.0000
3	0	1.0000	0.0000
4	0	0	1.0000
5	0	1.0000	0.0000
6	0	1.0000	0.0000
7	0	1.0000	0.0000
8	0	1.0000	0.0000
9	0	1.0000	0.0000
10	0	0	1.0000
11	0	1.0000	0.0000
12	1.0000	0	0.0000

Table 14. GLM Prediction of Input Response Gap Spacing

Wafer	$P(F_{1,1}/M_4)$	$P(F_{1,2}/M_4)$	$P(F_{1,3}/M_4)$	$P(F_{2,1}/M_4)$	$P(F_{2,2}/M_4)$	$P(F_{2,3}/M_4)$
1	0.3333	0.3333	0.3334	0.3333	0.3333	0.3334
2	0.3333	0.3333	0.3334	0.3333	0.3333	0.3334
3	0.3333	0.3333	0.3334	0.5363	0.2318	0.2319
4	0.1622	0.1622	0.6755	0.3333	0.3333	0.3334
5	0.2230	0.2230	0.5540	0.3333	0.3333	0.3334
6	0.1701	0.1701	0.6599	0.3333	0.3333	0.3334
7	0.3333	0.3333	0.3334	0.3333	0.3333	0.3334
8	0.3631	0.3838	0.2531	0.3333	0.3333	0.3334
9	0.1743	0.1743	0.6514	0.3325	0.3325	0.3350
10	0.0479	0.0479	0.9041	0.0787	0.1092	0.8121
11	0.1490	0.1490	0.7019	0.1490	0.1490	0.7019
12	0.5283	0.3333	0.1383	0.7189	0.1472	0.1339

Table 15. Fault Probabilities for Pressure and RF Power extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{3,1}/M_4)$	$P(F_{3,2}/M_4)$	$P(F_{3,3}/M_4)$	$P(F_{4,1}/M_4)$	$P(F_{4,2}/M_4)$	$P(F_{4,3}/M_4)$
1	0.3333	0.3333	0.3334	0.3333	0.3333	0.3333
2	0.6548	0.1726	0.1726	0.3333	0.3333	0.3333
3	0.2750	0.2750	0.4501	0.3333	0.3333	0.3334
4	0.1622	0.1622	0.6755	0.3333	0.3333	0.3334
5	0.3333	0.3333	0.3334	0.3333	0.3333	0.3334
6	0.3333	0.3333	0.3334	0.3333	0.3333	0.3334
7	0.0763	0.0763	0.8473	0.3333	0.3333	0.3334
8	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
9	0.3325	0.3325	0.3350	0.3315	0.3315	0.3369
10	0.0815	0.0815	0.8371	0.0631	0.0631	0.8738
11	0.1490	0.1490	0.7019	0.0000	0.0000	1.0000
12	0.7189	0.1472	0.1339	0.9999	0.0001	0.0001

Table 16. Fault Probabilities for Gas Ratio and Total Gas Flow extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{5,1}/M_4)$	$P(F_{5,2}/M_4)$	$P(F_{5,3}/M_4)$
1	0.1946	0.6108	0.1946
2	0.3333	0.3333	0.3334
3	0.2750	0.2750	0.4501
4	0.3333	0.3333	0.3334
5	0.2230	0.2230	0.5540
6	0.1701	0.1701	0.6599
7	0.0763	0.0763	0.8473
8	0.3333	0.3333	0.3333
9	0.1736	0.1736	0.6528
10	0.0815	0.0815	0.8371
11	0.0503	0.0503	0.8995
12	0.7189	0.1472	0.1339

Table 17. Fault Probabilities for Gap Spacing extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{1,1})$	$P(F_{1,2})$	$P(F_{1,3})$	$P(F_{2,1})$	$P(F_{2,2})$	$P(F_{2,3})$
1	0	1.0000	0.0000	0.1905	0.6262	0.1833
2	0	1.0000	0.0000	0.2204	0.2204	0.5592
3	0.1784	0.2760	0.5455	0.1002	0.7048	0.1951
4	0	1.0000	0.0000	0.1125	0.7320	0.1555
5	0.2706	0.2706	0.4588	0.0833	0.0833	0.8333
6	0.1734	0.1674	0.6592	0.1400	0.1401	0.7199
7	0.1594	0.1485	0.6920	0.0945	0.6995	0.2060
8	0.2135	0.2147	0.5718	0.7000	0.2500	0.0500
9	0.1405	0.3339	0.5256	0.1325	0.1360	0.7315
10	0.0751	0.1180	0.8069	0.0434	0.0560	0.9006
11	0.0949	0.0732	0.8319	0.0588	0.0653	0.8759
12	0.5877	0.2449	0.1674	0.8842	0.0368	0.0789

Table 18. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Responses Pressure and RF Power

Wafer	$P(F_{3,1})$	$P(F_{3,2})$	$P(F_{3,3})$	$P(F_{4,1})$	$P(F_{4,2})$	$P(F_{4,3})$
1	0.1281	0.7638	0.1081	0.5929	0.1775	0.2296
2	0	0	1.0000	0.2577	0.1371	0.6053
3	0.2750	0.2750	0.4501	0.4504	0.2222	0.3274
4	0.6977	0.1249	0.1774	0.7047	0.0786	0.2167
5	0.2453	0.2453	0.5095	0.3299	0.3299	0.3403
6	0.2500	0.5000	0.2500	0.2207	0.2207	0.5586
7	0.0763	0.0763	0.8473	0.2985	0.2222	0.4793
8	0.2222	0.5556	0.2222	0.1574	0.4533	0.3893
9	0.2455	0.1277	0.6269	0.2296	0.2489	0.5214
10	0.1640	0.0547	0.7813	0.1492	0.0446	0.8062
11	0.0594	0.0594	0.8813	0	0.5385	0.4615
12	0.6614	0.1241	0.2145	0.6734	0.0758	0.2508

Table 19. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Responses Gas Ratio and Total Gas Flow

Wafer	$P(F_{5,1})$	$P(F_{5,2})$	$P(F_{5,3})$
1	0.6457	0.2486	0.1057
2	0.2083	0.1667	0.6250
3	0.3342	0.2687	0.3971
4	0.1179	0.7815	0.1006
5	0	1.0000	0.0000
6	0.1581	0.1389	0.7030
7	0.1714	0.1384	0.6901
8	0.7000	0.1000	0.2000
9	0.1131	0.3156	0.5713
10	0.0607	0.0431	0.8963
11	0.0606	0.0450	0.8944
12	0.7315	0.1130	0.1556

Table 20. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Response Gap Spacing

Appendix E2

Lam TCP 9600 DOE Data Classification Results

- Extracted Probabilities for Evidence Variables

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
Tree Model Prediction of Pressure	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$
Tree Model Prediction of Top Power	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$
Tree Model Prediction of RF Power	$E_{3,1}$	$E_{3,2}$	$E_{3,3}$
Tree Model Prediction of Gas Ratio	$E_{4,1}$	$E_{4,2}$	$E_{4,3}$
Tree Model Prediction of Total Gas Flow	$E_{5,1}$	$E_{5,2}$	$E_{5,3}$

Table 1. Evidence Labels for Tree Model Prediction of Input Responses

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
Tree-Based Prediction of Pressure	$P(F_{1,1}/M_2)$	$P(F_{1,2}/M_2)$	$P(F_{1,3}/M_2)$
Tree-Based Prediction of Top Power	$P(F_{2,1}/M_2)$	$P(F_{2,2}/M_2)$	$P(F_{2,3}/M_2)$
Tree-Based Prediction of RF Power	$P(F_{3,1}/M_2)$	$P(F_{3,2}/M_2)$	$P(F_{3,3}/M_2)$
Tree-Based Prediction of Gas Ratio	$P(F_{4,1}/M_2)$	$P(F_{4,2}/M_2)$	$P(F_{4,3}/M_2)$
Tree-Based Prediction of Total Gas Flow	$P(F_{5,1}/M_2)$	$P(F_{5,2}/M_2)$	$P(F_{5,3}/M_2)$

Table 2. Probabilities for Tree Model Prediction Based on Combinations of Evidence

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
GLM Prediction of Pressure	$E_{6,1}$	$E_{6,2}$	$E_{6,3}$
GLM Prediction of Top Power	$E_{7,1}$	$E_{7,2}$	$E_{7,3}$
GLM Prediction of RF Power	$E_{8,1}$	$E_{8,2}$	$E_{8,3}$
GLM Prediction of Gas Ratio	$E_{9,1}$	$E_{9,2}$	$E_{9,3}$
GLM Prediction of Total Gas Flow	$E_{10,1}$	$E_{10,2}$	$E_{10,3}$

Table 3. Evidence Labels for GLM Prediction of Input Responses

Evidence Variable (r)	High (s=1)	Medium (s=2)	Low (s=3)
GLM-Based Prediction of Pressure	$P(F_{1,1}/M_4)$	$P(F_{1,2}/M_4)$	$P(F_{1,3}/M_4)$
GLM-Based Prediction of Top Power	$P(F_{2,1}/M_4)$	$P(F_{2,2}/M_4)$	$P(F_{2,3}/M_4)$
GLM-Based Prediction of RF Power	$P(F_{3,1}/M_4)$	$P(F_{3,2}/M_4)$	$P(F_{3,3}/M_4)$
GLM-Based Prediction of Gas Ratio	$P(F_{4,1}/M_4)$	$P(F_{4,2}/M_4)$	$P(F_{4,3}/M_4)$
GLM-Based Prediction of Total Gas Flow	$P(F_{5,1}/M_4)$	$P(F_{5,2}/M_4)$	$P(F_{5,3}/M_4)$

Table 4. Probabilities for GLM Prediction Based on Combinations of Evidence

Model Averaged Result	High (s=1)	Medium (s=2)	Low (s=3)
Tree/GLM Prediction of Pressure	$P(F_{1,1})$	$P(F_{1,2})$	$P(F_{1,3})$
Tree/GLM Prediction of Top Power	$P(F_{2,1})$	$P(F_{2,2})$	$P(F_{2,3})$
Tree/GLM Prediction of RF Power	$P(F_{3,1})$	$P(F_{3,2})$	$P(F_{3,3})$
Tree/GLM Prediction of Gas Ratio	$P(F_{4,1})$	$P(F_{4,2})$	$P(F_{4,3})$
Tree/GLM Prediction of Total Gas Flow	$P(F_{5,1})$	$P(F_{5,2})$	$P(F_{5,3})$

Table 5. Final Fault Probabilities for Combined Model Prediction of Input Responses

Wafer	$E_{1,1}$	$E_{1,2}$	$E_{1,3}$	$E_{2,1}$	$E_{2,2}$	$E_{2,3}$
5	0	1.0000	0	0.8889	0	0.1111
7	0	1.0000	0	0.9000	0	0.1000
10	0.6667	0	0.3333	0.9091	0	0.0909
14	0	1.0000	0	0	1.0000	0
17	0	1.0000	0	0.9167	0	0.0833
19	0.5714	0.1429	0.2857	0.9231	0	0.0769
20	0	1.0000	0	0.9286	0	0.0714
24	0.5000	0.1250	0.3750	0.9333	0	0.0667
25	0	0	1.0000	0	0	1.0000
26	0	0	1.0000	0	0	1.0000
27	0.5556	0.1111	0.3333	0	0.0769	0.9231
30	0	0.0714	0.9286	0	1.0000	0
33	0	0.0667	0.9333	0	0.4000	0.6000
36	0	0.0625	0.9375	0	0.0714	0.9286
43	0.6250	0	0.3750	0	0.9231	0.0769
45	0.5556	0	0.4444	0	0.9286	0.0714
46	0	1.0000	0	0	0.3333	0.6667
48	0.6000	0	0.4000	0	0.4286	0.5714
50	0	1.0000	0	0	0.5000	0.5000
51	0	1.0000	0	0.9375	0	0.0625

Table 6. Tree Model Prediction of Input Responses Pressure and Top Power

Wafer	$E_{3,1}$	$E_{3,2}$	$E_{3,3}$	$E_{4,1}$	$E_{4,2}$	$E_{4,3}$
5	0.1667	0.1667	0.6667	0	0	1.0000
7	0.1429	0.1429	0.7143	0	0.1667	0.8333
10	0.2500	0.1250	0.6250	0	0	1.0000
14	0.6250	0.3750	0	0	0	1.0000
17	0.4444	0.3333	0.2222	0.1429	0.1429	0.7143
19	0.4000	0.4000	0.2000	0.0500	0	0.9500
20	0.3333	0.1111	0.5556	0.0476	0.0476	0.9048
24	0.3636	0.3636	0.2727	0.2500	0.1250	0.6250
25	0.3333	0.4166	0.2500	0.0455	0.0455	0.9091
26	0	0	1.0000	0.2500	0.1250	0.6250
27	0.3077	0.3846	0.3077	0.0435	0.0435	0.9130
30	0	0.1111	0.8889	0.2222	0.2222	0.5556
33	0	0.1000	0.9000	0.0417	0.0833	0.8750
36	0	0.1818	0.8182	0.2000	0.2000	0.6000
43	0.2000	0.4000	0.4000	0.6000	0.4000	0
45	0.1667	0.3333	0.5000	0.5000	0.3333	0.1667
46	0.2857	0.2857	0.4286	0.4286	0.4286	0.1429
48	0.3750	0.2500	0.3750	0.3750	0.5000	0.1250
50	0.6667	0.3333	0	0.0400	0.0800	0.8800
51	0.3000	0.2000	0.5000	0.0385	0.0769	0.8846

Table 7. Tree Model Prediction of Input Responses RF Power and Gas Ratio

Wafer	$E_{5,1}$	$E_{5,2}$	$E_{5,3}$
5	0.8000	0	0.2000
7	0.6667	0.1667	0.1667
10	0.5714	0.2857	0.1429
14	0.2000	0.4000	0.4000
17	0.8000	0.1000	0.1000
19	0.8182	0.0909	0.0909
20	0.1000	0	0.9000
24	0.7500	0.0833	0.1667
25	0.0909	0.0909	0.8182
26	0.5000	0.3750	0.1250
27	0.0833	0.0833	0.8333
30	0.1667	0.5000	0.3333
33	0.7692	0.0769	0.1538
36	0.1538	0.0769	0.7692
43	0.1429	0.4286	0.4286
45	0	0.6667	0.3333
46	0.5556	0.3333	0.1111
48	0.1429	0.5714	0.2857
50	0.7857	0.0714	0.1429
51	0.1429	0.0714	0.7857

Table 8. Tree Model Prediction of Input Response Total Gas Flow

Wafer	$P(F_{1,1}/M_2)$	$P(F_{1,2}/M_2)$	$P(F_{1,3}/M_2)$	$P(F_{2,1}/M_2)$	$P(F_{2,2}/M_2)$	$P(F_{2,3}/M_2)$
5	0.2952	0.2988	0.4059	0.3238	0.3238	0.3524
7	0.2872	0.3221	0.3908	0.3423	0.3274	0.3304
10	0.2954	0.3556	0.3490	0.2595	0.3275	0.4130
14	0.6264	0.3297	0.0440	0.4286	0.4286	0.1429
17	0.4148	0.4441	0.1410	0.4467	0.4485	0.1048
19	0.4201	0.4450	0.1348	0.3760	0.4145	0.2095
20	0.1688	0.1753	0.6559	0.1833	0.1833	0.6334
24	0.4190	0.4266	0.1543	0.3594	0.3974	0.2432
25	0.0436	0.0876	0.8687	0.0631	0.0608	0.8761
26	0.0667	0.0848	0.8485	0.0430	0.1490	0.8080
27	0.0673	0.0984	0.8343	0.2031	0.4150	0.3819
30	0.2331	0.2322	0.5347	0.1578	0.1605	0.6817
33	0.1556	0.1715	0.6728	0.0394	0.0527	0.9079
36	0.0658	0.1704	0.7638	0.1525	0.1401	0.7074
43	0.2430	0.3420	0.4150	0.3328	0.3545	0.3127
45	0.2440	0.3735	0.3825	0.3108	0.3432	0.3460
46	0.2755	0.3078	0.4168	0.4039	0.4106	0.1855
48	0.2576	0.3483	0.3941	0.3189	0.3491	0.3320
50	0.3603	0.3204	0.3193	0.4793	0.4765	0.0442
51	0.1682	0.1868	0.6450	0.1940	0.1932	0.6128

Table 9. Fault Probabilities for Pressure and Top Power extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$P(F_{3,1}/M_2)$	$P(F_{3,2}/M_2)$	$P(F_{3,3}/M_2)$	$P(F_{4,1}/M_2)$	$P(F_{4,2}/M_2)$	$P(F_{4,3}/M_2)$
5	0.4467	0.4467	0.1067	0.3114	0.3114	0.3771
7	0.4877	0.4012	0.1111	0.2759	0.3801	0.3441
10	0.3166	0.5048	0.1785	0.3118	0.3061	0.3821
14	0.4286	0.4286	0.1429	0.2912	0.2637	0.4451
17	0.4772	0.4698	0.0529	0.2437	0.3055	0.4507
19	0.4307	0.4575	0.1117	0.2901	0.2747	0.4352
20	0.3296	0.3296	0.3409	0.3196	0.3229	0.3575
24	0.4462	0.3998	0.1540	0.3015	0.2796	0.4188
25	0.1107	0.1211	0.7682	0.0604	0.0366	0.9030
26	0.1824	0.2187	0.5989	0.0182	0.2000	0.7818
27	0.2795	0.2501	0.4703	0.2597	0.2510	0.4892
30	0.2115	0.2730	0.5155	0.4432	0.2181	0.3387
33	0.1840	0.2766	0.5394	0.4179	0.1567	0.4254
36	0.1198	0.1379	0.7423	0.1442	0.0575	0.7983
43	0.2751	0.3564	0.3686	0.2611	0.2415	0.4974
45	0.3147	0.3405	0.3448	0.2324	0.2584	0.5092
46	0.4547	0.3179	0.2274	0.3083	0.2946	0.3972
48	0.3061	0.3329	0.3610	0.2386	0.2391	0.5223
50	0.3825	0.3888	0.2287	0.3244	0.2668	0.4088
51	0.3410	0.3370	0.3220	0.3058	0.3266	0.3675

Table 10. Fault Probabilities for Top Power and Gas Ratio extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$P(F_{5,1}/M_2)$	$P(F_{5,2}/M_2)$	$P(F_{5,3}/M_2)$
5	0.3524	0.3524	0.2952
7	0.3247	0.4058	0.2695
10	0.3803	0.2348	0.3849
14	0.5000	0.5000	0
17	0.4524	0.4444	0.1031
19	0.5568	0.2749	0.1683
20	0.3914	0.4037	0.2049
24	0.5187	0.2620	0.2193
25	0.2258	0.2274	0.5468
26	0.2121	0.0606	0.7273
27	0.3028	0.3009	0.3962
30	0.1869	0.1002	0.7129
33	0.1457	0.0663	0.7880
36	0.2603	0.0686	0.6711
43	0.4691	0.2536	0.2772
45	0.4322	0.2612	0.3066
46	0.3449	0.3597	0.2955
48	0.3828	0.2746	0.3427
50	0.4124	0.4195	0.1681
51	0.3731	0.3906	0.2362

Table 11. Fault Probabilities for Total Gas Flow extracted from Tree Model Prediction Based on Combinations of Evidence

Wafer	$E_{6,1}$	$E_{6,2}$	$E_{6,3}$	$E_{7,1}$	$E_{7,2}$	$E_{7,3}$
5	0.4929	0.5071	0	1.0000	0	0.0000
7	0.5000	0.5000	0	1.0000	0	0.0000
10	0	1.0000	0.0000	1.0000	0	0.0000
14	0	0.9962	0.0038	0	1.0000	0.0000
17	0.1147	0.8853	0	1.0000	0	0.0000
19	0	0.9999	0.0001	1.0000	0	0.0000
20	0	0	1.0000	1.0000	0	0.0000
24	1.0000	0	0.0000	1.0000	0	0.0000
25	0	0	1.0000	0	0	1.0000
26	0	1.0000	0.0000	0	1.0000	0.0000
27	1.0000	0	0.0000	0	0	1.0000
30	0	0.4919	0.5081	1.0000	0	0.0000
33	0	0	1.0000	0.5000	0.5000	0
36	0	0	1.0000	0	0	1.0000
43	0	0	1.0000	0	1.0000	0.0000
45	1.0000	0	0.0000	0	1.0000	0.0000
46	0	1.0000	0.0000	0	1.0000	0.0000
48	1.0000	0	0.0000	0	1.0000	0.0000
50	0	0	1.0000	0	1.0000	0.0000
51	0	1.0000	0.0000	0	0	1.0000

Table 12. GLM Prediction of Input Responses Pressure and Top Power

Wafer	$E_{8,1}$	$E_{8,2}$	$E_{8,3}$	$E_{9,1}$	$E_{9,2}$	$E_{9,3}$
5	0.3835	0.1323	0.4842	0	0	1.0000
7	0.0197	0.0445	0.9358	0	0	1.0000
10	0.3620	0.1099	0.5281	0	0	1.0000
14	0.0224	0.1184	0.8592	0	0	1.0000
17	0.6893	0.2408	0.0699	0	0.1106	0.8894
19	0.5287	0.2700	0.2013	0	0	1.0000
20	0.3791	0.3003	0.3206	0	0	1.0000
24	0.3805	0.0699	0.5496	1.0000	0	0.0000
25	0.1001	0.1709	0.7291	0	0	1.0000
26	0.0671	0.2106	0.7224	0	0	1.0000
27	0.1325	0.6137	0.2539	0	0	1.0000
30	0.3720	0.3295	0.2984	0	0	1.0000
33	0.0697	0.2175	0.7128	0	0	1.0000
36	0.0695	0.1461	0.7844	0	0	1.0000
43	0.2524	0.3570	0.3907	0.5000	0.5000	0
45	0.5960	0.4040	0	1.0000	0	0.0000
46	0.0045	0.1808	0.8148	0	1.0000	0.0000
48	0.4817	0.5183	0	0.8847	0.1153	0.0000
50	0.0576	0.2249	0.7176	0	0	1.0000
51	0.0598	0.1218	0.8184	0	0	1.0000

Table 13. GLM Prediction of Input Responses RF Power and Gas Ratio

Wafer	$E_{10,1}$	$E_{10,2}$	$E_{10,3}$
5	0.3096	0	0.6904
7	0.3937	0	0.6063
10	0.8744	0	0.1256
14	0.5884	0.0032	0.4085
17	0.1865	0	0.8135
19	0.7168	0	0.2832
20	0.1370	0	0.8630
24	0.8311	0	0.1689
25	0.1554	0	0.8446
26	0.3980	0.6020	0
27	0.0336	0.9664	0
30	0.6287	0.1096	0.2617
33	0.7494	0.2506	0.0000
36	0.1988	0	0.8012
43	0.3077	0.6923	0
45	0.2126	0.7874	0
46	0.0906	0	0.9094
48	0.1208	0.8792	0
50	0.5627	0.2745	0.1628
51	0.0688	0	0.9312

Table 14. GLM Prediction of Input Response Total Gas Flow

Wafer	$P(F_{1,1}/M_4)$	$P(F_{1,2}/M_4)$	$P(F_{1,3}/M_4)$	$P(F_{2,1}/M_4)$	$P(F_{2,2}/M_4)$	$P(F_{2,3}/M_4)$
5	0.2485	0.2485	0.5030	0.2599	0.3905	0.3496
7	0.1484	0.1484	0.7032	0.2410	0.2469	0.5121
10	0.3800	0.3800	0.2400	0.3800	0.3800	0.2400
14	0.2363	0.2272	0.5365	0.2295	0.2295	0.5410
17	0.3387	0.3479	0.3134	0.3318	0.3819	0.2863
19	0.4097	0.4097	0.1805	0.4097	0.4097	0.1806
20	0.2566	0.2566	0.4868	0.0425	0.0425	0.9151
24	0.5248	0.2666	0.2086	0.5248	0.2667	0.2086
25	0.0052	0.1591	0.8357	0.0612	0.0612	0.8777
26	0.3796	0.3796	0.2408	0.3796	0.3796	0.2408
27	0.1992	0.1992	0.6017	0.3937	0.5217	0.0846
30	0.3732	0.4140	0.2128	0.2430	0.2430	0.5140
33	0.3768	0.3856	0.2376	0.0356	0.0356	0.9288
36	0.0046	0.1617	0.8337	0.0617	0.0617	0.8765
43	0.2451	0.4198	0.3351	0.2521	0.4357	0.3122
45	0.1919	0.7020	0.1060	0.3892	0.3484	0.2625
46	0.3281	0.3441	0.3278	0.3282	0.3441	0.3278
48	0.1965	0.6166	0.1869	0.3518	0.3552	0.2931
50	0.3400	0.3307	0.3293	0.0420	0.0420	0.9160
51	0.0014	0.1919	0.8067	0.0814	0.0814	0.8373

Table 15. Fault Probabilities for Pressure and Top Power extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{3,1}/M_4)$	$P(F_{3,2}/M_4)$	$P(F_{3,3}/M_4)$	$P(F_{4,1}/M_4)$	$P(F_{4,2}/M_4)$	$P(F_{4,3}/M_4)$
5	0.3849	0.3849	0.2301	0.3197	0.3203	0.3600
7	0.3989	0.3989	0.2021	0.3312	0.3312	0.3376
10	0.4791	0.4791	0.0419	0.2438	0.3541	0.4021
14	0.4310	0.4310	0.1379	0.3137	0.3393	0.3469
17	0.3724	0.3565	0.2712	0.3028	0.3350	0.3622
19	0.4528	0.4528	0.0945	0.2393	0.3320	0.4287
20	0.0457	0.0457	0.9086	0.2411	0.3794	0.3795
24	0.8874	0.0563	0.0563	0.3667	0.2376	0.3957
25	0.0768	0.1286	0.7946	0.1171	0.0429	0.8399
26	0.5000	0.5000	0.0000	0.2676	0.2962	0.4362
27	0.3333	0.3333	0.3333	0.3333	0.3333	0.3333
30	0.3495	0.3495	0.3009	0.2908	0.3233	0.3859
33	0.3333	0.3333	0.3333	0.5023	0.2625	0.2352
36	0.0728	0.1391	0.7881	0.1189	0.0566	0.8246
43	0.1667	0.5128	0.3205	0.3311	0.4580	0.2109
45	0.2625	0.4750	0.2625	0.0430	0.0634	0.8937
46	0.3031	0.3937	0.3031	0.3279	0.3359	0.3362
48	0.3070	0.4000	0.2931	0.0313	0.0291	0.9395
50	0.2791	0.2791	0.4419	0.6014	0.2010	0.1977
51	0.0229	0.0229	0.9541	0.0793	0.0793	0.8414

Table 16. Fault Probabilities for RF Power and Gas Ratio extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{5,1}/M_4)$	$P(F_{5,2}/M_4)$	$P(F_{5,3}/M_4)$
5	0.5138	0.3248	0.1614
7	0.3489	0.3391	0.3120
10	0.4120	0.4120	0.1761
14	0.3556	0.3556	0.2888
17	0.5511	0.4171	0.0318
19	0.4664	0.4664	0.0672
20	0.2265	0.2265	0.5471
24	0.5870	0.2065	0.2065
25	0.2069	0.1108	0.6823
26	0.3796	0.3796	0.2408
27	0.3333	0.3333	0.3333
30	0.3403	0.3403	0.3194
33	0.0957	0.0957	0.8085
36	0.2038	0.1001	0.6961
43	0.3064	0.3872	0.3064
45	0.6026	0.1987	0.1987
46	0.4524	0.2761	0.2716
48	0.6760	0.1620	0.1620
50	0.0942	0.0941	0.8117
51	0.0605	0.0605	0.8789

Table 17. Fault Probabilities for Total Gas Flow extracted from GLM Prediction Based on Combinations of Evidence

Wafer	$P(F_{1,1})$	$P(F_{1,2})$	$P(F_{1,3})$	$P(F_{2,1})$	$P(F_{2,2})$	$P(F_{2,3})$
5	0.2464	0.7536	0	0.9500	0	0.0500
7	0	1.0000	0	0.8128	0.0818	0.1053
10	0.2427	0.5228	0.2345	0.8033	0.0950	0.1017
14	0	0.9981	0.0019	0	1.0000	0.0000
17	0.2170	0.6694	0.1136	0.9687	0	0.0313
19	0.3825	0.4887	0.1289	0.8230	0.1024	0.0746
20	0	1.0000	0	0.9722	0	0.0278
24	0.6994	0.1121	0.1885	0.8562	0.0667	0.0771
25	0.0134	0.0598	0.9268	0.0339	0.0333	0.9328
26	0	1.0000	0.0000	0	1.0000	0.0000
27	0.1332	0.1488	0.7180	0	0.0313	0.9688
30	0.1480	0.2932	0.5588	0.3440	0.3282	0.3278
33	0.0389	0.0554	0.9057	0.0375	0.0442	0.9184
36	0.0176	0.0944	0.8880	0.0484	0.0585	0.8931
43	0.1215	0.1710	0.7075	0.1462	0.6857	0.1681
45	0.8333	0	0.1667	0	0.9773	0.0227
46	0.0820	0.8360	0.0820	0.2840	0.5413	0.1747
48	0.8235	0	0.1765	0.2474	0.5133	0.2393
50	0	0	1.0000	0.0420	0.0420	0.9160
51	0	1.0000	0.0000	0.1743	0.0824	0.7433

Table 18. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Responses Pressure and Top Power

Wafer	$P(F_{3,1})$	$P(F_{3,2})$	$P(F_{3,3})$	$P(F_{4,1})$	$P(F_{4,2})$	$P(F_{4,3})$
5	0.2632	0.1376	0.5992	0.1578	0.1579	0.6843
7	0.4433	0.4000	0.1566	0.1518	0.2135	0.6347
10	0.4205	0.2945	0.2850	0.1389	0.1650	0.6960
14	0.5604	0.3681	0.0714	0.1512	0.1508	0.6980
17	0.5001	0.3437	0.1562	0.1644	0.2156	0.6200
19	0.4602	0.3843	0.1554	0.1416	0.1517	0.7067
20	0.2568	0.2143	0.5289	0.1491	0.1845	0.6664
24	0.5379	0.2096	0.2525	0.4795	0.1501	0.3703
25	0.1696	0.1876	0.6428	0.0520	0.0279	0.9201
26	0.1874	0.2323	0.5803	0.1169	0.1468	0.7363
27	0.3064	0.2917	0.4018	0.1561	0.1539	0.6900
30	0.2324	0.2624	0.5051	0.2381	0.1729	0.5891
33	0.1468	0.2261	0.6271	0.1116	0.0535	0.8349
36	0.0655	0.1391	0.7954	0	0	1.0000
43	0.2664	0.3390	0.3946	0.2611	0.2415	0.4974
45	0.5960	0.4040	0	0.4751	0.1429	0.3820
46	0.4547	0.3179	0.2274	0	1.0000	0.0000
48	0.3944	0.4591	0.1465	0.7151	0.2395	0.0455
50	0.6471	0.3529	0	0.0872	0.0789	0.8339
51	0.1383	0.1533	0.7084	0.0881	0.0994	0.8124

Table 19. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Responses RF Power and Gas Ratio

Wafer	$P(F_{5,1})$	$P(F_{5,2})$	$P(F_{5,3})$
5	0.4940	0.1693	0.3368
7	0.3247	0.4058	0.2695
10	0.4120	0.4120	0.1761
14	0.1429	0.4286	0.4286
17	0.6029	0.3375	0.0595
19	0.6517	0.2020	0.1463
20	0.3914	0.4037	0.2049
24	0.6901	0.1318	0.1781
25	0.1850	0.1092	0.7058
26	0.4898	0.3398	0.1704
27	0.1987	0.4158	0.3855
30	0.3233	0.2425	0.4341
33	0.7556	0.1729	0.0714
36	0.2168	0.0577	0.7255
43	0.1818	0.3636	0.4545
45	0.5174	0.2299	0.2526
46	0.3449	0.3597	0.2955
48	0.5294	0.2183	0.2523
50	0.4124	0.4195	0.1681
51	0.1244	0.0860	0.7896

Table 20. Final Fault Probabilities for Combined Tree/GLM Prediction of Input Response Total Gas Flow

Appendix F

GLM Results - Coefficients, Linear Predictions, and Fitted Values

Lam Rainbow 4400 DOE Data

Sensor Signal	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
Intercept	6033.465	1103.46	194.8378	15760.9	-3597.385
Endpoint A	0.1180625	0.05647331	5.57e-06	0.0	0.7582921
Endpoint B	-0.0512072	-0.01877498	0.00064116	0.0	-0.8847277
Endpoint C	0.0	0.0	0.0	0.0	-0.117877
Measured Pressure	0.0	0.0	0.0	-0.6512974	2.113492
Measured Power	-3.662645	0.0	0.0	0.0	0.0
RF Tune	-1.363389	-0.2441685	-0.01007087	-1.095258	0.0
RF Load	1.036644	0.1320364	-0.0059625	0.6196095	0.0
Impedance	0.2277873	0.03630299	-0.0028008	0.1866142	0.2038256
Phase	-0.02358554	0.0	0.002000781	0.0	0.0
Voltage	0.0	0.0	0.0	0.0	26.5984
DC Bias	2.417183	-0.2965973	0.02772226	0.0	4.747143
MFC3	0.0	0.0	0.0	0.09734656	0.0
MFC6	0.0	0.0	0.0	0.2525355	0.0

Table 1. Coefficients α (Intercept) and β (Sensor Signals) for GLM high/not high

Sensor Signal	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
Intercept	-535.076	875.3149	1012.463	-830.7934	94717.59
Endpoint A	0.08441036	-0.07649621	0.01117361	0.0	3.643574
Endpoint B	-0.03095719	0.04533356	-0.01752973	0.0	6.628609
Endpoint C	0.0	0.0	0.0	0.0	0.541935
Measured Pressure	0.0	0.0	0.0	-0.2577119	-29.71684
Measured Power	-5.26588	0.0	0.0	0.0	0.0
RF Tune	0.5057438	0.05350584	-0.2942862	-0.5693966	0.0
RF Load	-0.681124	-0.09976484	0.2807996	0.2882431	0.0
Impedance	-0.1080034	-0.03289002	0.05214612	0.01801979	-8.63831
Phase	0.04779316	0.0	-0.03101532	0.0	0.0
Voltage	0.0	0.0	0.0	0.0	-602.0749
DC Bias	-3.882982	-0.227088	-0.3693392	0.0	291.726
MFC3	0.0	0.0	0.0	-0.1455538	0.0
MFC6	0.0	0.0	0.0	-0.1029862	0.0

Table 2. Coefficients α (Intercept) and β (Sensor Signals) for GLM low/not low

Wafer Index	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
13	-31.07658	-328.24014	-21.34860	-0.04975429	-80.78645
14	69.11116	20.99552	23.44724	-0.09929155	163.19890
15	-88.99304	23.11608	55.05386	-4.73418358	-268.42191
16	-23.37999	-175.09496	22.14806	-1.31987168	-31.97798
17	-103.69070	26.53172	-69.35085	0.79702548	-152.60930
18	-21.95443	-138.34974	-24.26163	-1.31712144	-31.57047
19	-56.10889	-22.41984	22.88390	0.48083937	-110.49687
20	21.93407	-60.08844	-23.02642	-0.30928852	-24.55979
21	48.97817	219.96305	-61.98213	-0.84107893	-75.24095
22	-29.86198	-131.47422	-24.73730	-2.35891552	-24.98774
23	30.65133	-236.08670	26.43880	6.93334464	-21.74644
24	-44.92538	-336.25333	-25.02860	-2.35935046	-56.96295
25	22.16068	348.01912	-66.95689	-2.05706377	71.12561
26	-22.12044	-138.20404	28.94177	-0.56195441	-75.05824
27	-22.34751	-243.67748	-30.00589	-0.82402679	-22.81168
28	-58.15216	150.69097	-24.01055	-0.36317654	-106.54479
29	21.69884	-20.93446	24.33314	1.85150175	21.08834
30	61.98047	-850.77324	-51.49595	7.67720130	227.92449
31	-52.81324	-68.88791	-30.69198	-1.09760642	57.73761
32	-21.49399	-22.38293	-23.42370	-1.10424828	-20.84794
33	-23.65133	-22.07345	55.17651	-1.93657495	22.03989
34	-40.03901	-169.50612	-25.42994	-1.71814908	-63.65668
35	-24.07404	-45.19692	-23.12809	-2.01708995	-34.96037
36	-75.07544	21.80197	22.38799	-0.18950309	23.64132

Table 3. Linear prediction values η for GLM high/not high (training data set)

Wafer Index	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
13	-23.19661	-21.95591	-27.13670	-17.7231395	-20.70016
14	-114.49963	-1539.05417	-57.36638	-0.6252895	-197.90720
15	131.09434	-24.21029	-120.80896	1.9309701	23.51952
16	-28.72042	-22.16124	-75.16143	-11.1588042	-77.67232
17	21.47838	-22.94332	22.06067	-28.2328600	21.88822
18	-43.23877	-4498.09476	-27.80798	-7.4121666	-59.27739
19	46.14436	21.66595	-65.11199	-17.1326511	23.50919
20	-72.75644	-8788.57304	-23.93627	-13.9856070	-36.76921
21	-105.26392	-11077.0561	25.13792	-2.8764302	21.99208
22	-42.25181	-6535.93198	-28.90650	-0.6780194	-81.17452
23	-23.48250	8588.72182	-24.69102	-10.5264270	98.53513
24	-22.02992	-2746.17455	-28.97507	-3.9321943	-61.44253
25	-96.83088	-13585.0389	21.44094	0.1068055	-105.29563
26	23.37219	1629.69226	-70.40249	-0.2489874	24.03229
27	-55.98897	-4920.35123	-21.85021	-16.9278863	-83.30660
28	-20.54845	-3347.58797	-21.09412	-17.5702044	-25.59050
29	-27.89210	22.59182	-50.86425	-1.8927016	-28.86773
30	-22.97592	21.96230	72.41534	-1.5119588	-50.40413
31	-24.10040	-9823.50739	-21.57770	-21.2442108	-152.87744
32	-32.40460	-8516.77679	-29.26409	-12.2632610	-52.48250
33	23.60827	22.84979	-24.32695	4.4984370	-23.39450
34	-26.49978	-6382.57298	-25.74306	-5.7309749	-45.19601
35	-34.31020	-10126.6211	-28.26885	-2.2017806	-57.15653
36	21.42460	-521.12242	-61.60648	-12.8495331	-162.49988

Table 4. Linear prediction values η for GLM low/not low (training data set)

Index	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
13	3.188687e-14	5.350844e-10	0.487563992	2.220446e-16	2.220446e-16
14	1.000000e+00	1.000000e+00	0.475197485	1.000000e+00	1.000000e+00
15	2.220446e-16	1.000000e+00	0.008713038	2.220446e-16	1.000000e+00
16	7.017781e-11	1.000000e+00	0.210839644	1.294614e-14	2.220446e-16
17	2.220446e-16	2.220446e-16	0.689337843	2.220446e-16	1.000000e+00
18	2.919514e-10	2.906081e-11	0.211297609	1.945902e-14	2.220446e-16
19	2.220446e-16	1.000000e+00	0.617946059	2.220446e-16	1.833095e-10
20	1.000000e+00	9.994354e-11	0.423288412	2.156852e-11	2.220446e-16
21	1.000000e+00	2.220446e-16	0.301307599	2.220446e-16	1.000000e+00
22	1.074254e-13	1.806034e-11	0.086359723	1.405923e-11	2.220446e-16
23	1.000000e+00	1.000000e+00	0.999026214	3.594504e-10	2.220446e-16
24	2.220446e-16	1.349633e-11	0.086325412	2.220446e-16	2.220446e-16
25	1.000000e+00	2.220446e-16	0.113340570	1.000000e+00	1.000000e+00
26	2.472939e-10	1.000000e+00	0.363095368	2.220446e-16	2.220446e-16
27	1.970609e-10	9.302681e-14	0.304909553	1.238831e-10	2.220446e-16
28	2.220446e-16	3.735528e-11	0.410190833	2.220446e-16	1.000000e+00
29	1.000000e+00	1.000000e+00	0.864303329	1.000000e+00	8.096171e-10
30	1.000000e+00	2.220446e-16	0.999536945	1.000000e+00	2.220446e-16
31	2.220446e-16	4.684295e-14	0.250188648	1.000000e+00	2.220446e-16
32	4.626762e-10	6.717656e-11	0.248944741	8.827808e-10	1.902038e-10
33	5.350060e-11	1.000000e+00	0.126024616	1.000000e+00	2.591921e-10
34	2.220446e-16	9.034801e-12	0.152109727	2.220446e-16	2.220446e-16
35	3.505705e-11	9.028168e-11	0.117420232	6.559987e-16	2.220446e-16
36	2.220446e-16	1.000000e+00	0.452765499	1.000000e+00	1.000000e+00

Table 5. Fitted probability values, μ, for GLM high/not high (training data set)

Index	Pressure Model	Power Model	Gas Ratio Model	Total Gas Flow Model	Gap Spacing Model
13	8.430285e-11	1.639389e-12	0.487563992	2.220446e-16	2.915213e-10
14	2.220446e-16	2.220446e-16	0.475197485	1.000000e+00	2.220446e-16
15	1.000000e+00	2.220446e-16	0.008713038	2.220446e-16	3.059171e-11
16	3.364198e-13	2.220446e-16	0.210839644	1.294614e-14	2.374089e-10
17	1.000000e+00	1.000000e+00	0.689337843	2.220446e-16	1.086030e-10
18	2.220446e-16	8.378143e-13	0.211297609	1.945902e-14	2.220446e-16
19	1.000000e+00	2.220446e-16	0.617946059	2.220446e-16	1.000000e+00
20	2.220446e-16	4.023557e-11	0.423288412	2.156852e-11	2.220446e-16
21	2.220446e-16	1.000000e+00	0.301307599	2.220446e-16	2.220446e-16
22	2.220446e-16	2.792984e-13	0.086359723	1.405923e-11	2.220446e-16
23	6.334010e-11	1.891590e-11	0.999026214	3.594504e-10	1.000000e+00
24	2.707254e-10	2.607868e-13	0.086325412	2.220446e-16	2.220446e-16
25	2.220446e-16	1.000000e+00	0.113340570	1.000000e+00	2.220446e-16
26	1.000000e+00	2.220446e-16	0.363095368	2.220446e-16	1.000000e+00
27	2.220446e-16	3.240207e-10	0.304909553	1.238831e-10	2.220446e-16
28	1.191031e-09	6.901445e-10	0.410190833	2.220446e-16	2.220446e-16
29	7.702221e-13	2.220446e-16	0.864303329	1.000000e+00	1.000000e+00
30	1.051200e-10	1.000000e+00	0.999536945	1.000000e+00	1.000000e+00
31	3.414510e-11	4.255218e-10	0.250188648	1.000000e+00	2.220446e-16
32	8.450061e-15	1.953284e-13	0.248944741	8.827808e-10	2.220446e-16
33	1.000000e+00	2.722316e-11	0.126024616	1.000000e+00	1.000000e+00
34	3.099494e-12	6.605861e-12	0.152109727	2.220446e-16	2.220446e-16
35	1.256813e-15	5.284365e-13	0.117420232	6.559987e-16	2.220446e-16
36	1.000000e+00	2.220446e-16	0.452765499	1.000000e+00	2.220446e-16

Table 6. Fitted probability values, μ, for GLM low/not low (training data set)

Lam TCP 9600 DOE Data

Sensor Signal	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
Intercept	126808.3	571126.5	-2213.071	1028127	-8120.773
Endpoint A	-0.1525693	0.4750922	-0.00043223	-3.783755	0.0
Endpoint B	0.01247271	0.004142023	0.002598772	-0.1269219	0.0
Measured Pressure	0.0	0.0	0.0	0.0	-0.00082065
Measured Power	-24.96604	-11.05698	0.0	0.0	0.0
TCP Tune	-0.1995803	0.0	0.01170266	-1.012942	-3.2174e-06
TCP Load	0.0	0.0	0.004482656	0.0	0.001655944
TCP Impedance	0.0	-0.6840775	-0.00460416	0.0	0.0
RF Tune	0.0	0.0	0.0	-0.3779512	0.0
RF Load	0.6559789	0.0	0.0	1.132243	0.0
RF Impedance	-0.1540733	0.1709703	0.0	-0.3108275	0.00682788
RF Phase	-0.0054989	0.0	0.0	0.1734938	-0.00073575
Voltage	0.0	0.0	0.0	0.0	0.0
DC Bias	-127.5749	-570.082	1.936264	-1021.112	8.068185

Table 7. Coefficients α (Intercept) and β (Sensor Signals) for GLM high/not high

Sensor Signal	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
Intercept	10215.83	1184932	13230.52	-145190.1	112480.7
Endpoint A	-0.2632891	-2.167433	-0.00339255	-0.00675703	0.0
Endpoint B	0.06479333	-0.06077437	0.000564964	0.02157689	0.0
Measured Pressure	0.0	0.0	0.0	0.0	-0.00805184
Measured Power	-0.3290491	-46.04254	0.0	0.0	0.0
TCP Tune	0.1898102	0.0	0.002906738	0.2725461	0.02121397
TCP Load	0.0	0.0	0.001146442	0.0	0.01426199
TCP Impedance	0.0	0.6234987	-0.00381780	0.0	0.0
RF Tune	0.0	0.0	0.0	0.1548167	0.0
RF Load	-0.00231974	0.0	0.0	-0.4319301	0.0
RF Impedance	0.07064434	0.3491924	0.0	0.03831186	0.01390422
RF Phase	0.02240194	0.0	0.0	0.02674001	0.003302609
Voltage	0.0	0.0	0.0	0.0	0.0
DC Bias	-15.7884	-1214.476	-13.43734	143.3538	-115.0203

Table 8. Coefficients α (Intercept) and β (Sensor Signals) for GLM low/not low

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
1	-23.12199	-64.24108	-0.30782572	-195.83192	-2.53208126
2	-26.86475	-164.78477	-0.31429055	-350.22813	-2.30366961
3	233.24269	50.82985	0.11461674	23.25948	1.34853588
4	23.15793	-116.07525	-0.64710135	-213.60133	-0.52626246
6	-131.82206	-135.04470	-1.00707346	-347.96685	0.49209909
8	22.77709	-43.54938	0.05196833	-73.34766	-1.33572139
9	159.04514	-111.41799	1.74942983	-22.87685	0.34649166
11	34.09225	-75.21046	0.25185028	-254.76242	-0.25204792
12	21.90498	21.39099	-1.42943914	-73.38416	0.77783488
13	-34.01366	-22.01135	-2.05392635	-218.20932	-2.07050027
15	25.36962	22.97087	0.72954494	-681.76665	1.81700839
16	-93.85693	24.76387	0.04511707	-1047.53296	0.11841855
18	-128.77479	-21.64451	0.33786948	-332.03534	0.54109000
21	-114.23792	40.31385	-0.01653602	-1138.48021	2.17170068
22	-98.53045	-67.75134	-0.94326606	-169.26171	-1.84147286
23	63.71161	-98.41308	-0.18106859	-96.47542	-0.98227855

Table 9. Linear prediction values η for GLM high/not high (training data set

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
28	49.02813	182.91577	-1.40360596	-463.42972	-1.14575883
29	-112.44885	-154.22545	-1.40631081	-294.00885	-0.62105773
31	-21.69848	-46.07821	-2.37272770	-21.90305	0.13569527
32	-66.06819	67.59136	-1.78032759	-720.58940	1.00754544
34	-62.96012	-54.29230	-2.69808094	-135.36201	-1.47335644
35	-22.91028	-156.99596	-0.97009713	48.54970	-0.40849703
37	-22.59925	-22.07748	-2.04434899	-740.85889	0.40653179
38	-112.12183	-150.59695	-2.78148839	22.65956	-0.61737315
39	-45.23462	-51.07007	-2.23417296	-22.74371	-0.31100904
40	-37.02198	-116.82895	-3.14523983	-53.87784	-1.29200035
41	-107.21968	-48.19683	-1.84242826	-554.74381	-0.06039086
42	-103.13782	-57.86249	-2.15182427	21.60124	-1.45254824
44	-108.95108	-30.40193	-2.00216818	-21.93837	-2.85360672
47	-105.27672	-282.49376	-2.51331582	23.14843	-2.37380526
49	-97.77791	-149.95261	-3.47881605	-419.34878	-1.06211947
52	-88.69589	-152.72293	-0.02749357	-354.65243	0.61849757
53	-102.20915	-87.42280	-2.06650518	-207.95393	-2.08661887
54	-177.08711	-163.66335	-1.01478493	-21.96210	-0.99845952
55	-166.27119	-181.11006	-1.41116071	-561.91903	-1.06757284
56	-94.90394	23.55405	3.65134263	-1211.02363	-0.82892780

Table 9. Linear prediction values η for GLM high/not high (training data set

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
1	-22.49493	-179.48679	-0.4641348	-86.20728	-2.9607087
2	-55.69559	-475.25637	-2.4849975	-84.45179	-23.3567141
3	-47.40044	-176.00605	-1.4242214	-235.50700	-13.7569177
4	-88.52616	39.57912	-0.7828168	-114.47777	-1.4636564
6	37.41693	137.10859	-1.4409745	-22.68929	0.1026716
8	-39.96814	217.29334	-0.1098644	-41.44805	5.3718234
9	-26.11608	245.37777	-1.0446225	-74.45533	-4.6009229
11	-62.21728	22.13977	-0.7472296	-88.36537	-1.5463064
12	-24.43930	-43.02400	-1.1080556	-136.06231	-6.5017312
13	-69.70875	-345.50959	-1.1885036	-118.16375	-4.6821806
15	-40.77277	-353.02300	-1.7447896	-133.32530	-10.1973239
16	24.29124	-380.52487	-1.6425716	-44.88466	-12.5920431
18	51.25104	22.05788	-0.5960765	-35.82273	-1.0966399
21	23.25408	-302.25929	-1.8216777	-57.17682	-8.2047600
22	23.86183	21.02750	-0.6573965	-22.50254	0.3237691
23	-61.03758	117.02024	-1.0345026	-74.82388	-6.9701883

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
28	-101.90259	-336.23003	0.3264096	-175.41340	7.2237131
29	-21.31181	-24.94111	-1.3158827	-76.30005	-8.7020739
31	-61.75863	-22.19272	-1.2014275	-130.16767	0.0308671
32	-23.27054	-485.97983	-2.3686081	-104.41596	-12.7361656
34	-67.21301	-77.86475	-0.8581280	-104.86872	-0.3490551
35	-45.40865	361.64055	-0.5672419	-55.57780	1.0032438
37	50.58500	-22.34782	-2.3767763	-22.97924	-21.9153101
38	22.43172	449.44605	-0.8808015	-22.96400	-2.9550986
39	-53.40477	-211.21427	-1.8249431	-128.99438	-13.9824443
40	-77.59026	-234.41778	-2.2972119	-121.65839	-13.0518982
41	-36.28335	-213.09400	-1.6004555	-81.59023	-7.4951353
42	-25.41597	-20.74548	-0.8895970	-63.76137	-1.1136333
44	-66.34598	-322.61038	-0.4614485	-93.07087	-1.7213092
47	-77.62354	239.56230	-1.4623047	-22.64447	-8.8414534
49	-114.07724	-268.92563	-2.9495072	-78.43454	-23.1217399
52	-22.73295	570.83240	-0.3018914	23.53215	1.2887924
53	-69.62763	-203.98377	-1.0461081	-73.50866	-6.8050103
54	40.86102	402.55365	-0.5182663	22.05232	-0.7210980
55	21.81938	-51.57152	-1.4224097	-23.25576	-13.9576160
56	125.58630	-703.09463	-1.5169249	23.31366	-23.5914287

Table 10. Linear prediction values η for GLM low/not low (training data set)

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
1	9.083388e-11	2.220446e-16	0.42364554	2.220446e-16	0.07363954
2	2.151718e-12	2.220446e-16	0.42206781	2.220446e-16	0.09081950
3	1.000000e+00	1.000000e+00	0.52862286	1.000000e+00	0.79389016
4	1.000000e+00	2.220446e-16	0.34364304	2.220446e-16	0.37138903
6	2.220446e-16	2.220446e-16	0.26755297	2.220446e-16	0.62060080
8	1.000000e+00	2.220446e-16	0.51298916	2.220446e-16	0.20821455
9	1.000000e+00	2.220446e-16	0.85188087	1.160677e-10	0.58576656
11	1.000000e+00	2.220446e-16	0.56263187	2.220446e-16	0.43731950
12	1.000000e+00	1.000000e+00	0.19318609	2.220446e-16	0.68521329
13	1.690663e-15	2.757978e-10	0.11365625	2.220446e-16	0.11199727
15	1.000000e+00	1.000000e+00	0.67470540	2.220446e-16	0.86020677
16	2.220446e-16	1.000000e+00	0.51127736	2.220446e-16	0.52957009
18	2.220446e-16	3.980241e-10	0.58367290	2.220446e-16	0.63206594
21	2.220446e-16	1.000000e+00	0.49586609	2.220446e-16	0.89767928
22	2.220446e-16	2.220446e-16	0.28024109	2.220446e-16	0.13687719
23	1.000000e+00	2.220446e-16	0.45485612	2.220446e-16	0.27243990

Table 11. Fitted probability values, μ, for GLM high/not high (training data set

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
28	1.000000e+00	1.000000e+00	0.19724452	2.220446e-16	0.24126460
29	2.220446e-16	2.220446e-16	0.19681659	2.220446e-16	0.34954093
31	3.771122e-10	2.220446e-16	0.08527613	3.073438e-10	0.53387186
32	2.220446e-16	1.000000e+00	0.14426269	2.220446e-16	0.73253951
34	2.220446e-16	2.220446e-16	0.06308669	2.220446e-16	0.18643299
35	1.122515e-10	2.220446e-16	0.27486114	1.000000e+00	0.39927256
37	1.532044e-10	2.581501e-10	0.11462463	2.220446e-16	0.60025598
38	2.220446e-16	2.220446e-16	0.05833274	1.000000e+00	0.35037912
39	2.220446e-16	2.220446e-16	0.09672344	1.325966e-10	0.42286846
40	2.220446e-16	2.220446e-16	0.04127925	2.220446e-16	0.21551442
41	2.220446e-16	2.220446e-16	0.13676436	2.220446e-16	0.48490687
42	2.220446e-16	2.220446e-16	0.10416087	1.000000e+00	0.18960970
44	2.220446e-16	6.260539e-14	0.11897547	2.966790e-10	0.05449518
47	2.220446e-16	2.220446e-16	0.07492995	1.000000e+00	0.08519211
49	2.220446e-16	2.220446e-16	0.02992103	2.220446e-16	0.25690463
52	2.220446e-16	2.220446e-16	0.49312704	2.220446e-16	0.64987677
53	2.220446e-16	2.220446e-16	0.11239522	2.220446e-16	0.11040422
54	2.220446e-16	2.220446e-16	0.26604448	2.897216e-10	0.26924441
55	2.220446e-16	2.220446e-16	0.19605105	2.220446e-16	0.25586494
56	2.220446e-16	1.000000e+00	0.97470043	2.220446e-16	0.30387183

Table 11. Fitted probability values, μ, for GLM high/not high (training data set

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
1	1.700491e-10	2.220446e-16	0.38600540	2.220446e-16	0.07363954
2	2.220446e-16	2.220446e-16	0.07691662	2.220446e-16	0.09081950
3	2.220446e-16	2.220446e-16	0.19400065	1.000000e+00	0.79389016
4	2.220446e-16	1.000000e+00	0.31371313	2.220446e-16	0.37138903
6	1.000000e+00	1.000000e+00	0.19139448	2.220446e-16	0.62060080
8	2.220446e-16	1.000000e+00	0.47256151	2.220446e-16	0.20821455
9	4.549142e-12	1.000000e+00	0.26025906	1.160677e-10	0.58576656
11	2.220446e-16	1.000000e+00	0.32142527	2.220446e-16	0.43731950
12	2.433026e-11	2.220446e-16	0.24823356	2.220446e-16	0.68521329
13	2.220446e-16	2.220446e-16	0.23352667	2.220446e-16	0.11199727
15	2.220446e-16	2.220446e-16	0.14870558	2.220446e-16	0.86020677
16	1.000000e+00	2.220446e-16	0.16211545	2.220446e-16	0.52957009
18	1.000000e+00	1.000000e+00	0.35524185	2.220446e-16	0.63206594
21	1.000000e+00	2.220446e-16	0.13923268	2.220446e-16	0.89767928
22	1.000000e+00	1.000000e+00	0.34132468	2.220446e-16	0.13687719
23	2.220446e-16	1.000000e+00	0.26221211	2.220446e-16	0.27243990

Table 12. Fitted probability values, μ, for GLM low/not low (training data set)

Wafer Index	Pressure Model	TCP (Top) Power	RF (Bottom) Power	Gas Ratio Model	Total Gas Flow Model
28	2.220446e-16	2.220446e-16	0.58088553	2.220446e-16	0.24126460
29	5.551367e-10	1.473043e-11	0.21150411	2.220446e-16	0.34954093
31	2.220446e-16	2.300509e-10	0.23122138	3.073438e-10	0.53387186
32	7.829510e-11	2.220446e-16	0.08559802	2.220446e-16	0.73253951
34	2.220446e-16	2.220446e-16	0.29773062	2.220446e-16	0.18643299
35	2.220446e-16	1.000000e+00	0.36187350	1.000000e+00	0.39927256
37	1.000000e+00	1.970003e-10	0.08496085	2.220446e-16	0.60025598
38	1.000000e+00	1.000000e+00	0.29301171	1.000000e+00	0.35037912
39	2.220446e-16	2.220446e-16	0.13884180	1.325966e-10	0.42286846
40	2.220446e-16	2.220446e-16	0.09135413	2.220446e-16	0.21551442
41	2.220446e-16	2.220446e-16	0.16791797	2.220446e-16	0.48490687
42	9.161906e-12	9.780352e-10	0.29119300	1.000000e+00	0.18960970
44	2.220446e-16	2.220446e-16	0.38664225	2.966790e-10	0.05449518
47	2.220446e-16	1.000000e+00	0.18811507	1.000000e+00	0.08519211
49	2.220446e-16	2.220446e-16	0.04975981	2.220446e-16	0.25690463
52	1.340310e-10	1.000000e+00	0.42509517	2.220446e-16	0.64987677
53	2.220446e-16	2.220446e-16	0.25997316	2.220446e-16	0.11040422
54	1.000000e+00	1.000000e+00	0.37325773	2.897216e-10	0.26924441
55	1.000000e+00	2.220446e-16	0.19428409	2.220446e-16	0.25586494
56	1.000000e+00	2.220446e-16	0.17991478	2.220446e-16	0.30387183

Table 12. Fitted probability values, μ, for GLM low/not low (training data set)

Appendix G

Fault Category		Pattern 1		Pattern 2		Pattern 3-1		Pattern 3-2	
#	Description	%	prob	%	prob	%	prob	%	prob
1	baseline	0.4444	0.4222	0	0	0	0	0	0
2	HH extreme	0	0	0	0.2800	0	0	0	0
3	LL extreme	0	0	1.0000	1.0000	0.8000	0.8000	0	0
4	HL extreme	0.2500	0.2000	0	0	0	0	0.8667	0.8667
5	LH extreme	0.1667	0.1167	0	0	0	0	0	0
6	HH midrange	0	0	1.0000	0.6000	0	0	0.0556	0.0333
7	LL midrange	0.1250	0.0750	0	0	0	0	0	0
8	HL midrange	0.5000	0.4167	0	0	0	0	0	0
9	LH midrange	0.2500	0.1750	0	0.2400	0	0	0	0.0600

Fault Category		Pattern 4-1		Pattern 4-2		Pattern 4-3		Pattern 5	
#	Description	%	prob	%	prob	%	prob	%	prob
1	baseline	0.1481	0.0889	0	0.2133	0	0.0133	1.0000	0.6400
2	HH extreme	0.5000	0.4750	0	0	0	0	1.0000	0.8000
3	LL extreme	0	0	1.0000	0.9000	0	0.2400	0	0
4	HL extreme	0	0	0.5417	0.6167	0.8750	0.7383	0	0
5	LH extreme	0	0	0	0	0	0	1.0000	1.0000
6	HH midrange	0	0	0	0	0	0	0	0.4800
7	LL midrange	0.8889	0.5333	0	0	0	0	1.0000	1.0000
8	HL midrange	0	0.0667	0	0	0	0	0	0
9	LH midrange	0.3889	0.4733	0	0	0	0	1.0000	1.0000

Table 1. Percent of observations and average probability linking fault group to pattern

Fault Category		Pattern 6		Pattern 7		Pattern 8		Pattern 9	
#	Description	%	prob	%	prob	%	prob	%	prob
1	baseline	0	0	0	0	0	0	0	0.3200
2	HH extreme	0	0.1600	0	0	0	0	0	0
3	LL extreme	0.5000	0.5400	0	0	0	0	0	0
4	HL extreme	0.5556	0.5467	0.5556	0.3556	0	0.4800	0	0
5	LH extreme	0	0	0	0	0	0	0	0
6	HH midrange	1.0000	1.0000	0.1111	0.3600	0	0.2133	0	0
7	LL midrange	0	0.2400	0	0	0	0	0.2500	0.3700
8	HL midrange	0	0	1.0000	0.9333	1.0000	1.0000	0	0
9	LH midrange	0	0.3600	0	0	0	0	0.5000	0.7400

Table 2. Percent of observations and average probability linking fault group to pattern

Fault Category		Pattern 1			Pattern 2		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
		0.8000	0	0	0.6000	0	0
1	baseline	0.8000	0	0	0.6000	0	0
		0.8000	0.2500	0.2000	0	0.8000	0
	tune	0.6000	1.0000	0.6000	0.6000	0	0
10	capacitor	0.4500	0	0	0	0	0
	disabled	0.6000	1.0000	0.6000	0.6000	0	0
	load	0.8000	0	0	0.6000	0	0
11	capacitor	0.8000	0.7500	0.6000	0.6000	0	0
	disabled	0.8000	0	0	0.6000	0	0
1	baseline	0.8500	0.6000	0.5100	0.6000	0	0

Fault Category		Pattern 3-1			Pattern 3-2		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
		0.8000	0	0	0.8000	0	0
1	baseline	0.6000	0	0	0.6000	0	0
		0.8000	0	0	0.8000	0	0
	tune	0.6000	0	0	0.6000	0	0
10	capacitor	0.6000	0	0	0.6000	0	0
	disabled	0.6000	0	0	0.6000	0	0
	load	0.8000	0.1000	0.0800	0.8000	0.6000	0.4800
11	capacitor	0.8000	0	0	0.8000	1.0000	0.8000
	disabled	0.8000	0	0	0.8000	1.0000	0.8000
1	baseline	0.8000	0	0	0.8000	0.8000	0.6400

Table 3. Probabilities of matching pattern shape, position, and overall fit - failure data

Fault Category		Pattern 4-1			Pattern 4-2		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
1	baseline	0.6000	0	0	0.6000	0	0
		0.6000	0	0	0.6000	0	0
		0.6000	0.8000	0.4800	0.6000	0	0
10	tune capacitor disabled	0.3333	0	0	0.3333	0	0
		0.3333	0	0	0.3333	0	0
		0.3333	0	0	0.3333	0	0
11	load capacitor disabled	0.6000	0	0	0.6000	0	0
		0.6000	0	0	0.6000	0	0
		0.6000	0	0	0.6000	0	0
1	baseline	0.6000	0	0	0.6000	0	0

Fault Category		Pattern 4-3			Pattern 5		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
1	baseline	0.6000	0	0	0.8000	0	0
		0.6000	0	0	0.8000	0	0
		0.6000	0	0	0.8000	0	0
10	tune capacitor disabled	0.3333	0	0	0.6000	1.0000	0.6000
		0.3333	0	0	0.6000	1.0000	0.6000
		0.3333	0	0	0.6000	1.0000	0.6000
11	load capacitor disabled	0.6000	0	0	0.8000	0	0
		0.6000	0	0	0.8000	0	0
		0.6000	0	0	0.8000	0	0
1	baseline	0.6000	0	0	0.8000	0	0

Table 4. Probabilities of matching pattern shape, position, and overall fit - failure data

Fault Category		Pattern 6			Pattern 7		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
1	baseline	0.6000	0.8667	0.5200	0.8000	0.8000	0.6400
		0.6000	1.0000	0.6000	0.7333	0.8000	0.5867
		0.6000	0.6000	0.3600	0.8000	0.8000	0.6400
10	tune capacitor disabled	0.6000	0.6667	0.4000	0	0.8000	0
		0.6000	0	0	0	0	0
		0.6000	0.8000	0.4800	0	0.8000	0
11	load capacitor disabled	0.6000	0.6667	0.4000	0.7333	0	0
		0.6000	1.0000	0.6000	0.7333	0	0
		0.6000	0.7333	0.4400	0.8667	0	0
1	baseline	0.6000	0.8667	0.5200	1.0000	0.8000	0.8000
Fault Category		Pattern 8			Pattern 9		
#	Description	P(shape)	P(pos)	P(match)	P(shape)	P(pos)	P(match)
1	baseline	0.6000	0.6000	0.3600	0.3000	0	0
		0.6000	0.6000	0.3600	0	0	0
		0.6000	0	0	0.3000	1.0000	0.3000
10	tune capacitor disabled	0.6000	0	0	0	0	0
		0.6000	0	0	0	0	0
		0.6000	0	0	0	0	0
11	load capacitor disabled	0.2000	0.2000	0	0.3000	0.3000	0.1800
		0	0	0	0	0	0
		0.6000	0	0	0	0	0
1	baseline	0.6000	1.0000	0.6000	0	0	0

Table 5. Probabilities of matching pattern shape, position, and overall fit - failure data