Fault Diagnosis of Plasma Etch Equipment

Anna M. Ison, Wei Li and Costas J. Spanos

Department of EECS, University of California, Berkeley, CA 94720-1772
office:(510)642-9584, fax:(510)642-2739, email:annaison@eecs.berkeley.edu,
WWW: http://bcam.eecs.berkeley.edu

Abstract- The development and implementation of robust methods for fault detection promises to enhance manufacturing by improving our capability to monitor equipment and processes. In order to fully utilize this capability, it is important that the machine fault is not only detected, but also diagnosed as belonging to a fault category so that appropriate corrective action can promptly be taken. In this paper we examine the diagnostic performance of two probabilistic modeling techniques in using sensor signals to classify faults. We also discuss how the strengths of these models may be combined in a hierarchical architecture giving rise to a more powerful diagnostic tool.

INTRODUCTION

To guarantee continued success in the industry, semiconductor manufacturers compete in product differentiation and development, specifically by taking advantage of decreasing circuit geometries and through tighter specifications. However, accommodating larger wafer sizes and meeting more stringent design demands necessitates accurate and robust characterization of the manufacturing process as well as reliable prediction and control of its effects on the final wafer product. Process improvements translate directly to increased efficiency, decreased machine downtime, and savings in that early detection and diagnosis of potential problems can prevent long runs of misprocessed product. Furthermore, to be competitive economically involves increasing throughput, maintaining high yield, lowering the cost of machine ownership and speeding up the process development cycle.

Achieving these goals set by the industry requires better process characterization and control. This motivates the development of a manufacturing tool to monitor equipment and diagnose problems or abnormalities in machine behavior. Our approach is to use probabilistic models to characterize variability in the process and to identify modes of operation or machine states.

Although the techniques presented are general, we have chosen the plasma etch process as a test vehicle for the methodology. Much attention has focused on plasma etching because it is considered a critical manufacturing process and yield limiter. However, due to its complexity, the process is not easily represented by physically based models. Furthermore, in the current production situation, data gathering capabilities are surpassing the development of useful analytical tools. A system is needed to automatically extract inferences from the various data sources in a timely fashion, so that appropriate action can be taken. Ironically, although the total volume of data is significant, relevant representative data properly annotated with machine log information is still a rare commodity, and hence the situation lends itself well to statistical methods to draw inferences from a sample representative of a larger population.

In this paper we examine two techniques for diagnosing faults in machine sensor data. Tree-based models are shown to be an effective method of identifying sensor signals most sensitive to changes in the input settings of the machine. This method is compared with the performance of generalized linear models built to predict levels of the input settings based on sensor signals. Finally, we discuss how the strengths of the two methods can be combined to enhance diagnostic performance.

For practical reasons, the system is constructed using data collected easily and economically from the machine without interrupting the process. A designed experiment was conducted on a Lam Rainbow 4400 plasma etcher in the U.C. Berkeley microfabrication Laboratory, providing the multivariate real-time tool data used to build the models in this work.

MONITORING AND FAULT DETECTION

Through the use of real-time tool signals, in particular, SECSII machine information collected in-situ, we can effectively monitor the machine state without interrupting the process. Using data collected as part of the regular production process, we have built time-series models to monitor machine behavior on different time-scales, namely on a lot-to-lot, wafer-to-wafer, and real-time basis, and have used SPC techniques to detect faults [1][2]. The detection of an out-of-control condition by the fault detection mechanism 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 diagnostic system in a probabilistic framework is developed which will classify faults into discrete categories. In itself, the system serves as a tool to assist engineers in identifying problems affecting machine performance which could result in damage to the product, and as an early warning system to aid in scheduling preventative maintenance events, potentially reducing machine downtime. However, in the larger framework, this classification capability identifies modes of equipment operation, allowing us to utilize the appropriate blend of models best suited for that operation mode. Accordingly, wafer characteristics can be predicted based on a better estimate of the machine state.

EXPERIMENTAL SETUP

Data Description and Experimental Design

The monitored signals used in this work are those suspected to be most sensitive to changes in the chamber state of the etcher [2]. These signals are known as real-time tool signals and are collected while wafers are being processed at a rate

0-7803-3752-2 /97/$10

of 1 Hz. The changes we wish to detect and classify in this paper 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. There are five input settings which are varied over three levels according to a central composite design; this is summarized in Table 1. The design includes 36 runs with 9 centerpoints and is meant to cover a range of different faulty operating conditions. The purpose of this investigation is to determine a method for predicting 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.

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

Table 1. Input settings for the plasma etcher

Signal Selection

In this study, the probability of a *high, medium*, or *low* value for an input setting to a plasma etcher is determined using real-time tool signals collected from the plasma chamber as predictors. To determine a preliminary set of predictor variables to be used for modeling, boxplots are used to view the distributions of the real-time tool signals as a function of each input setting. Table 2 summarizes 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.

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

Table 2. Predictor variables for input setting responses

The signal selection, model construction and validation were implemented using S-PLUS software in an S-PLUS environment [3][4].

TREE-BASED MODELS

Description

Tree-based modeling is an exploratory technique which can be used to devise prediction rules, to select or screen 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 **x** into homogeneous regions, attempting to make the conditional distribution of the response **y** given **x**, **f(y/x)**, independent of **x**. The algorithm accomplishes this task by using a criterion minimizing a measure of deviance.

Classification trees are based on the multinomial distribution. If we consider a vector, for example, $y = (0,1,0)$, to represent the response **y** belonging to the second of three factor levels, then the probability corresponding to a response falling into each level would be given by $\mu = (p_1, p_2, p_3)$, with the constraint $\Sigma\, p_i = 1$, $i = 1,2,3$.

The model consists of a stochastic component given by

$$y_i \sim M(\mu_i),\ i = 1,2,\ldots,N$$

and a structural component

$$\mu_i = \tau(x_i)$$

The deviance is defined as minus twice the log likelihood

$$D(\mu_i; y_i) = -2 \sum_{k=1}^{K} y_{ik} \log(p_{ik}) \qquad (1)$$

and 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 compared by how well the partition corresponds to the true decision rule for problem. 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. The reader is referred to [3] and [4] for a more detailed discussion of this topic.

Model Construction and Validation

The data set was divided into two mutually exclusive sets by arbitrarily picking 12 runs out of the 36 to use as a validation set. Classification trees for each factor response (input setting) were then constructed from the training data of 24 runs using the preliminary set of predictors identified in Table 2.

An example of a classification tree built for the factor response ***gas ratio*** is summarized in Table 3. Although the model was originally given the predictors listed in Table 2, the tree algorithm selected RFTune and RFCoil as the most discriminating signals for classifying the gas ratio response.

The root at the top of the tree contains all of the observations, in this case there are 24 runs. Following the root are numbers corresponding to nodes where a split condition is defined, partitioning the space in a binary fashion according to whether the observations fall above or beneath the specified predictor split value. The terminal nodes are marked with an asterisk (*) and represent final values or predictions.

These nodes are also called leaves. Note that there is a distribution by class within each node giving the probability of observations belonging to each factor response level. The shaded boxes indicate the diagnosis which is based on the level with the highest probability value.

Node) Split Condition	Runs	P(High)	P(Low)	P(Mid)
1) Root	24	0.3750	0.1667	0.4583
* 2) RFTune<11797.7	9	0.7778	0.1111	0.1111
3) RFTune>11797.7	15	0.1333	0.2000	0.6667
*6) RFCoil<5655.53	8	0.0000	0.0000	1.0000
*7) RFCoil>5655.53	7	0.2857	0.4286	0.2857

Table 3. Classification tree for Gas Ratio Response.

A partition of the predictor space for the ***gas ratio*** response is displayed in Figure 1. The partitions are based on the simplified classification trees obtained after snipping unnecessary nodes. The resulting trees for each response span a domain defined by no more than two predictor variables, thus enabling the plots of the partitioned space.

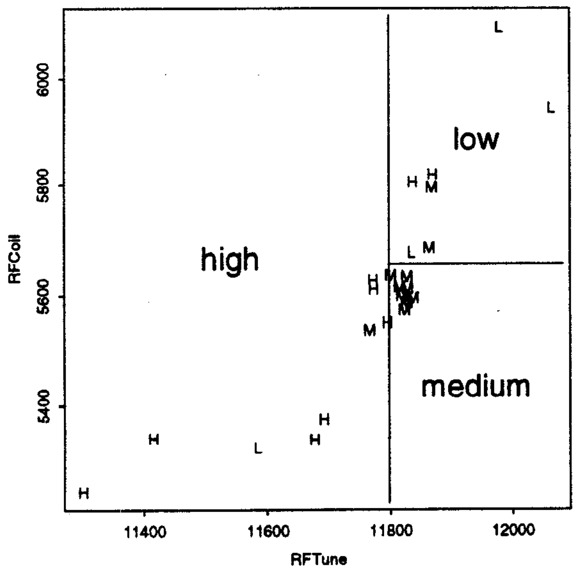

Figure 1. Partition for Gas Ratio Response

Response	Predictors	Misc	Valid
Pressure	DCBias, Power	0.1667	0.6667
RF Power	Endpoint A	0.0416	0.0833
Gas Ratio	RFTune, RFCoil	0.25	0.4167
Total Flow	MFC3	0.125	0.5
Gap Spacing	Endpoint C	0.0833	0.1667

Table 4. Summary of Classification Tree Results

Table 4 lists the predictors used in the final classification tree models for each response, along with a summary of the diagnostic results. The "Misc" column contains the misclassification rate, a measure of the model fit to the training data. The "Valid" column contains the misclassified points in the validation set. In general, reasonable models were obtained for the responses ***RFpower, total flow*** and ***gap spacing***, and these were tested by the validation set consisting of runs not used in building the models. However, the models for ***pressure*** and ***gas ratio*** performed rather poorly based on the validation sets.

GENERALIZED LINEAR MODELS

Description

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, $g(\mu) = \beta^T x$, where the linear predictor is the logit link function

$$\eta = \log\left(\frac{\mu}{1-\mu}\right) \quad (2)$$

or

$$\mu = \frac{e^{\eta}}{1+e^{\eta}} \quad (3)$$

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 $V(\mu) = \mu(1-\mu)$.

Model Construction and Validation

Two sets of models were built by encoding each factor response into a binary response. The first set was based on the high level as a "success" encoded with value "1", while the medium and low levels were grouped together as a "fa ure" and encoded with value "0". The second set reversed t high and low roles, with low being a "success" encoded w value "1", and medium and high together encoded as "0".

GLM models were constructed using the training data set 24 runs to predict the probability of success for each fac response. As in the building of classification trees, the line predictors for the models were chosen using the same set preliminary variables identified for each factor in Table2. Fo example, the form of the model fitted for ***RFpower*** is represented symbolically as

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

where μ is the probability of high ***RFpower*** for the first set of models. Table 5 shows the results of model building based on the training data for each factor response. A measure of goodness for the models was calculated using the formula

$$D_{\mu_0} - D_{\mu} \sim \chi^2_{p-q} \quad (4)$$

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. The interested reader is referred to [3] for a more thorough description of this test criteria. All of the models were found to be significant according to this test.

Response	High Misc	High Valid	Low Misc	Low Valid
Pressure	0.0000	0.0000	0.0000	0.0000
RF Power	0.0000	0.0000	0.0000	0.0000
Gas Ratio	0.2500	0.2500	0.1250	0.2500
Total Flow	0.0000	0.0000	0.0000	0.6667
Gap Spacing	0.0000	0.3333	0.0000	0.0000

Table 5. Summary of GLM Results

Model validation was conducted on the remaining set of 12 runs not used in building the models. The results are shown in Table 5. Perfect prediction (diagnosis) was obtained for the high and low levels for ***RFPower*** and ***pressure,*** high ***total gas flow,*** and low ***gap spacing***. The misclassification results are summarized in Table 5 for predictions of high and low responses respectively. The misclassification headings for the training and validation are the same as those in Table 4.

DECISION TREE ARCHITECTURE

The two modeling techniques for diagnosis examined in this paper can be combined in a decision tree architecture which makes use of the conditioning and partitioning of the input space in the classification trees, and the greater flexibility for modeling probabilities provided by the GLM's. Specifically, it is worth noting that the performance of the classification trees could be vastly improved if the partitions were not constrained to being constant functions in a single variable. Using the logit link function to model the probabilities alleviates this constraint. Similarly, because the GLM's are fit to a single binary response, their performance as predictors could be improved by conditional knowledge of the operating space (knowledge of the other responses). This could be achieved using the natural hierarchy provided by a tree-based model to partition the input space. For further exploration of these ideas, the interested reader is referred to [5].

SUMMARY AND FUTURE WORK

Classification trees are shown to be effective in predicting changes in the input settings using only a small subset of the real-time tool signals. In all cases, the trees are reduced to operate on a space defined by at most two predictor variables (real-time signals), without an increase in the misclassification rate. Unfortunately the performance of these models is highly dependent on the data set used to train them. In our case, with such a small data set, the results of the diagnosis varied substantially from one factor response to another.

Our investigation of GLM's for modeling binary response data shows that the increased flexibility in this modeling technique can lead to promising diagnostic results. Again the models can only be improved with a larger data set.

We plan to expand our study to include data taken from manufacturing machines which have been known to exhibit real problems. This will test our assumptions about using designed experiments as simulations for faulty conditions, and will also provide a real forum to test our methodology. We also expect improvements in diagnostic performance as we iterate through and combine various models, using their respective strengths to complement each others pitfalls.

CONCLUSIONS

Automated diagnosis of faults will provide a systematic method of drawing inferences from the available evidence, while accounting for uncertainty by retaining a measure of likelihood for each classification decision. The architecture under investigation gives structure to the problem and deals naturally with its inherent complexities. This is accomplished by successively dividing the input space into operating regions defined by fault categories and keeping track of independence assumptions. Future work includes automating the calculation and updating of the model parameters and incorporating the models through the use of a hierarchical mixture of experts (HME) architecture [5].

A diagnostic 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. When implemented and used in conjunction with good engineering practices, this tool provides a means of mastery over the increasingly overwhelming amounts of data. By looking into the future and anticipating the needs generated by the advancement of technology, we set our research goals to advance the state of the art in manufacturing tools which brings us that much closer to automation and better control in the fab.

ACKNOWLEDGEMENTS

The authors are grateful to Texas Instruments, Lam Research, Motorola, Triant Technologies, National Semiconductor, Digital Semiconductor, and to the SRC (97-FP-700) for support of this research.

REFERENCES

[1] A.M. Ison, C. J. Spanos, "Robust Fault Detection and Fault Classification of Semiconductor Manufacturing Equipment", ISSM96, Tokyo, Japan, October 2-4 1996.

[2] 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 Trans. Semiconductor Manufacturing, vol.8, no. 1, Feb. 1995, pp. 17-25.

[3] W. N. Venables, B. D. Ripley, *Modern applied statistics with S-Plus*, Springer-Verlag, NY, 1994.

[4] J.M. Chambers, T.J. Hastie (ed.), *Statistical Models in S*, Chapman & Hall, London, 1993.

[5] M.I. Jordan, R.A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm", *Neural Computation, 6*, pp. 181-214, 1994