A Study on Failure Prediction in a Plasma Reactor

Edward A. Rietman, *Member, IEEE,* and Milton Beachy

***Abstract*— We use several approaches to demonstrate that neural networks can detect precursors to failure. That is, they can detect subtle changes in the process signals. In some cases these subtle changes are early warnings that a subsystem failure is imminent. The results on detection of precursors and faults with various types of time-delay neural networks are discussed. We also measure the noise inherent in our database and place bounds on neural network prediction in the presence of noise. We observe that the noise level can be as high as 40% for detection of failures and can be at 30% to still detect precursors to failure. We note that although self-organizing networks for classification of faults seems like a good idea, in fact they do not perform well in the presence of noise. Lastly, we show that neural networks can induce, or self-build, Markov models from process data and these models can be used to predict system state to a significant distance in the future (e.g., 100 wafers).**

***Index Terms*—Failure prediction, neural networks, plasma reactors, time series analysis**

I. Introduction

THE ABILITY to flag and correctly identify subsystem failures within semiconductor processing equipment could have a significant impact on throughput. For example, consider the scenario, a plasma reactor's own internal diagnostic flags a pressure problem and the reactor is shutdown. Since the machine flagged the fault as a pressure problem, the maintenance technicians will investigate the relevant subsystems, maybe change a gasket or O-ring and then bring the machine up for production. Two hours later, the system is down again for an rf-electrode error. Now the technicians find the actual problem is a loose wire in the matching network. In the first case, the facility was brought down for a pressure problem. In fact, the rf-problem manifested itself as a pressure problem. By monitoring many *in situ* process signals and signatures, we could have flagged the correct subsystem fault thus improving throughput in the entire facility by eliminating down-time for trouble shooting.

We would like to be able to not only correctly identify causes of failure in processing equipment, but we would also like to predict, as far in advance as possible, that a specific type of failure event will occur, and we would like to have high confidence on this prediction. Our goal was to carry out feasibility studies for this supposition. In order to test these ideas, we used data from one plasma reactor in our Orlando factory. In this paper we report on several approaches to attacking this problem. We used a neural network to look at summary data. We also used neural networks to examine huge multidimensional time-series of process signals, and we discuss the limits on the ability of neural networks to operate on time-series with additive noise. The next part of the paper discusses the plasma reactor and the data samples. This is followed by a discussion of some theoretical issues involved in the modeling. The core of the paper discusses the results, which is followed by a discussion and conclusions section. Our conclusion, briefly, is that the prediction is possible even in the presence of 30% noise. This prediction can be up-to dozens of wafers in the future depending on the type of data used.

Manuscript received November 5, 1997; revised February 2, 1998.

E. A. Rietman is with Bell Labs, Lucent Technologies, Murray Hill, NJ 07974 USA (e-mail: ear@bell-labs.com).

M. Beachy is with Lucent Technologies, Orlando, FL 32819 USA.

Publisher Item Identifier S 0894-6507(98)08360-2.

II. Literature Survey

The most common failure diagnosis system consists of establishing a upper and lower limit for some monitored signal. If the signal exceeds these bounds it is said to be a fault. The vast majority of the literature on the subject focuses on fault identification not fault prediction. A sampling of the literature includes Markov models (cf. [1]–[3]), and neural networks (cf. [4]–[7]).

Isermann [8] wrote an excellent review article on fault detection based on modeling and system estimation. The basic idea is that if we have a good model of the process we can compare current process signatures and outputs with those from the model. If we detect values above or below some threshold this could indicate that a fault is imminent. The problem is that often we do not have any models let alone a good model. So, one attempts to induce a model using time-series analysis. The classical approach is to build autoregressive (AR), moving average (MA) and variations, such as ARMA (cf. [9]). Often associated with the ARMA approach is the cumulative sum (CUSUM) of the residuals method to identify faults (cf. [10]).

Usually, these models are inadequate because the real-systems generate multidimensional cross-correlated signals. In order to circumvent these problems, the usual approach is to resort to control charts, in what is known as, statistical quality control (cf. Duncan [11] and Montgomery [12]). Here the problem is that control of the process and fault identification are *a posteriori* and the end result is compared with the target specifications and the failure event has already occurred. In a manufacturing environment, that may mean that low quality product has already been manufactured.

With respect to plasma reactors, for fault diagnosis, there are three key papers: May and Spanos [13], Baker *et al.* [14], and Kim and May [15]. The May and Spanos paper discusses monitoring various process signatures in real-time and incorporating these with equipment maintenance history data and in-line metrology measurements of the produced

wafer. These three pieces of information are incorporated into probabilistic models, similar to [16], to deduce the type of failure event. Although the model is capable of operating without the metrology data, we suspect in that case the quality of the prediction would be degraded.

In a very similar and very innovative experiment, Kim and May [15] assembled neural network modules for the on-line diagnosis and in-line metrology. The maintenance module consisted of a neural network to compute the parameters for a Weibull distribution representing the cumulative distribution of failures. Using the T^2 statistic they report very good results at failure identification in real-time. The difficulty with failure identification in a plasma reactor is the fact that many of the process signatures are strongly correlated by direct mechanical or electrical coupling. For example, if there is a sudden vacuum failure, causing the pressure to increase rapidly, the plasma will be quenched. This will cause a failure in the dc bias, and reflected rf power. The applied rf power will try to compensate. All these events will happen, more or less, simultaneously. In this example, if we could have examined precursors to failure in an MFC we would have diagnosed the correct failure and prevented the self-shutdown of the plasma reactor.

The paper by Baker *et al.* [14] uses a different approach. Here, the authors describe a time-delay neural network to model the process dynamics with the goal of identifying process signals that are greater than some preset threshold. Their results indicate that the neural network is able to detect errors. They demonstrate the ability of time-delay neural networks (cf. [17]) to capture the dynamics of the process from process signatures. Of more significance, they demonstrate that small fluctuations in the process signals may be precursors to process faults. They also suggest that the neural net method may be superior to conventional ARMA time-series models because the neural network removes auto-correlations and cross-correlations from the inputs. One criticism of their work is that the sampling interval for their data collection was 50 Hz. So their prediction ahead, up to ten time units, looses some of it's significance. As does their observation that small fluctuations are precursors to process faults. Significant events do not occur that quickly in the plasma reactor. Although the faults observed by Baker *et al.* were real, one could still question the time-scale for precursor events. That is, how soon can we observe precursors? The question remains, are there fluctuations that act as precursors to faults and can neural networks detect these small changes? These questions are the primary focus of this paper.

III. Reactor, Process, and Databases

The plasma reactor used in our study was a Drytek Quad Reactor (Drytek was acquired by Lam Research). The facility is a cluster tool with four plasma etch chambers. Although we collected data for all four chambers we focus our study on chamber 1. The process was an etch process involved in defining the location of the transistors on silicon wafers and is a three step process, involving different etch conditions. During the processing of wafers, several machine signatures are monitored in 5-s time intervals. The monitored signals include: pressure, flow rate of four gases, dc bias, rf applied, and rf reflected. These signatures are stored in a buffer, along with a time stamp (relative to start of that step). At the end of the etch process the data in the buffer are written to an ASCII file, via the SECS interface, to a UNIX host computer. When this occurs there is a UNIX time stamp associated with the file. The file name has associated with it a lot number and slot number. The file contains the concatenation of the data for the three etch steps. Additional data from these time streams were written to an SQL database as a statistical summary of the time-series. The mean value and standard deviation of each of the process/machine signatures was recorded in a table along with a time stamp for when this occurred.

TABLE I
Mean Time Between Failures

type of event	number	MTBF
total	231	4.7 days
transport	55	2.9 weeks
rf and dc	49	3.3 weeks
pressure	45	3.5 weeks
slot valve	32	5 weeks
particles	27	5.9 weeks
water flow	15	10.6 weeks
software	4	40 weeks
pumps	4	40 weeks

Our studies have been carried out over the last year and data were collected off the production databases as needed. We collected maintenance data from January 1, 1992 to April 1, 1995 and these data were used in the "hard fault" counting. A second study involved counting "soft faults" from the SQL summary database. These data were from November 1, 1995 to March 31, 1996 and can be assembled into a sequential time series representing the state of the plasma reactor for each step for each wafer.

A third study involved data collected from May 1, 1997 to July 31, 1997. For this time period, we have complete time-stream (every 5 s during etch). These data were also assembled into a large sequential time-stream representing the entire activity of the reactor. In each case the collected data represent fluctuations around the set point. In as much as these are the data available in real-time on the plasma reactor the question arises, how reasonableare these data for modeling? Can we expect to observe indicators prior to a subsystem failure? May and Spanos [13] consider a simple example of the mass flow through a pipe. If F_1 represents the flow into the pipe and F_2 the flow out of the pipe, by conservation of mass $F_1 - F_2 = 0$. If the flows into and out of the pipe are monitored, a violation of this conservation law indicates either a leak or a sensor failure. So, by analogy of the pipe system with the mass flow controller (MFC) of a plasma reactor, by monitoring fluctuations about the set point of the MFC we may determine if a failure has occurred or if a failure is imminent. By extension we might be able to determine failures

TABLE II
PEARSON CORRELATION MATRIX

	MFC1	MFC2	MFC3	MFC4	rf appl	rf refl	press	dc bias
MFC1	1.0							
MFC2	-	1.0						
MFC3	-	-0.440	1.0					
MFC4	-	-0.087	0.502	1.0				
rf appl	-	0.445	-0.225	0.101	1.0			
rf refl	-	0.358	-0.201	-0.015	0.431	1.0		
pressure	-	-0.281	-0.280	0.036	0.460	0.104	1.0	
dc bias	-	0.530	0.235	0.223	0.694	0.364	-0.147	1.0

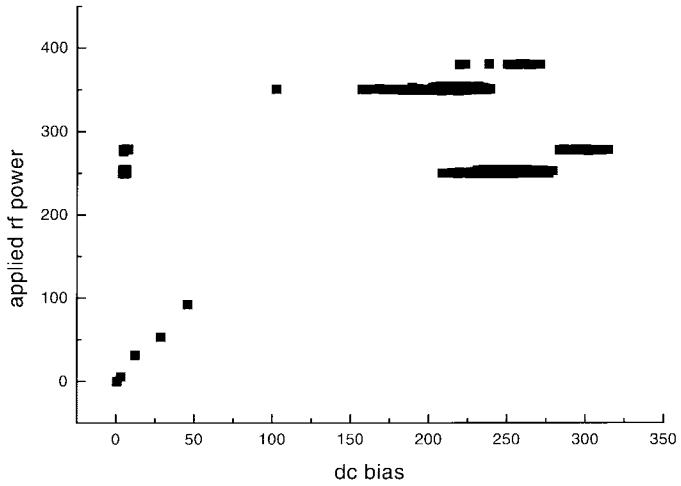


Fig. 1. Scatter plot of dc bias and pressure. The two variables are strongly correlated (0.694) but, the plot would be of little value for predictive modeling.

in the impedance matching network or in the pressure system by monitoring rf-applied, rf-reflected, dc-bias, and pressure, respectively.

Using these sources of data, we define "soft faults" as those in which the operator can quickly surmise the situation and make amends. An example of this would be a temporary malfunction in a load-lock chamber door resulting in the system not pumping down to base pressure. Faults of this type would be found in the time-stream data but, not in the maintenance database, since no repair technician was called. When the maintenance technician is called, something more significant needs attention and these system faults we classify as "hard faults." It is important to realize that the number of soft faults could be an order of magnitude more frequent than hard faults and "hard faults" would occur in the time-stream database, but not necessarily in the maintenance database.

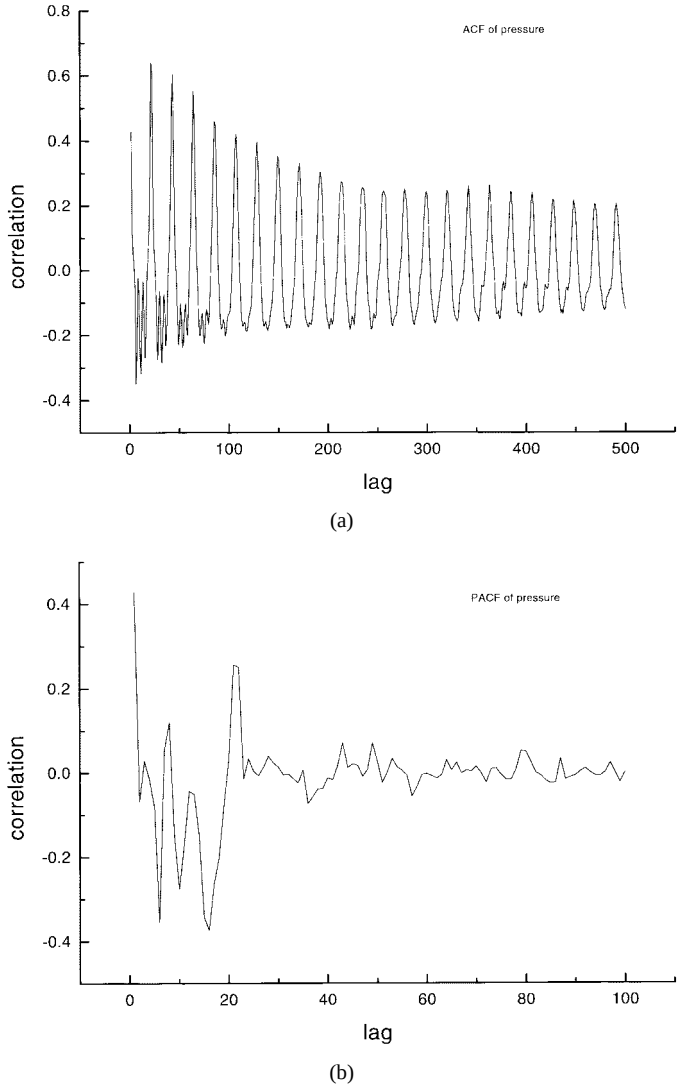


Fig. 2. (a) Plot of autocorrelation function for pressure and (b) plot of partial autocorrelation function for pressure.

IV. SIMPLE COUNTING MODELS

Simple frequency models can be computed from counting failure events. We do not have a measure of the actual distributions. However, the number of observations are so large (>140 000, processing steps or 46 000 wafers) that they suggest a degree of reliability based on the law of large numbers (cf. Shiryaev [18]), but we should also add the caveat that some of the failure events could also be classified as rare events. For example, the results suggest a vacuum pump failure once every 40 weeks and a pressure failure every 21 000 wafers.

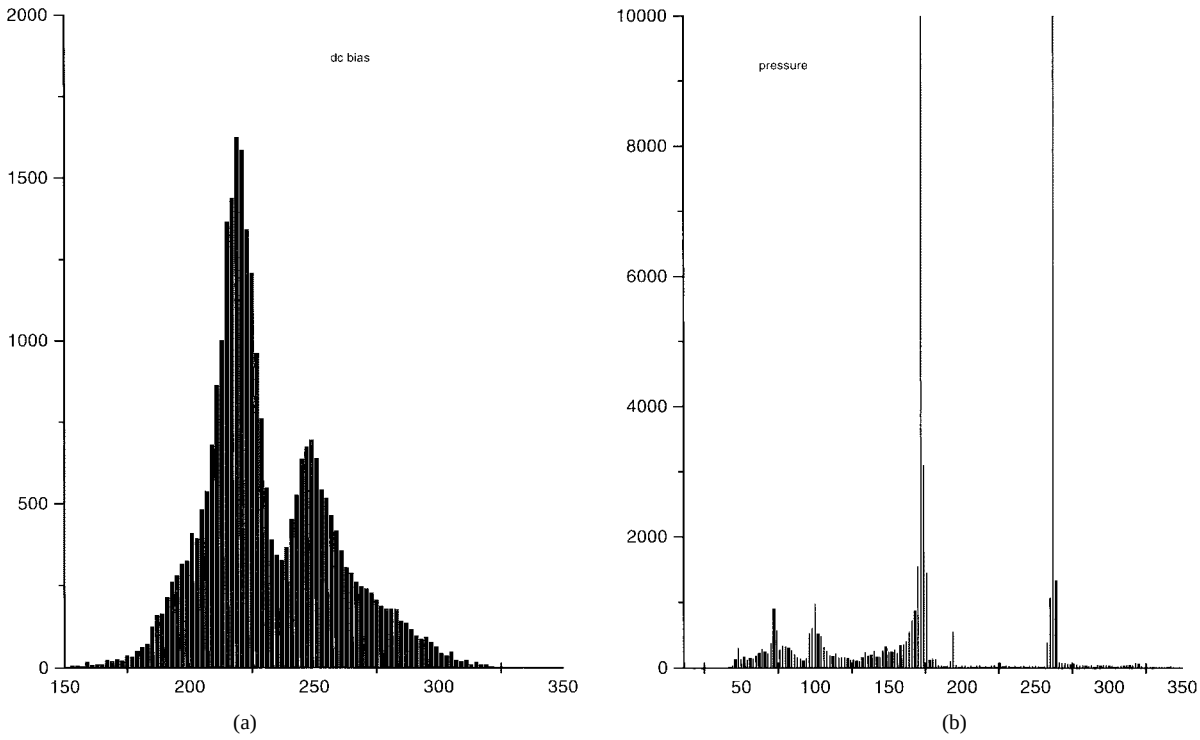


Fig. 3. (a) Histogram for dc-bias and (b) histogram for pressure.

From the maintenance database, the types of errors or faults that took the facility down for repair were lumped together to reduce the number of types of faults. This provided us with a gross or high-level view of the problems with the facility. The classifications were as follows. Faults involving the transport arm, the loader and the cassette chamber door not closing were classified as "transport" errors. All matching network problems, applied RF, reflected RF, and dc-bias problems were classified as "rf-dc" errors. Where the word pressure occurred we classified those errors as "pressure." Of course, the pressure problem may have been in a reaction chamber, the cassette chamber, or somewhere else. Leaky gaskets and O-rings were also classified as "pressure." Specific slot valve problems were classified as "slot valve." Where stated particle problems were classified as "particles." Water flow errors, electrode cooling problems, and chiller problems were classified as "water flow" errors. Specific software and computer related problems were classified as "software" and vacuum pump errors were classified as "pump" errors.

In three-and-one-half years, there was a total of 231 faults reported by the staff and entered in the maintenance databases. Transport faults are the major cause for system malfunction. This, of course is reasonable, being a mechanical system. This suggests that these systems are not as robust as we would like. About 24% of the faults were related to transport. Another 40% of the faults were attributed to matching network and pressure, with each of them accounting for 20%. In conversations with the engineering staff, the major problems that come to mind are software and pump-related. Yet in the three-and-one-half years of recorded hard faults, only four were due to software and four due to vacuum pumps.

The mean time between failures (MTBF), for specific types of events, were obtained from simply dividing the number months by the number of events. Table I presents our results on MTBF based on the maintenance databases. From this table, we can expect one failure event per week. (This reactor is over 10 yr old and still in production for noncritical etch processes.)

A second counting model was built with the data from our time-series summary statistics consists of mean values and standard deviation for each of the process signatures, for each process step of each wafer. (Recall, three process steps per wafer.) Since there are accurate time stamps in these data, it is possible to align the statistical summaries sequentially in time-series consisting of >140 000 values. The period of study was from November 1, 1995 to March 31, 1996. Unfortunately this time period did not overlap with data from the maintenance database. Failure events that deviated by 4 or more standard deviations from the mean, for a process variable, were classified as failure events. Based on these observations failures will occur every 9000 wafers. With a mean time between failures of 9000 wafers we can use the Poisson distribution to compute the probability of failure after processing n wafers. The cumulative distribution function is given by

$$f(x) = 1 - \exp\left(-\frac{x}{9000}\right).$$

So, if $x = 4500$ wafers that have already been processed, the probability of a failure now is about 0.4.

V. AR Models

Prior to attempting the neural network modeling we did a Pearson cross-correlation study and classical AR modeling.

Table II is shows the cross correlations. As can be expected there is a strong correlation (0.694) between the rf-applied and the dc-bias. Also there is a strong correlation (0.53) between one of the MFC's and the dc-bias. However, strong these may be, there is little predictive capability, as shown in Fig. 1, for scatter plot of dc-bias and rf-applied. The cause of this high correlation, but low predictability, as shown in the AR models, is likely do to the fact that the actual subsystems, (e.g., applied rf and dc-bias) are electrically related. A similar observation can be made about other pairs of variables. If the pressure suddenly increases due to a vacuum leak, the rf will show a significant variation.

A typical AR model involves autocorrelation. Consider N observations of a discrete time series. We can form $N-1$ pairs of observations of the type $(z_1, z_2), (z_2, z_3), \cdots, (z_{N-1}, z_N)$. The correlation coefficient can be written by regarding the first observation in each pair as one variable and the second observation as another variable. We can write the autocorrelation function (ACF) as

$$\rho_k = \frac{\text{Cov}(Z_t, Z_{t+k})}{\sqrt{\text{Var}(Z_t)}\sqrt{\text{Var}(Z_{t+k})}}$$

where Cov is the covariance and Var is the variance between two samples in time. The partial autocorrelation function (PACF) is given by

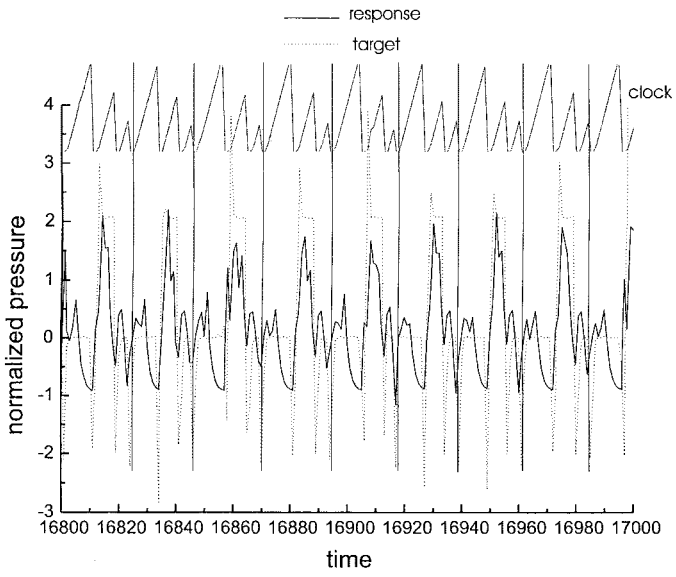
$$P_k = \frac{\text{Cov}[(Z_t - \hat{Z}_t), (Z_{t+k} - \hat{Z}_{t+k})]}{\sqrt{\text{Var}(Z_t - \hat{Z}_t)}\sqrt{\text{Var}(Z_{t+k} - \hat{Z}_{t+k})}}$$

where

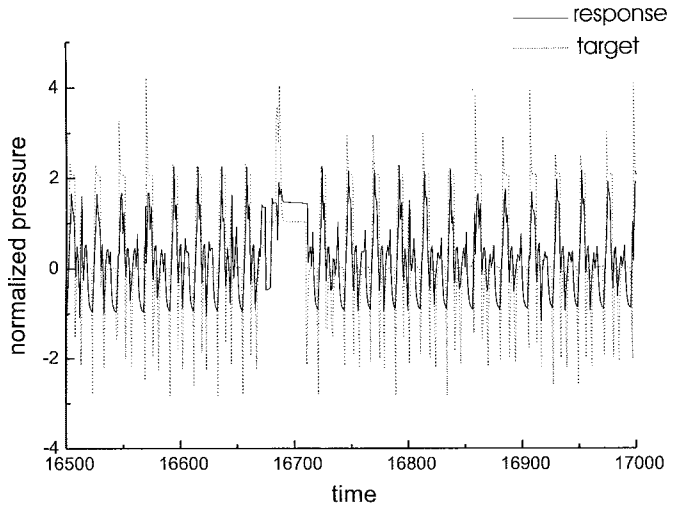
$$\hat{Z}_t = \beta_1 Z_{t+1} + \beta_2 Z_{t+2} + \cdots + \beta_{k-1} Z_{t+k-1}$$

and $\beta_i (1 \le i \le k-1)$ are the mean squared linear regression coefficients obtained by minimizing the expectation $E(Z_t - \hat{Z}_t)^2$. More details on the ACF and PACF are given by Wei [9]. Graphical plots of these two functions are shown in Fig. 2 for pressure. Notice that there are oscillations in the ACF plot. Oscillations indicate that the correlation is oscillating about the mean, just as the signal itself does. At further distances the correlations drop off to near zero. Similar plots were generated for the other variables to confirm that the process is an AR(30) process. Further modeling with ARMA variations is unwarranted since the system of interest is multidimensional and cross-correlated (cf. Wei [9]).

Lastly, it is known that AR(p) processes are Markov models or Markov state machines (cf. [19], [10], [20]). This is the cause for the high correlations between the variables. If we know the current state of the machine we can predict the next state with high confidence. But that has little predictive capability for predicting subsystem faults. Further, because of noise in the data our prediction of the next state may be in error. In the next section, we show that neural networks can be used to build Markov models from the data.



(a)



(b)

Fig. 4. (a) Prediction and target for pressure. This shows the Markov model induced from the database by training the neural network and (b) similar to Fig. 4(a), but includes an actual pressure failure event.

VI. Neural Network Model Based on Full-Time Streams

Neural networks are a nonlinear mapping technique, claimed to be modeled after the biological neural networks. The mathematical foundations go back to the turn of the twentieth century and focus on a conjecture by Hilbert [21]. He conjectured that it would not be possible to solve the following seventh-degree polynomial:

$$f^7 + xf^3 + yf^2 + zf + 1 = 0$$

with continuous functions of only two variables. Kolmogorov [22] refuted Hilbert's conjecture and showed that it is possible to represent continuous functions of many variables by superpositions of continuous functions of one variable, which is of the same form as neural networks. The output of a neural

network r is given by

$$r = \sum_j \left[W_{jk} \bullet \tanh\left(\sum_i W_{ij} \bullet x_i \right) \right].$$

This equation states that the ith element of the input vector x is multiplied by the connection matrix W_{ij}. This product is then the argument for a hyperbolic tangent function which results in another vector. This resulting vector is multiplied by another connection matrix W_{jk}. The subscript i spans the input space. The subscript j spans the space of hidden nodes and the subscript k spans the output space, in our case $k = 1$. The connection matrices are found by gradient search of the error space with respect to the matrices. The cost function for the minimization of the output response error is given by

$$L = \left[\sum_j (t - r)^2 \right]^{1/2} + \gamma \|W\|^2.$$

The first term represents the r.m.s error between the target, t and the response r. The second term is a constraint that minimizes the magnitude of the connection weights W. If γ (called the regularization coefficient) is large, it will force the weights to take on small magnitude values. This can cause the output response to be small. For example, if the weights are all small numbers then the product of these small numbers times the output from the hyperbolic tangent will also be a small number. Summing these small numbers over the span of the hidden unit space will result in a small number. Given this weight constraint, the cost function will try to minimize the error and force this error to the best optimal between all the training examples. The effect is to strongly bias the network. We make use of this fact later in an example. Details of neural networks can be found in [23] and [24], and details of learning with constraints can be found in [25]. In the interest of being brief, we focus on pressure or dc-bias for our examples of the modeling. Fig. 3 shows the histograms for both of these variables. Noise in the data results in the dc-bias appearing as a bimodal distribution and noise causes the pressure histogram to appear as a tetramodal distribution. Based on the recipe, the pressure should be bimodal and the dc bias trimodal. This is an indicator that the desired modeling will be difficult.

Time segments of all the process signatures (e.g., flow rate, pressure, rf, etc.) will be used as inputs to the neural network and pressure will be the output. The first model was based on using the 5 s interval time-streams from the plasma reactor. The input was a sliding window of 25 time samples (about one full wafer) for each of the process signatures and the output was pressure n time units in the future. The input was changed by sliding the window one time unit in the future. The neural network was often trained for 15 000 samples and tested on 30 000 samples. But the actual data file, for training and testing, consisted of over 60 000 samples. The total number of inputs was 226. This is computed from 25 time samples of nine process signatures (four gases, rf applied, rf reflected, pressure, dc bias, "time") and one input bias. The "time" input was the total number of seconds since the start of the etch. The network had 60 hyperbolic tangent nodes in one hidden layer and one linear output node. Thus the network was a 226-60-1 architecture with 13 620 connections or adjustable parameters in the two connection matrices. Although this may seem like a large number of connections, given that there were a far more

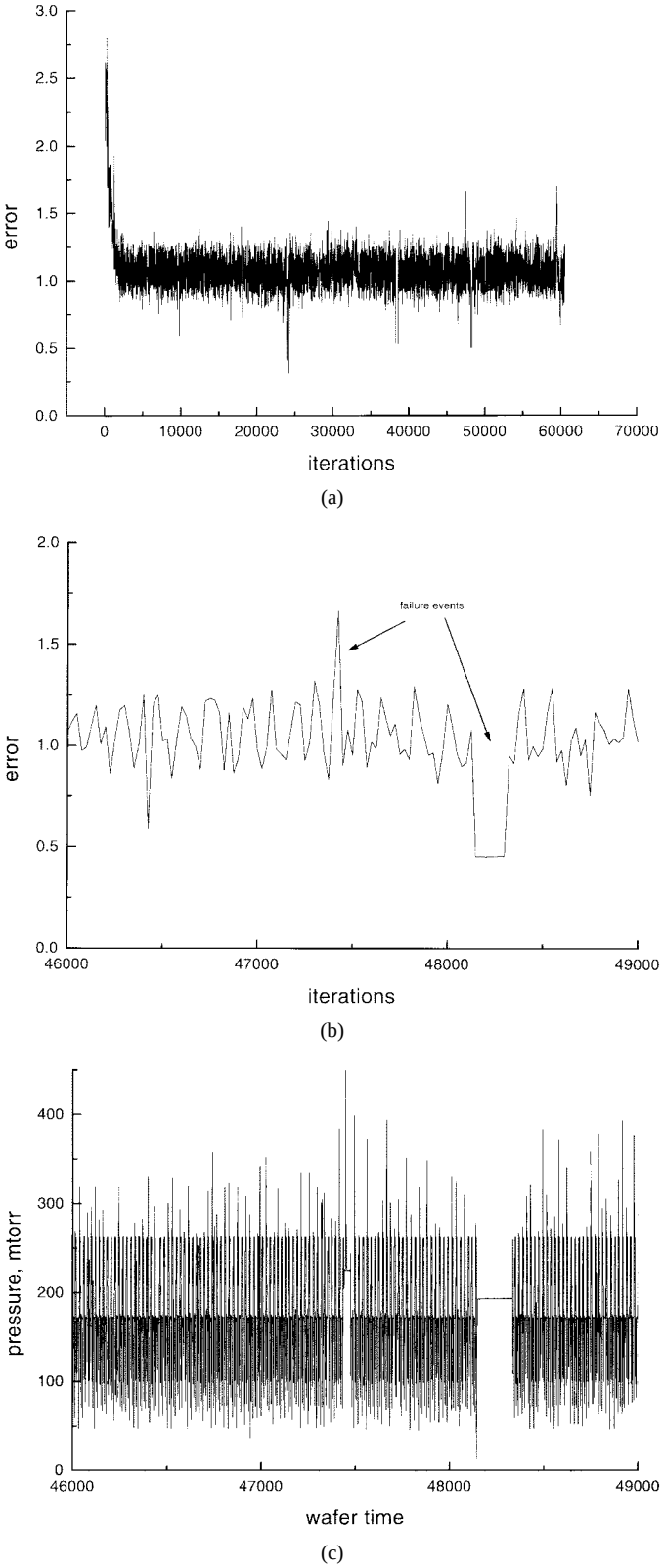
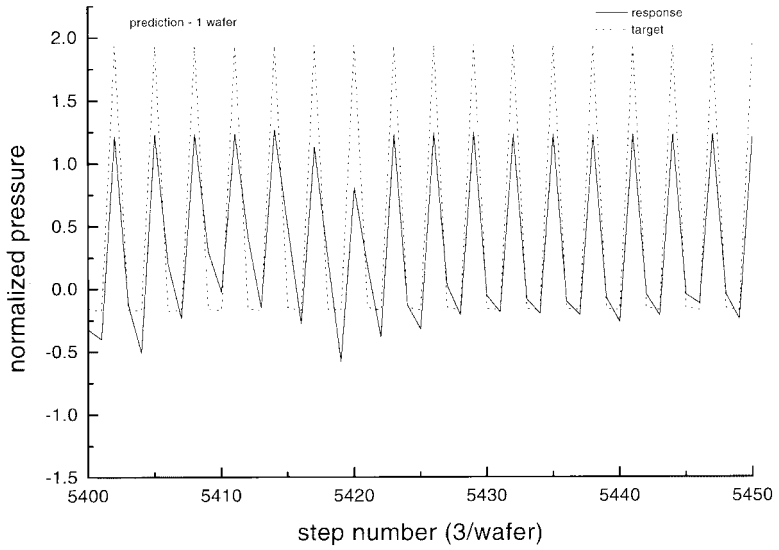
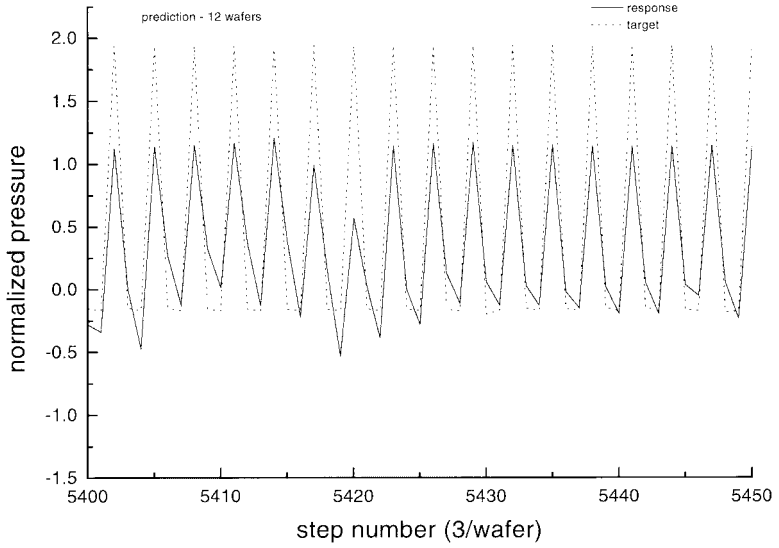


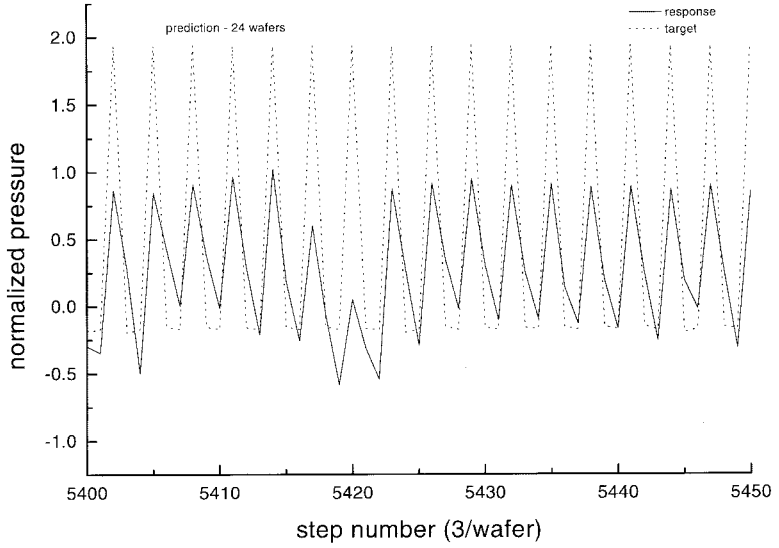
Fig. 5. (a) The full neural network learning curve, (b) expanded region of the learning curve showing two failure events, and (c) actual pressure data showing the same failures as detected in (b).



(a)



(b)



(c)

Fig. 6. (a) Pressure prediction one wafer in the future, (b) pressure prediction 12 wafers in the future, and (c) pressure prediction 24 wafers in the future.

training samples and that weight minimization was used, there was no over training observed. A prediction of five time units is equivalent to predicting 25 s, or one wafer in the future, this resulted in a pressure prediction error of 44 mtorr. A prediction of ten time units, two wafers, resulted in a prediction error of 57 mtorr and 100 time units, 20 wafers, a prediction of almost an entire cassette into the future, gave a prediction error of 63 mtorr. For comparison the standard deviation of the pressure (across the three steps) is 64 mtorr.

Fig. 4(a) shows the prediction and target for pressure. Also included on this figure is the clock signal to show the start of each step for each wafer. Fig. 4(b) shows the same type of plot but includes an actual pressure fault. In both of these figures the prediction was only one time unit in the future or 5 s. Furthermore, in these figures, the training was not stopped. So this shows the ability of the neural network to adapt to changes in the actual stream of process signals and to detect process faults. But, alas the prediction is not very far in the future. So the fault was not anticipated by the neural network. This can also be seen in the learning curve (i.e., r.m.s error versus training iterations). If the adaptive ability (i.e., the learning) is not disabled and the regularization coefficient is not too small the network will quickly converge to an r.m.s error of about 1.0. When there is an actual fault in the pressure this shows up in the learning curve as a large positive or negative error. Fig. 5(a) shows the entire learning curve and Fig. 5(b) shows an expanded segment of the learning curve. The comparable region of the actual pressure data is shown in Fig. 5(c) between 46 000–49 000 time units. An actual pressure failure occurred between 48 100–48 200. The neural network was able to detect this fault. It should also be noted that, estimating graphically, from Fig. 5(c), the noise spikes are almost 40% of the signal intensity. This is an issue we will revisit in a later section.

VII. Nontime Delay Neural Network Model Based on Statistical Summary Data

Building a model from the statistical summary data has the advantage that the mean and standard deviation will have less noise than the raw data. The statistical summary will thus act as a filtering tool prior to processing. This model was based on statistical summary data for the process signatures for 3000 wafers processed in sequence. At the end of each etch step, in the three step process, the mean value and standard deviation of the observed signals are written to a data file. So one wafer of data would consist of end-time (time called by the endpoint detector) total-time (included any over etch time), step number, mean, and standard deviation of four gases, rf applied, rf reflected, pressure, and dc bias. These eighteen numbers were the inputs along with etch time, a recurrent connection from the output and network bias giving a total of 21 inputs. The hidden layer consisted of five hyperbolic tangent units. The following experiments were done with a window of one wafer. The network had a 21-5-1 architecture with 110 connections. The input was not a sliding window with time delay, but rather summary data for individual steps followed by the next step of data.

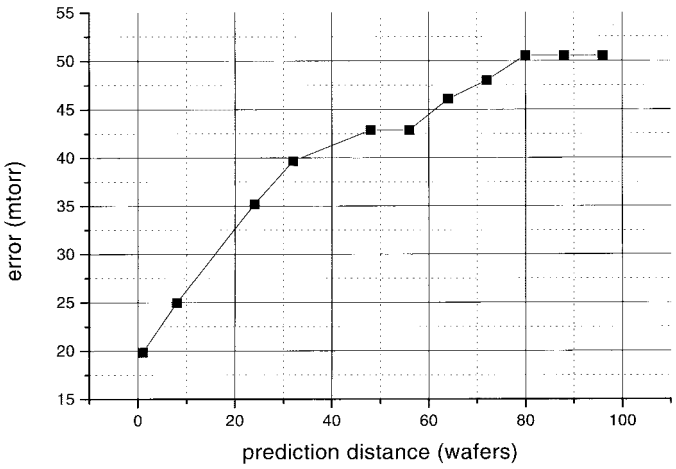


Fig. 7. Error curve showing prediction distance.

The above neural network prediction of pressure, for one, 12, and 24 wafers ahead is shown in Fig. 6(a)–(c). Fig. 7 shows prediction error up to 96 wafers into the future. The prediction is smoothly degraded till about 80 wafers where the error saturates at 51 mtorr. For comparison, the time delay network that used the entire time stream of data, not the summary statistics, had a prediction good to within 63 mtorr, at 20 wafers, and the actual pressure had a standard deviation of 64 mtorr. A reasonable speculation is that the neural network is inducing a Markov model from the time series data. Another possibility is that the neural network is building an associative memory of the vectors of wafer data.

So the prediction is not really too bad. It is unreasonable to predict this far into the future unless the neural network is inducing a Markov model from the time series data. Consider the following, we have a plasma tool that is always used for the same etch process and there are three steps in the process. Than if the current state of the tool is step 2 we can predict with high confidence that the next step will be step 3, the next step after that will be step 1, followed by step 2 again, etc. But, if there is a failure at one of these steps our prediction, based on the current step, will have a probability less than 1. However, not knowing in which step the failures will occur, our first guess would be that there will be no failure and so our prediction will still have a high confidence, albeit less than 1 probability. Thus, if we know the current state we can predict future states with high confidence; a simple Markov chain.

Markov chains can also be constructed that use a time segment of past information or a sliding window of state information [32]. A generalized Markov chain assumes a particular state is dependent on only n previous states. In this case it is necessary that N, the length of the entire chain, be $N \gg n$. So a time series may be viewed as a generalized Markov chain.

VIII. Time Delay Neural Network Model Based on Statistical Summary Data

The study with surrogate data showed that neural networks can be used to observe small fluctuations if the data stream is not too noisy. For the current experiment we conjectured

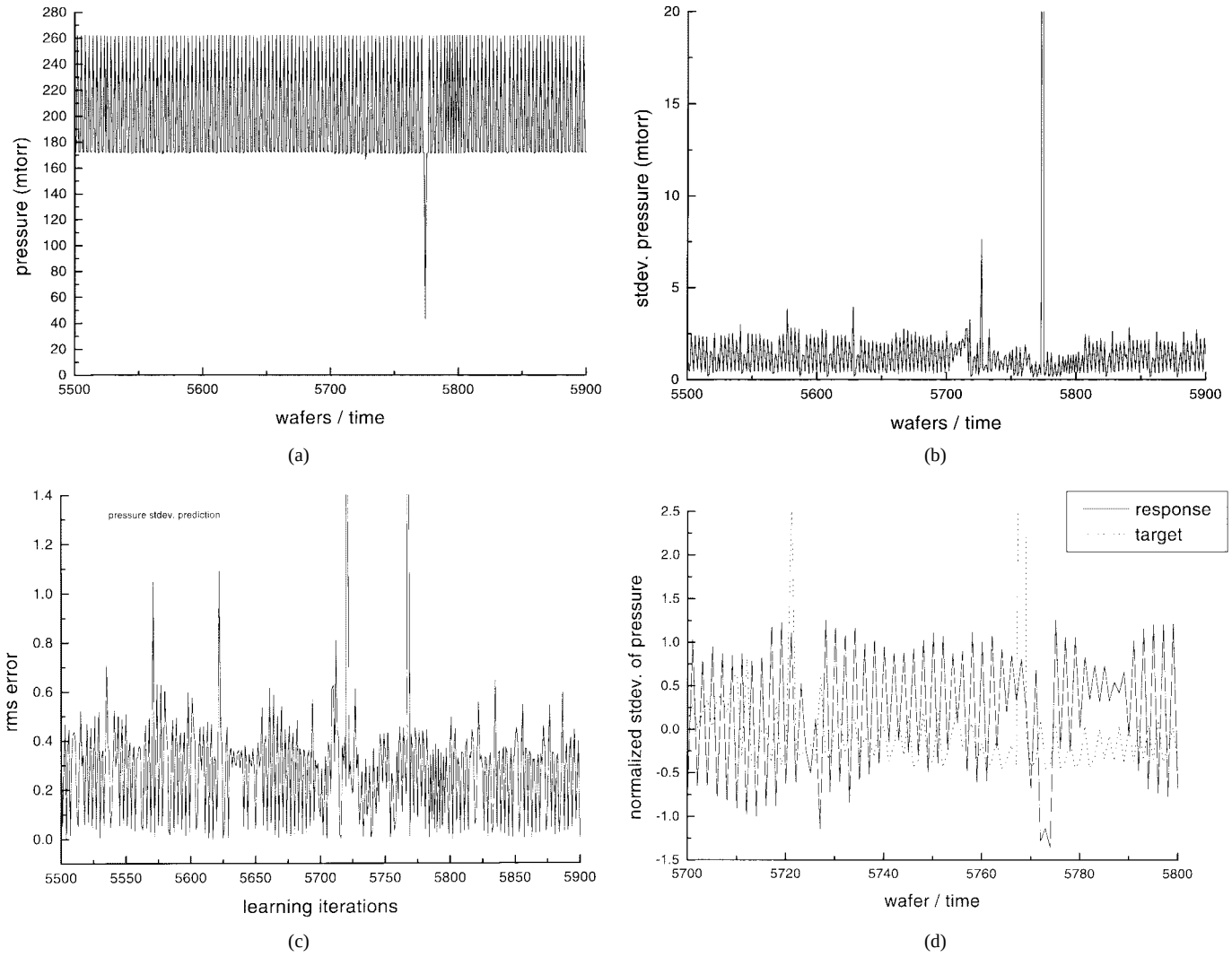


Fig. 8. (a) Mean value of pressure between 5500 and 5900 samples. A failure can be seen but no precursors to the failure are seen in the mean value data and (b) standard deviation of pressure for same time segment. Here precursors are seen at about 5700 and the failure at 5775. The precursors thus show up at about 12 wafers prior to the actual failure event. (c) Segment of neural network learning curve showing the detection of the precursors shown in (b) and (d) target and response curve for the same neural network predicting pressure.

that a time delay neural network, operating on the statistical summary data might be able to observe precursors if the neural network input consisted of a time delay segment of data and only one process signal. Initially we selected to examine the standard deviation time streams. We surmised that fluctuations in these signatures would be more indicative of actual precursors to failure and that it may be possible to disregard the noise associated with the full set of inputs, by examining single process signatures.

A time delay neural network of five delay units, one current time unit, one recurrent unit from the output, and one bias unit were the inputs. The network had one hidden layer of five hyperbolic tangent units and one linear output unit. So the network with (8-5-1) architecture had 45 connections.

Fig. 8(a) shows the time stream for pressure. The data in this figure are the mean value of pressure at each of the processing steps for the three step process. There is a pressure failure clearly seen at 5770. Fig. 8(b) shows the corresponding standard deviation time stream. The failure at 5770 is seen as well as precursors starting at 5710 and a strong signal at 5725. This strong signal is 15 wafers, prior to the actual failure event at 5770. Fig. 8(c) is an expanded region of the neural network learning curve for prediction of the standard deviation of pressure for a prediction distance of one wafer. It can clearly be seen that the neural network is able to detect the fluctuations in the standard deviation. However, as Fig. 8(d) indicates the fidelity of the reproduction of the process trace is not good. In this figure training was stopped at 5000 samples so all the data shown are for validation of the network prediction.

IX. Cluster Classification and Noise Analysis Experiments: Summary

In addition to the above, we conducted experiments in noise analysis and application of self-organizing neural networks for fault identification. Briefly, the self-organizing networks used the full time stream data as input with a window width of one time segment. This input space was shattered by a second order polynomial, and the connections from the polynomial layer were trained by the winner-take-all approach

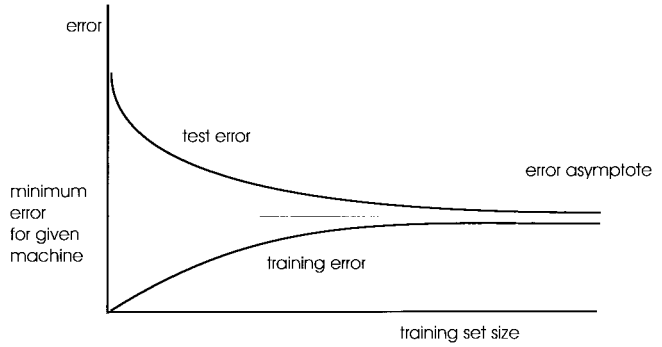


Fig. 9. Schematic showing the asymptotic limits for the training and testing curves.

(cf. [26], [27]). The network had 15 (arbitrarily chosen) output classes. Although the approach seems reasonable, (i.e., pattern matching input vectors to output classes) we found that apparently the noise level was too high to enable adequate clustering for classification.

In order to understand better the noise issues we applied some methods discussed by Cortes *et al.* [28], [29]. Referring to Fig. 9, to a first approximation, the asymptote of error for the training and testing curve for a neural network that has been optimized in number of nodes and trained with a huge training file, is equal to the noise inherent in the database. Thus, the neural network can act as a tool for measuring the noise in the training data. This will be an indication of the prediction error-limit with the existing data. Doing this with the statistical summary data and using the neural network described in the Section VII, we compute that the error inherent in the database is about 31%. So if the fault precursors are much below this level we will not succeed in seeing them with a neural network. Of course faults are much stronger signals and they can be seen by the neural network, as demonstrated in Sections VI and VII.

What is the noise limit at which one could detect fault precursors? Of course as seen in Fig. 3, the noise is not Gaussian, but for the sake of argument, assume Gaussian noise. Following some of the ideas of [30] and [31], we constructed a surrogate data set, or an artificial data set, to test noise limits in detection of precursors. Specifically, we wanted to quantify the noise level for detection of precursors with time delay neural networks. Using the artificially constructed data set, and successively adding Gaussian and random "spikey" noise, we found that precursors, (i.e., deviants on the order of 10%) could be detected with noise as large as 30% (all relative to the main signal). Further, we found that the noise could be as large as 40% to still detect failures, but the precursors were no longer observed.

X. Discussion and Conclusion

The majority of the literature on failure analysis involves identification and classification (cf. Basseville and Nikiforov [10]). The paradigm for most of the methods is based on having a model of the process. Then we can observe precursors as level changes in the target. We have shown results for several approaches to failure mode prediction of subsystems. Although our experiments have focused on plasma reactors, it should be obvious that similar techniques could be applied to many processing tools and processes.

Predicting pressure or some other reactor state is a first step in predicting failures. Without the ability to predict the state we can not predict a fault. We have examined an autoregressive model that shows a lag of 30 time units. This model is an example of a Markov process. If we know the pressure at time, t we can predict the pressure at $t + n$. However, our prediction accuracy will decrease if there is a failure between t and $t + n$. We have demonstrated that a neural network can induce, or build, a Markov model from a database of observations of the process state. The induced Markov model is only as accurate as the noise inherent in the database. In our case, we discovered that the noise level is at 31%.

Detecting precursors to failures is equivalent to predicting failures. We demonstrate that neural networks can detect precursors to failures, even in the presence of noise at 30%. The best approach we discovered is to use time-stream data of standard deviations of the process signals. In other words, by collecting the standard deviation of fluctuations of the process signals for each process step and assembling a time-stream of these data we can observe precursors about a dozen wafers prior to actual failure events. Faults, of course, are stronger signals than fault-precursors and they can be detected at a level of well above 30%. The neural network can filter a significant level of noise on the input, deal with cross correlations and autocorrelations. It can self-organize its own model of the process from a small training set and continue to adapt to changing process conditions.

Acknowledgment

The authors thank V. Bakshi of Sematech, V. Vapnik of AT&T Labs, A. Woodard, B. Kotzias, and S. Neston of Lucent Technologies, and R. Frye, of Bell Labs, for technical discussions. They also thank D. Hancock, Lucent Technologies, for database assistance, J. Plummer of National Semiconductor, and G. May of the Georgia Institute of Technology for a critical review and comments on the manuscript prior to submitting. Their insightful comments likely improved the readability of the manuscript.

References

[1] P. Smyth, "Hidden Markov models and neural networks for fault detection in dynamic systems," in *Neural Networks for Signal Processing, III, Proc. 1993 IEEE-SP Workshop*, C. A. Kamm, G. M. Kuhn, B. Yoon, R. Chiellappa, and S. Y. Kung, Eds. New York: IEEE Press, 1993, pp. 582–591.

[2] E. Hatzipantelis, A. Murray, and J. Penman, "Comparing hidden Markov models with artificial neural network architectures for condition monitoring applications," in *Proc. Artificial Neural Networks*, June 1995, pp. 369–374.

[3] O. A. Basir and H. C. Shen, "Modeling and fusing uncertain multisensory data," *J. Robot. Syst.*, vol. 13, no. 2, pp. 95–109, 1996.

[4] A. Alessandri and T. Parisini, "Model-based fault diagnosis using nonlinear estimators: A neural network approach," in *Amer. Control Conf.*, June 1997.

[5] W. Hafez, T. Ross, and D. Gadd, "Machine learning for model-based diagnosis," in *Amer. Control Conf.*, July 1997.

[6] I. K. Konstantopoulos and P. J. Antsaklis, "Controllers with diagnostic capabilities. A neural network implementation," *J. Intell. Robot. Syst.*, vol. 12, pp. 197–228, 1995.

[7] B. Dubuisson, M. H. Masson, and C. Frelicot, "Some topics in using pattern recognition for system diagnosis," *Elektronnoe Modelirovanie*, vol. 17, no. 5, pp. 76–88, 1995.
[8] R. Isermann, "Process fault detection based on modeling and estimation methods—A survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.
[9] W. W. Wei, *Time Series Analysis Univariant and Multivariant Methods.* Redwood City, CA: Addison-Wesley, 1990.
[10] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes, Theory and Application.* Englewood Cliffs, NJ: Prentice-Hall, 1993.
[11] A. J. Duncan, *Quality Control and Industrial Statistics*, 4th ed. Homewood, IL: Irwin, 1974.
[12] D. C. Montgomery, *Introduction to Statistical Quality Control*, 2nd ed. New York: Wiley, 1991.
[13] G. S. May and C. J. Spanos, "Automated malfunction diagnosis of semiconductor fabrication equipment: A plasma etch application," *IEEE Trans. Semiconduct. Manufact.*, vol. 6, pp. 28–40, Feb. 1993.
[14] M. D. Baker, C. D., Himmel, and G. S. May, "Time series modeling of reactive ion etching using neural networks," *IEEE Trans. Semiconduct. Manufact.*, vol. 8, pp. 62–71, Feb. 1995.
[15] B. Kim and G. S. May, "Real-time diagnosis of semiconductor manufacturing equipment using a hybrid neural network expert system," *IEEE Trans. Comp., Packag., Manufact. technol. C*, vol. 20, pp. 39–47, Jan. 1997.
[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1988.
[17] A. S. Weigend and N. A. Gershenfeld, *Time Series Prediction: Forecasting the Future and Understanding the Past.* Reading, MA: Addison-Wesley, 1994.
[18] A. N. Shiryaev, *Probability*, 2nd ed., R. P. Boas, Ed. New York: Springer, 1996, translated.
[19] D. T. Gillespie, *Markov Processes an Introduction for Physical Scientists.* New York: Academic, 1992.
[20] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains.* Princeton, NJ: Princeton Univ. Press, 1994.
[21] D. Hilbert, "Mathematical problems 13. Impossibility of the solution of the general equation of the 7th degree by means of functions of only two arguments," *Bull. Amer. Math. Soc.*, vol. 8, pp. 461–462, 1902.
[22] A. N. Kolmogorov, "On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition," *Dokl. Akad. Nauk.*, vol. 114, pp. 679–681, 1957.
[23] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.
[24] M. H. Hassoun, *Fundamentals of Artificial Neural Networks.* Cambridge, MA: MIT Press, 1995.
[25] V. N. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.
[26] D. E. Rumelhart and D. Zipser, "Feature discovery by competitive learning," in *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 151–193.
[27] T. Kohonen, *Self-Organization and Associative Memory*, 2nd ed. New York: Springer-Verlag, 1988.
[28] C. Cortes, L. D. Jackel, S. A. Solla, V. N. Vapnik, and J. S. Denker, "Learning curves: Asymptotic values and rate of convergence," in *NIPS 6, Neural Information Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds. San Francisco, CA: Morgan Kaufmann, 1994, vol. 6, pp. 327–334.
[29] C. Cortes, L. D. Jackel, and W. P. Chiang, "Limits on learning machine accuracy imposed by data quality," in *NIPS 7, Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen, Eds. Cambridge, MA: MIT Press, 1995, pp. 239–246.
[30] M. B. Kennel and S. Isabelle, "Method to distinguish possible chaos from colored noise and to determine embedding parameters," *Phys. Rev. A*, vol. 46, no. 6, pp. 3111–3118, 1992.
[31] J. Theiler, S. Eubank, A. Longtin, B. Galdrikian, and J. D. Farmer, "Testing for nonlinearity in time series: The method of surrogate data," *Physica D*, vol. 58, pp. 77–94, 1992.
[32] Y. Bar-Yam, *Dynamics of Complex Systems.* Reading, MA: Addison-Wesley, 1997.

Edward A. Rietman (M'90) received the B.S. degrees in both physics and chemistry, the M.S. degree in materials science, and the Ph.D. degree in cybernetics.

He has been with Bell Labs, Murray Hill, NJ, since 1982, and worked on solid-state ionics, charge density wave materials, and superconductors. He has been doing research on neural network applications for 11 years. His other interests are computer integrated manufacturing, complex systems, and evolutionary dynamics.

Milton Beachy received the M.S. and B.S. degrees in computer science from the Illinois Institute of Technology, Chicago.

He is a Senior Process Engineer in the Plasma Etch area at Lucent Microelectronics, Orlando, FL. He is responsible for the quality of wafer dry etch processing. Prior to his current assignment, he developed software for IC manufacturing computer integrated manufacturing systems.