ARTIFICIAL INTELLIGENCE IN SEMICONDUCTOR MANUFACTURING

INTRODUCTION

Fabrication of semiconductor devices and integrated circuits continues to be a profitable and extremely expensive operation. The increasing usefulness of integrated circuits in multimedia and video applications brings the total market value of the semiconductor industry well above $200 billion. In fact, semiconductor manufacturing that includes nanotechnologies has become so capital-intensive that only a few very large companies participate. A typical state-of-the-art, high volume manufacturing facility built to serve one or two generations of technology today costs about $3 billion, which represents a 10 fold increase over the cost of a comparable facility 20 years ago (Fig. 1). Furthermore, the introduction of newer generations of more advanced chip-making technology into a product line is as frequent as every 18 months, which brings the estimated net-present-cost (*NPC*) of building and operating a wafer fabrication facility over the next 10 years amounts to $7 billion. If this trend continues at its present rate, by the turn of the next decade the semiconductor fabrication facility start up and operation costs may exceed the yearly revenue for top semiconductor companies.

As a result of rising start up and operation costs, the challenge before semiconductor manufacturers is to offset an extremely large capital investment with a greater amount of technological innovation, efficiency, and flexibility in the fabrication process. In other words, the objective is to use the latest developments in computer hardware and software technology to enhance the manufacturing methods, which are becoming increasingly expensive and complex. In effect, this effort in computer-integrated manufacturing of integrated circuits (IC-CIM) is aimed at optimizing the cost effectiveness of integrated circuit manufacturing as computer-aided design (*CAD*) has dramatically affected the economics of circuit design (2).

Under the overall heading of reducing manufacturing costs, several important subtasks have been identified, which include increasing chip fabrication yield, reducing product cycle time, maintaining consistent levels of produce quality and performance, improving the reliability of processing equipment, and forming solid interactions between design and manufacturing. Unlike the manufacturing of discrete parts, such as electrical appliances, where relatively little rework is required and a yield greater than 95% on sellable products is often realized, the manufacture of integrated circuits faces unique obstacles. For example, semiconductor fabrication processes typically include over 300 sequential steps after raw silicon wafers are released into a manufacturing line that contains over 100 dedicated manufacturing tools. At each step in the manufacturing process, yield loss occurs. As a result, IC manufacturing processes have yields as low as 20–80%. The problem of low yield is particularly severe for new methodologies and fabrication sequences and is expected to worsen as device features shrink and process integration become more complex.

Manufacturing efficiency remains a top priority in the semiconductor industry. Maintaining product quality in an IC manufacturing facility requires strict control of literally hundreds or even thousands of process variables. As devices become more complex, process integration issues also add to the challenge of reducing semiconductor manufacturing costs and continually improving the production process. The implementation of effective IC-CIM systems offers the promise of overcoming such obstacles. The interdependent issues of high yield, high quality, and low cycle time have been addressed in part by the ongoing development of several critical capabilities in state-of-the-art IC-CIM systems: *in situ* process monitoring, process/equipment modeling, real-time closed-loop process control, and equipment malfunction diagnosis. Each of these activities increases throughput and improves yield by preventing potential misprocessing, but each also presents significant engineering challenges in effective implementation and deployment.

ARTIFICIAL INTELLIGENCE TOOLS

As semiconductor manufacturing grows increasingly complex, so does the challenge of modeling semiconductor fabrication processes. Advanced modeling and process control tools are required to resolve the subtle relationships between processing steps and output parameters and to provide adequate malfunction diagnosis in advanced manufacturing systems. Artificial intelligence tools of interest include, but are not limited to, methodologies for advanced learning, modeling, control, and prediction. Proper implementation of these tools will serve to continually improve product yield and ultimately influence semiconductor manufacturing costs.

Neural Networks

The use of artificial neural networks in various manufacturing applications has steadily increased (3)and the semiconductor manufacturing area has benefited as well. Neural networks have emerged as a powerful technology for assisting IC-CIM systems in performing process monitoring, modeling, control, and diagnostic functions. Because of their inherent learning capability, adaptability, robustness, and ability to generalize, neural nets are used to solve problems that have heretofore resisted solutions by other more traditional methods.

A neural network can be described generally as a machine that models the way in which the brain performs a task or function (4). Such networks have found increasing usage in computational tasks including modeling, signal processing, and pattern recognition. Although the term neural network stems from the fact that these systems crudely mimic the behavior of biological neurons, the neural networks used in semiconductor manufacturing applications actually have little to do with biology. However, they share some of the advantages that biological organisms have over standard computation systems. Neural networks are capable of performing highly complex mappings

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 2007 John Wiley & Sons, Inc.

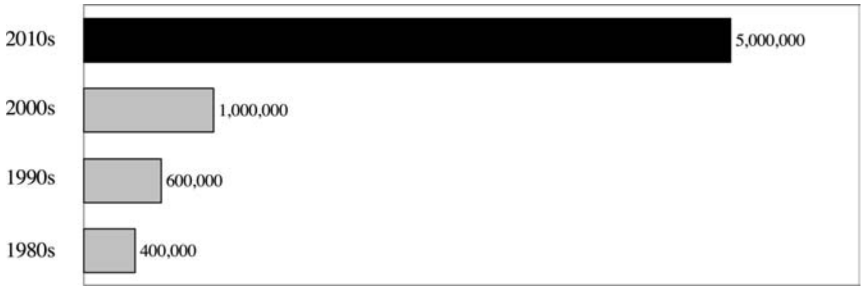


Figure 1. Graph of rising integrated circuit fabrication costs in thousands of dollars over the last several decades (1).

on noisy and/or nonlinear data, thereby inferring unassuming relationships between diverse sets of input and output parameters. Moreover, these networks can also generalize well enough to learn from input data and provide a reasonable output from data not encountered during the learning process.

Several neural network architectures and training algorithms exist for manufacturing applications. Hopfield networks, for example, have been used for solving combinatorial optimization problems, such as optimal scheduling (5). However, the back-propagation (*BP*) algorithm is the most generally applicable and most popular approach for semiconductor manufacturing (6, 11). Feed-forward neural networks (Fig. 2) trained by BP consist of several layers (input, hidden, and output) of simple processing elements called neurons (Fig. 3).

These rudimentary processors are interconnected so that information relevant to input/output mappings is stored in the weight of the connections between them. The basic elements of the neuron are the connection, the adder, and the activation function. The connecting links between the neurons are known as synapses. The synapses are characterized by the weights assigned to them. The adder determines the weight or strength of a neuron by summing the weights of its input signals, or synapses. As the network learns the relationships between input and output data, the weights of synapses are adjusted so that the network output approaches the desired output. The activation function serves to limit or "squash" the amplitude of the output of the neuron to some finite value. Together, the layers of neurons in BP networks receive, process, and transmit critical information about the relationships between the input parameters and corresponding responses. Unlike the input and output layers, the "hidden" layers of neurons do not interact with the outside world, but assist in performing nonlinear feature extraction on information provided by the input and output layers.

In the BP learning algorithm, the network uses both forward and backward computational passes. Initially, the network weights are randomized. Then, an input vector is presented and fed forward through the network, and the output is calculated by using this initial weight matrix. Next, the calculated output is compared with the measured output data, and the squared difference between these two vectors determines the system error. The accumulated error for all of the input–output pairs is defined as the Euclidean distance in the weight space that the network attempts to minimize. Minimization is accomplished via the *gradient descent* approach, in which the network weights are adjusted in the direction of decreasing error. It has been demonstrated that, if a sufficient number of hidden neurons are present, a three-layer BP network can encode any arbitrary input–output relationship (12).

To begin the learning process, weights of the neurons are randomized and a set of training examples is passed through the neural network. The outputs of neurons in the lth layer become inputs to the neurons in the next layer k. The internal activity level $s_j^{(1)}(n)$ for neuron j in layer l is

$$s_j^{(l)}(n) = i = 0 \sum w_{ji}^{(l)}(n) o_i^{(l-1)}(n) \tag{1}$$

where $o_i^{(l-1)}(n)$ is the function signal of neuron i in the previous layer (l-1) at iteration n, $w_{ji}^{(l)}(n)$ is the synaptic weight of neuron j in layer l that is fed from neuron i in layer l-1, and p is the number of neurons in the lth layer. For i=0, $o_0^{(l-1)}(n) = -1$ and $w_{j0}^{(l)}(n) = \theta_j^{(l)}(n)$, where $\theta_j^{(l)}(n)$ is the threshold applied to neuron j in layer l. Then, the output signal of neuron j in layer l is

$$o_j^{(l)}(n) = \{ \begin{array}{ll} \dfrac{1}{1 + exp[-s_j^{(l)}(n)]}, & 1l < L \\ y_j(n), & l = L \end{array} \tag{2}$$

where $x_j(n)$ is the jth element of the input vector in the first hidden layer (i.e., l=1) and L denotes the last layer. The output of the network $y_\mathrm{k}(n)$ is then compared with the desired response $d_\mathrm{k}(n)$, and the error signal is generated. The error signal is mathematically expressed as

$$e_k(n) = 1/2[d_k(n) - y(n)]^2 \tag{3}$$

where $e_\mathrm{k}(n)$ is the error of neuron k at time step n. After a forward pass through the network, this error signal is used to apply a corrective adjustment to the neuron. Learning occurs by minimizing error through modification of the weights, one layer at a time. The error signal is minimized using the *generalized delta rule* based on the gradient descent approach. The expressions for the weight changes (i.e., "deltas") of the output layer and other layers are

$$\delta_j^{(L)}(n) = [^\mathrm{a} y_j - y_j^{(L)}(n)] y_j(n)[1 - y_j(n)] \tag{4}$$

$$\delta_j^{(l)}(n) = \delta_j^{(l)}(n)[1 - \delta_j^{(l)}(n)]k \sum \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \tag{5}$$

Once the outputs of the last layer are calculated, weights are updated by the deltas for each node calculated from the output layer and back-propagated to the input layer. The generalized delta rule is

$$\Delta w_{ji}^{(l)}(n) = [w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)] \tag{6}$$

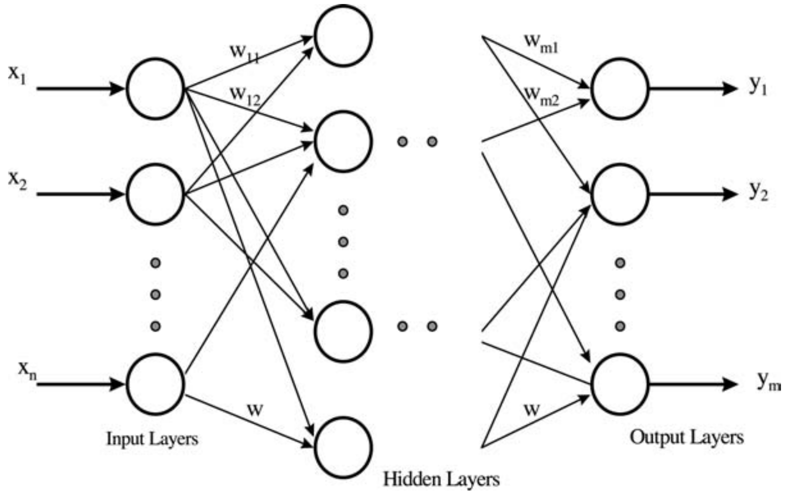


Figure 2. Diagram of neural network.

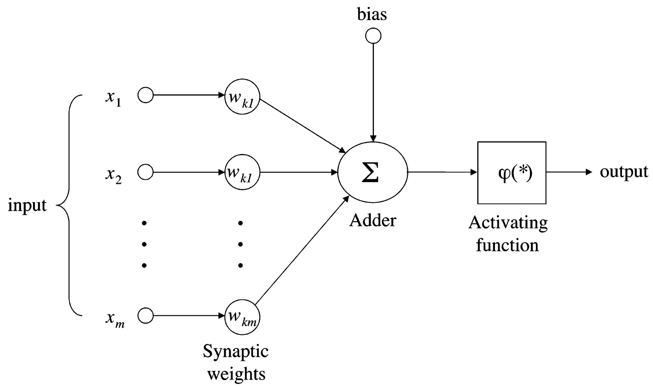


Figure 3. Model of a neuron.

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta\delta_j^{(l)}(n)o_i^{(l-1)}(n) + \alpha\Delta w_{ji}^{(l)}(n) \qquad (7)$$

where n is the number of iterations, η is the *learning rate*, and α is the *momentum*. The learning rate is a constant that represents the rate at which a weight will be changed along its slope to the minimum error. The momentum coefficient is a constant that includes a portion of the previous weight change. The momentum coefficient which generally ranges between 0 and 1, may have the benefit of preventing the learning process from terminating in a shallow local minimum on the error surface. When the network is fully trained, appropriate weights, w_{kj}, are derived such that the network output represents the relationship between the inputs and outputs of the data set.

Genetic Algorithms

Genetic algorithms (GAs) are particularly promising for the optimization of semiconductor manufacturing processes (13). Theoretical analyses suggest that GAs quickly locate high performance regions in extremely large and complex search spaces and possess some natural insensitivity to noise (14). In essence, GAs are a powerful optimization tool that reduces the likelihood of getting stuck at a local optimum and instead locates the global optimum necessary for improving manufacturing yield. As GAs determine multiple searching points for the next evaluation, a slight disadvantage becomes evident: The convergence time speed near the global optimum becomes slow. Fortunately, the distinct searching and optimization performance of GAs usually outweigh the lack of convergence speed.

Genetic algorithms are essentially guided stochastic search techniques based on the principles of genetics (14, 15). They use three fundamental operations found in natural genetics to guide their trek through the search space: *selection*, *crossover*, and *mutation*. Using these operations, GAs search through large, irregularly shaped spaces quickly, requiring only objective function values (detailing the quality of possible solutions) to guide the search, which is an inviting characteristic, considering that the majority of commonly used search techniques require derivative information, continuity of the search space, or complete knowledge of the objective function to guide their search. Again, GAs take a more global view of the search space than many methods currently encountered in engineering optimization.

In computing terms, a genetic algorithm maps a problem onto a set of binary strings. Each string represents a potential solution. Then the GA manipulates the most promising strings in searching for improved solutions. A GA operated typically through a simple cycle of four states:

1) creation of a population of strings; 2) evaluation of each string; 3) selection of best strings; and 4) genetic manipulation to create the new population of strings. During each computational cycle, a new generation of possible solutions for a given problem is produced. At the first stage, an initial population of potential solutions is created as a starting point for the search process. Each element of the population is encoded into a string (the "chromosome") to be manipulated by the genetic operators. In the next stage, the performance (or "fitness") of each individual of the population is evaluated. Based on each individual string's fitness, a selection mechanism chooses "mates" for the genetic manipulation process. The selection policy is responsible for assuring survival of the most-fit individuals.

Binary strings are typically used in coding genetic searches. A common method of coding multi-parameter optimization problems is concatenated, multi-parameter, mapped fixed-point coding. Using this procedure, if an unsigned integer x is the decoded parameter of interest, then x is mapped linearly from $[0, 2^l]$ to a specified interface $[\mathrm{U}_{\min}, \mathrm{U}_{\max}]$ (where l is the length of the binary string). In this way, both the range and precision of the decision variables are controlled. The precision (π) of this coding is calculated as

$$\pi = \frac{U_{\max} - U_{\min}}{2^l - 1}$$

To construct a multi-parameter coding, as many single-parameter strings as required are simply concatenated. Each coding has its own sublength (i.e., its own $\mathrm{U}_{\max}$ and $\mathrm{U}_{\min}$). Figure 4 shows an example of a two-parameter coding with four bits in each parameter.

The string manipulation process employs genetic operators to produce a new population of individuals ("offspring") by manipulating the genetic "code" possessed by members ("parents") of the current population. It consists of selection, crossover, and mutation operations. Selection is the process by which strings with high fitness values (i.e., good solutions to the optimization problem under consideration) receive large numbers of copies in the new population.

In one popular method of selection, strings with fitness value F_i are assigned a proportionate probability of survival into the next generation. This probability distribution is determined according to

$$P_i = \frac{F_i}{\sum F} \tag{8}$$

Thus, an individual string fitness n times better than another's will produce n times the number of offspring in the subsequent generation. Once the strings have reproduced, they await the actions of the crossover and mutation operators.

The crossover operator takes two chromosomes and interchanges part of their genetic information to produce two new chromosomes (Fig. 5). After the crossover point is randomly chosen, portions of the parent strings (P1 and P2) are swapped to produce the new offspring (O1 and O2) based on a specified crossover probability.

Mutation is motivated by the possibility that the initially defined population might not contain all of the information necessary to solve the problem. The mutation operation is implemented by randomly changing a fixed number of bits in every generation according to a specified mutation probability (Fig. 6). Typical values for the probabilities of crossover and bit mutation range from 0.6 to 0.9 and 0.001 to 0.03, respectively. Higher rates disrupt good string building blocks more often, and for smaller populations, sampling errors tend to wash out the predictions.

Fuzzy Logic

Fuzzy set theory, first initiated by Zadeh (18), is another promising tool for control of semiconductor manufacturing processes (19–21). This theory allows the treatment of vague, imprecise, and ill-defined information in an exact mathematical way. In essence, fuzzy sets facilitate reasoning in decision making without complete and precise information. In a manufacturing environment, this technology can be used to solve problems that are complex given various assumptions and approximations. For example, the planning and scheduling of wafer production can be a major undertaking. Understanding when to schedule work that will satisfy production requests is very complex. It involves management of man hours, production tools, and cycle time, while at the same time considering manufacturing goals and processing capacity. Fuzzy logic has been implemented in this application and proven useful for modeling the uncertainty that is characteristic of semiconductor manufacturing challenges (22).

Let a classic set X (also called a crisp set) be defined as a group of x elements or objects where $x \in X$. In this explanation, X is also called a reference superset (universe of discourse). Now, let A be a crisp subset of X. The set A can be described as a set of pairs $(x, \mu_A(x))$ in which x is the element of interest and $\mu_A(x)$ is the membership function of x in the subset A, where

$$\mu_A(x) = \{ \begin{matrix} 1 \text{ if } x \in A \\ 0 \text{ if } x \notin A \end{matrix} \tag{9}$$

Once the sets are defined, fundamental operations based on the use of the membership function, $\mu_A(x)$, can be performed. These oprations include section (eq. 10), union (eq. 11), and complement (eq. 12), which are defined as

$$C = A \cap B = \{(x, \mu_C(x)) | x \in X, \mu_C(x) = min\{\mu_A(x), \mu_B(x)\}\} \tag{10}$$

$$C = A \cup B = \{(x, \mu_C(x)) | x \in X, \mu_C(x) = max\{\mu_A(x), \mu_B(x)\}\} \tag{11}$$

Complement A^C of A as

$$A^C = \{(x, \mu_{A^C}(x)) | x \in X, \mu_{A^C}(x) = 1 - \mu_A(x)\} \tag{12}$$

Through the use of fundamental operations, fuzzy set reasoning makes it possible to evaluate vague problems that are less predictable (23–26).

Dempster–Shafer Theory

The ability to ascertain if and when severe process variations occur greatly influences product yield. Identifying the onset of drift (or malfunctions) in fabrication processes can be quite tedious without the use of advanced classification schemes. One useful classification scheme for real-time malfunction diagnosis involves the Dempster-Shafer theory of evidence (developed by Arthur P. Dempster and

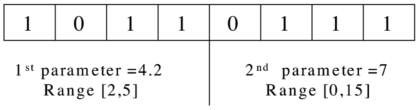


Figure 4. Example of multi-parameter binary coding (16).

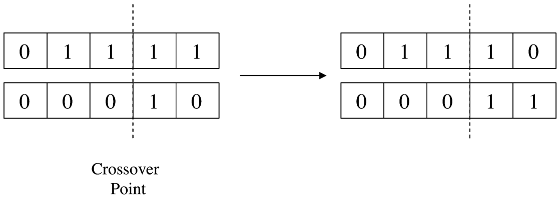


Figure 5. The crossover of two parent chromosomes resulting in two offspring (17).

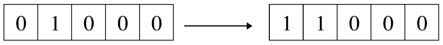


Figure 6. Chromosome experience mutation (17).

Glenn Shafer) (27, 28). This theory can be used to combine separate pieces of evidence (i.e., data offered by process metrology and analysis) to determine the likelihood of a specific event. Dempster–Shafer (D–S) theory allows one to take into account the confidence in the probabilities that accompany each possible outcome. Its implementation can result in time-varying, non-monotonic belief functions that reflect the current status of a diagnostic conclusion at any point in time.

D–S theory uniquely uses functions to represent partial belief (rather than a Bayesian probability distribution) that can develop when the finite set is interpreted as the degree of belief that the truth lies in a subset of the finite set. D–S theory also allows belief about propositions to be represented as intervals. The two bounded values in the interval are belief and plausibility, where belief < plausibility. Belief in a hypothesis is supported by the sum of the beliefs of all sets enclosed by it. It is the degree of belief that *supports* a hypothesis in part and forms the lower bound of the interval. Plausibility, on the other hand, is the 1 minus the sum of beliefs of all sets whose intersection with the hypothesis is null. It essentially represents an upper bound on the possibility that the hypothesis could actually happen. Therefore, the likelihood of a fault proposition A can be expressed as a bounded interval $[s(A), p(A)]$ that lies in [0, 1], where the parameter $s(A)$ represents the support (or belief) of proposition A and $p(A)$ is called plausibility of A. Then, the uncertainty of A can be defined as $u(A) = p(A) - s(A)$, which is the difference between the evidential plausibility and support. For instance, the evidence interval of [0.3, 0.5] for proposition A indicates that the probability of A is between 0.3 and 0.5 with an uncertainty of 0.2.

It is assumed that total belief can be divided into various portions, each assigned to a subset of the *frame of discernment*, Θ. Evidential intervals for individual faults are derived from a *basic probability mass distribution* (*BPMD*). A BPM is a function m satisfying

$$m(\phi) = 0 \tag{13}$$

$$A \subseteqq \Theta \sum m\langle A\rangle = 1 \tag{14}$$

The quantity $m\langle A\rangle$ is called the proposition A's basic probability mass, which is the measure of belief committed exactly to A, and not to any of its subsets, given a certain piece of evidence. To obtain the measure of the total belief committed to A, one must add to $m\langle A\rangle$ the quantities $m\langle A\rangle$ for all proper subsets B of A. The function assigning each subset A of Θ the sum of all basic probability numbers for subset of A is called a *belief function*, which is interpreted as a measure of the total belief committed to A, or

$$\mathrm{Bel(A)} = B \subseteqq A \sum m\langle B\rangle \tag{15}$$

The belief function with the simplest structure is obtained by setting $m\langle\Theta\rangle = 1$ and $m\langle A\rangle = 0$ for all $A \neq \Theta$. In other words,

$$\mathrm{Bel(B)} = \begin{cases} 0 \text{ if B does not contain A} \\ \text{s if B contains A but } \mathrm{B} \neq \Theta \\ 1 \text{ if } \mathrm{B} = \Theta \end{cases} \tag{16}$$

A subset A of a frame Θ is called a focal element of a belief function if $m\langle A\rangle > 0$. The union of all the focal elements of a belief function is called its core. Other types of belief functions are *Bayesian* belief functions, whose focal elements are singleton and simple support functions that have only one focal element in addition to that of Θ. Note that the belief in a proposition A and the belief in its negation $\overline{A}$ does not necessarily sum to 1, which is a major difference between Dempster–Shafer theory and traditional probability theory. According to Dempster–Shafer theory, the belief of $\overline{A}$ can be expressed by the *degree of doubt*: $\mathrm{Dou}(A) = \mathrm{Bel}(\overline{A})$. A more useful quantity is *plausibility*: $\mathrm{P}(A) = 1 - \mathrm{Bel}(\overline{A})$, which defines to what extent one fails to doubt in A or finds A plausible. It is straightforward to show that

$$\mathrm{P}(A) = B \cap A \neq \phi \sum m\langle B\rangle \tag{17}$$

The quantity Bel(A) can be interpreted as a global measure of one's belief that proposition A is true, whereas P(A) may be viewed as the amount of belief that could be placed in A if further information of belief became available.

Two BPM's m_1 and m_2 over the same frame of discernment Θ can also be combined by Dempster's rule of combination to yield a new BPM, $m = m_1 \oplus m_2$, called the orthogonal sum of m_1 and m_2, which is defined by

$$m\langle Z\rangle = \frac{\sum m_1\langle X\rangle m_2\langle Y\rangle}{1-k} \tag{17}$$

where $Z = X \cap Y$ and

$$k = \sum m_1\langle X\rangle m_2\langle Y\rangle \tag{18}$$

where $X \cap Y = \phi$.

PROCESS MODELING

Accurate process modeling is essential to semiconductor manufacturing. However, first principle models must occasionally be simplified because of environmental constraints such as hardware limitations, cost, time, or limitations in modeling methodologies. The ability of neural networks to learn input/output relationships from limited data is quite beneficial in semiconductor manufacturing, where a plethora of highly nonlinear fabrication processes exist and where experimental data for process modeling are expensive to obtain. The use of artificial neural networks to model semiconductor manufacturing process with limited fabrication information has yielded very impressive results in various applications including chemical vapor deposition (*CVD*) processes (19–34), reactive ion etch (*RIE*) processes (35–39), photolithography processes (40, 41), rapid thermal process (*RTP*) (42), chemical and mechanical polishing (*CMP*) processes (43), packaging processes (16–47), and production scheduling (7, 48). In so doing, the basic strategy is usually to perform a series of statistically designed characterization experiments and then to train neural nets to model the experimental data. The process characterization experiments typically consist of a factorial or reduced factorial exploration of the input parameter space, which may be subsequently augmented by a more advanced experimental design. Each set of input conditions in the design corresponds to a particular set of measured process responses. This input/output mapping is precisely what the neural network learns. In general, standard BP neural networks are the most popular to model semiconductor process.

Standard Modeling

As an example of the neural network-based process modeling procedure, Pratap et al. present modeling and sensitivity analysis of circuit parameters for flip-chip interconnects using standard BP neural networks (47). Flip-chip technology has emerged as an attractive interconnection scheme for high frequency RF applications because flip-chip interconnect technology provides higher packaging density and superior electrical, mechanical, and thermal performance with lower package profile and cost. To enhance microwave circuits, precise modeling and characterization of interconnections as a function of layout parameters is essential to optimize the performance of the flip-chip signal transition. To achieve this optimization, Pratap et al. developed a standard BP-based model for the electrical performance of flip-chip transition up to 35 GHz in terms of the physical and geometrical parameters. In this work, system performance was characterized by s-parameter measurements. The data used to derive the equivalent circuit model, as well as the s-parameters, was generated using a 2^{5-1} fractional factorial experiment. Data from these experiments were subsequently used to train neural networks using the back-propagation algorithm. Empirical analysis of the simple flip-chip configuration shown in Fig. 7 led to the selection of the following factors for experimental design and model development:

o conductor overlap (bumps are always placed in the center of the overlap area);
w CPW signal line width;
d distance from ground bump center to the edge of the ground plane;
a bump diameter; and
h bump height.

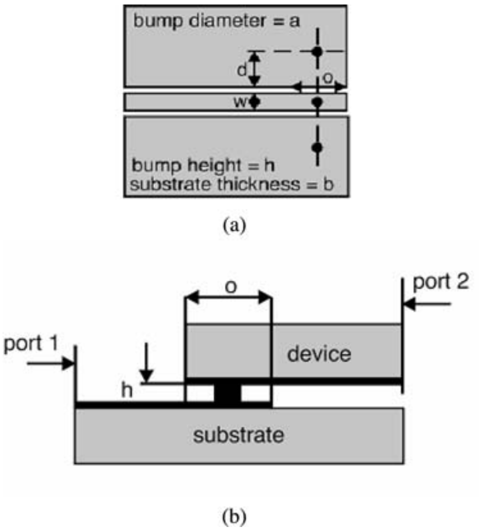
The output variables for the experiments were S_{11} (dB) (reflection coefficient) and S_{21} (dB) (insertion loss). As the structure is symmetrical, $S_{11} = S_{22}$ and $S_{12} = S_{21}$.

The s-parameters obtained were then used to obtain the inductance (L) and capacitance (C1 = C2 = C) values of the π lumped element model shown in Fig. 8. Thus, all the required components for the complete characterization of the electrical behavior of the flip-chip transitions were extracted.

To model four electrical parameters (S_{11}, S_{21}, L, and C), four separate neural networks were used for greater accuracy. Network training was accomplished using the Object-Oriented Neural Network Simulator (ObOrNNS) a Java-based software package developed by the Intelligent Semiconductor Manufacturing group at Georgia Tech (49). Overall, 75% of the data was used to train the models and the remaining 25% of the data was used for validation. The modeling results indicate prediction errors from 3-17%. This accuracy is reasonable, considering the fact that the test data set was at the boundary of the training data. The trained neural networks were used to further study the impact of various layout parameters on the electrical properties of the flip-chip transitions.

J. Müller et al. of Robert Bosch GmbH in Germany, DuPont Photomasks in France, and Infineon Technology in Germany used another type of neural network, the self-organizing map (*SOM*), for analysis of semiconductor manufacturing parameters both in the front-end and back-end part of the fabrication process (50). Based on production data from two major European semiconductor manufacturing lines, layers of metallization processes were characterized. Also, SOM was used to find correlation between equipment and key process parameters. The authors concluded that SOM has advantages in detecting small process misalignments or process drifts.

In terms of a commercialized example, NeuMath, formerly known as IBEX Process Technology, provides advanced solutions to optimize process control and maximize yield in the semiconductor manufacturing industry (51), for which they combine neural networks and advanced mathematical techniques to model the complex processes used in semiconductor manufacturing. Using neu-



(a)

(b)

Figure 7. (*a*) Schematic of bump configuration. (*b*) Side view (47).

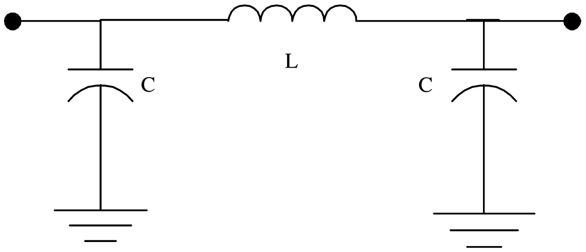


Figure 8. Lumped element model of flip-chip transition (21).

ral network-based advanced analysis technology, NeuMath has developed several solutions, including a yield optimizer and a dynamic neural controller (*DNC*).

Hybrid Methods

As shown in previous neural process modeling examples, implementation of the single standard neural network was employed to perform process modeling tasks. However, innovative modifications of standard BP have also been developed for certain other applications of semiconductor process modeling. In one case, BP has been combined with simulated annealing to enhance model accuracy. A second adjustment has been developed that incorporates knowledge of process chemistry and physics into a semi-empirical or hybrid model, with advantages over the purely empirical black-box approach previously described.

Neural Networks and Simulated Annealing. Simulated annealing (*SA*) is a popular combinational optimization technique. SA is based on the physics of annealing and is a process in which a material is heated and then cooled very slowly to a freezing point, resulting in a highly ordered crystal lattice without any impurities such that the system ultimately winds up in a state of very low energy. This SA mechanism can be applied to neural network training by means of a stochastic weight update. For example, at low temperature, the network is very sensitive to state change, but has difficulty reaching the equilibrium state. In contrast, at high temperature, the network ignores small energy differences and rapidly approaches equilibrium. A typical SA process starts with a very high temperature, where the system state is generated at random.

Kim and May used neural networks and SA to model the reactive ion etching (*RIE*) process. RIE in a radio frequency (*RF*) glow discharge is one of the most effective means of dry etching in semiconductor manufacturing (36). For this reason, many researchers have been focusing on development of accurate RIE process models. Due to the limitations of plasma etch modeling from a fundamental physical standpoint, adaptive learning techniques that use neural networks combined with statistical experimental design methods have been developed. To increase the modeling performance, Kim and May developed an alternative learning rule, the "K-step prediction" rule, and used it for BP neural network training as an alternative to the generalized delta rule.

The rationale for this new rule is as follows: Neural network training rules adjust synapse strengths to satisfy the constraints given to the network. This new update scheme is expressed as

$$w_{i\,jk}(n+1) = w_{i\,jk}(n) + \eta \Delta w_{i\,jk}(n+1) + 1 \sum \gamma_K w_{i\,jk}(n-K) \tag{19}$$

where w_{ijk} is the connection strength between the *j*th neuron in layer (k-1) and the *i*th neuron in layer *k*, Δw_{ijk} is the calculated change in that weight that reduces the error function of the network, and η is the learning rate. The last term in the above expression provides the network with a degree of "long-term memory" (52). The integer K determines the number of sets of previous weights stored and the γ_k factor allows the system to place varying degree, of emphasis on weight sets from different training epochs. This memory-based weight update scheme is combined with a variation of SA to assist the BP algorithm in minimizing the system error function. In neural network training, the system error plays a role similar to the energy state of a system under cooling of annealing process at thermodynamics. Applying the concept of SA in neural network training is analogous to using the following "thermo-squashing" function in place of the usual sigmoidal transfer function:

$$\frac{1}{1+e^{-(\frac{net_{ik}+\beta_{ik}}{\lambda T_0})}} \tag{20}$$

where $net_{i,k}$ is the weighted sum of neural inputs and β_{ik} is the neural threshold. Annealing the network at high temperature early on leads to rapid location of the general vicinity of the global minimum of the error surface. The training algorithm will then remain within the attractive basin of the global minimum as the temperature decreases, preventing any significant uphill excursion.

To model the RIE process, pressure, RF power, and gas flow of O_2 and CHF_3 were used as modeling inputs, and etch rate, anisotropy, etch uniformity, and selectivity were considered as modeling outputs. For the K-step prediction rule, K was set to two and various values of $(_1$, $(_2$, T_0, and (were systematically investigated. Increased accuracy was consistently obtained for larger values for $(_1$ and smaller values of $(_2$, indicating the relative importance of more recent training epochs. The best overall results for all four etch response models were achieved for $(_1 = 0.89$, $(_2 = 0.08$, $T_0 = 100$, and $\gamma = 0.999$. A comparison of network prediction results for SA-based K-step prediction and the conventional generalized delta rule showed more than 50% of improvement using the former.

Neural Networks and Principal Component Analysis. Hong et al. used neural networks and principal component analysis (*PCA*) to model RIE using optical emission spectroscopy (*OES*) data (38). Although OES is an excellent tool for monitoring plasma emission intensity, a primary issue with its use is the large dimensionality of the spectroscopic data. To alleviate this concern, PCA was implemented as a mechanism for feature extraction to reduce the dimensionality of OES data. PCA is a well-known statistical method that can reduce the dimension of a multivariate data set (53).

Consider a vector $\mathbf{x}$ that consists of p random variables. Let Σ be the covariance matrix of $\mathbf{x}$. Then, for $k = 1, 2, \ldots, p$, the kth principal component (*PC*) is given by

$$\mathbf{t}_k = \mathbf{u}_k^T \mathbf{x} \tag{21}$$

where u_k is an eigenvector of Σ corresponding to its kth largest eigenvalue and T represents the transpose operation. Dimensionality reduction through PCA is achieved by transforming the OES data to a new set of coordinates (i.e., selected eigenvectors), which are uncorrelated and ordered such that the first few retain most of the variation present in the original data set. Generally, if the eigenvalues are ordered from largest to smallest, then the first few PCs will account for most of the variation in the original vector x. A simplified example of PCA with two measurement variables, x_1 and x_2, is presented in Fig. 9.

OES data were generated from a 2^4 factorial experiment designed to characterize RIE process variation during the etching of benzocyclobutene (*BCB*) in a SF_6/O_2 plasma, with controllable input factors consisting of the two gas flows, RF power, and chamber pressure. The OES data, consisting of 226 wavelengths sampled every 20 seconds, were compressed into five principal components using PCA. Selected features by PCA were subsequently used to establish multilayer perceptron neural networks trained using error back-propagation to model etch rate, uniformity, selectivity, and anisotropy. Hong et al. applied autoencoder neural networks (AENNs) to capture the features of OES data and reduce its dimensionality in a similar manner to PCA (38).

An AENN is illustrated in Fig. 10. It usually has the same number of inputs and outputs. The number of hidden neurons can be adjusted to suit the problem at hand. The autoencoder bottleneck structure, with n inputs, h hidden-neurons ($h < n$), and n outputs, forces the network to form a compressed representation of the data. Training a network to reproduce its inputs seems pointless on the surface, but in reproducing the input signals at the output, the autoencoder, after training, represents the input pattern in compressed form in its hidden neurons. The hidden layer is also called the "compression layer" because it represents a compressed form of the input signals.

Hong et al. developed both PCA-based neural network models and AENN-based neural network models, and their prediction results are shown in Figs. 11 and 12. The performance of the trained neural networks was evaluated with seven vectors retained for testing purposes. Tests were repeated for three combinations of training and testing sets, and the average testing errors are shown in Table 1. For PCA-based neural network models, the models exhibited an average RMS error of 3% in training and 4.61% in testing, and the AENN-based neural network models showed an average of 3% RMS error on training and 3.47% on test data.

Semi-empirical Process Modeling. Brown and May developed a semi-empirical hybrid neural network to estimate the parameters of the kinetic model of molecular beam epitaxy (*MBE*) and analyze the microscopic processes occurring at the interfaces of the mixed anion III–V heterostructures (55). The hybrid model was constructed by characterizing the MBE growth of $GaAs_{1-y}P_y/GaAs$ heterostructures using a statistically designed experiment. These structures were formed by allowing a P_2 flux to impinge on a static As-stabilized (001) GaAs surface. The phosphorus composition (y) at the interfaces of these structures is modeled as a function of substrate temperature (T_s), phosphorus exposure time ($t_{\exp}$), and arsenic stabilizing flux (P_{As4}).

The structure of the hybrid neural network designed to predict anion intermixing for the GaAsP/GaAs het-

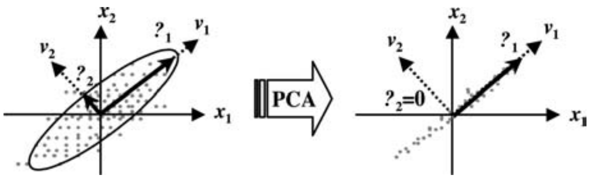


Figure 9. An illustration of principal component analysis of two measurement variables: x_1 and x_2 indicated mean centered sample data, v_1 and v_2 are eigenvectors, and σ_1 and σ_2 are corresponding standard deviations (53).

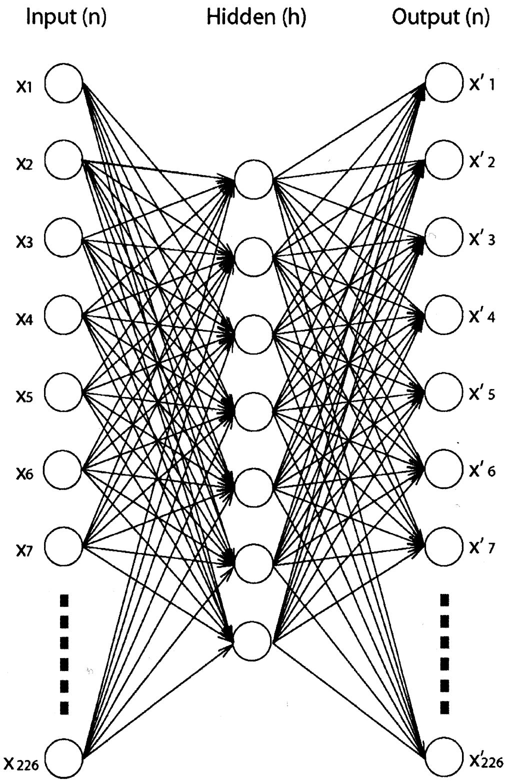


Figure 10. Bottleneck structure of an autoencoder neural network (AENN) (54).

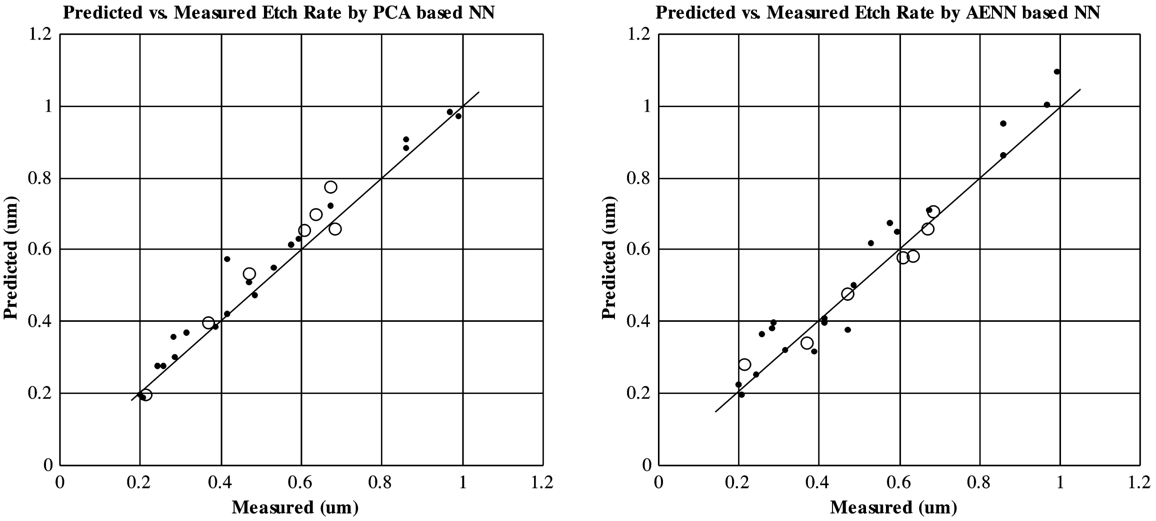


Figure 11. Neural network model predictions for etch rate generated by: (*a*) a PCA-based neural network and (*b*) an AENN-based neural network (54). (Note: Circles represent test data not used during network training.)

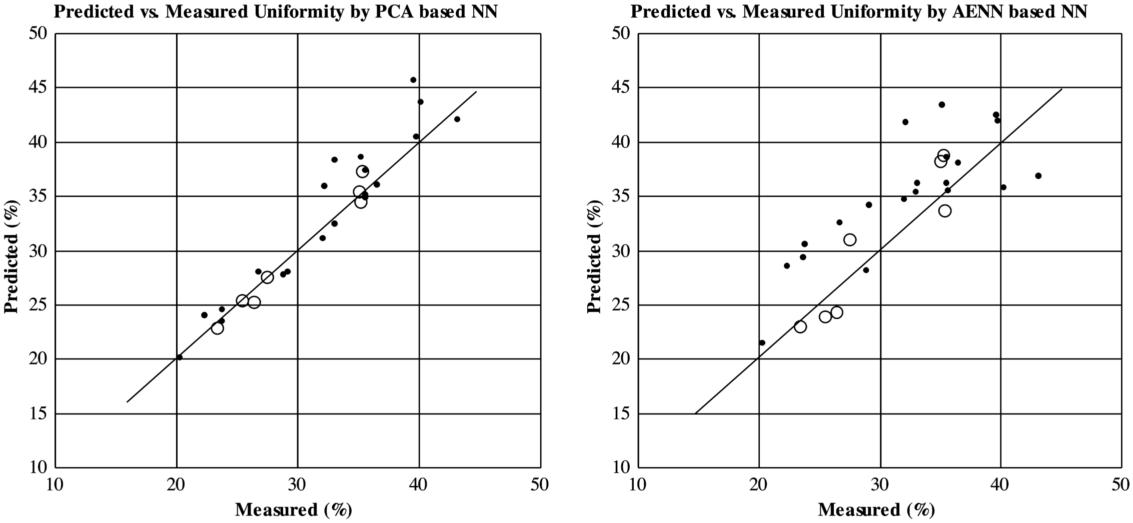


Figure 12. Neural network model predictions for uniformity generated by: (*a*) a PCA-based neural network and (*b*) an AENN-based neural network (38). (Note: Circles represent test data not used during network training.)

Table 1. Modeling Results for Etch Responses

	% Error of PCA based NN	% Error of AENN
Etch rate	1.72	1.774
Selectivity	13.6	1.536
Uniformity (%)	0.215	9.067
Anisotropy (%)	2.897	1.494

Table 2. Kinetic Parameters Obtained by the Semi-empirical Hybrid Neural Network Model

Kinetic Parameters – $As_4 = 4 \times 10^{-6}$ Torr				
s	τ_{0d}	E_d	D_0’	E_a
(cm^2)	(s)	(eV)	(molecules/s)	(eV)
0.358	6.412	2.985	17.358	0.11
Kinetic Parameters – $As_4 = 2 \times 10^{-6}$ Torr				
s	τ_{0d}	E_d	D_0’	E_a
(cm^2)	(s)	(eV)	(molecules/s)	(eV)
0.369	3.665	2.427	26.994	0.05

Table 3. Optimized Training Factors and RMSEs for Conventional BPNN Models

Etch Outputs	TT	NHN	IWD	g_b	g_l	RMSE
Profile Angle (°)	0.11	4	0.4	0.8	0.4	2.85
Al Selectivity	0.12	2	0.2	1.2	1.2	2.26
DC Bias (V)	0.08	3	1.4	0.4	0.4	53.6
Al Etch Rate (Å/min)	0.12	4	0.8	0.4	0.4	434

Table 4. Optimized Training Factors and RMSEs for GA-BPNN Models

Etch Outputs	TT	NHN	IWD	g_b	g_b	RMSE	% Improvement
Profile Angle (°)	0.0820	3	1.2737	0.7682	0.5169	2.22	22.1
Al Selectivity	0.1126	5	2.7168	0.8891	0.5356	1.65	26.9
DC Bias (V)	0.6064	3	2.5601	1.5114	0.8019	48.3	9.9
Al Etch Rate (Å/min)	0.0803	3	2.554	1.8024	1.2685	142	67.3

Table 5. Experimental Results Comparison of Recipe Synthesis Methods

Method	Film Thickness (μm)	Via Yield (%)	Via Angle (degree)	Film Retention (%)	Film Non-uniformity (%)
GA	7.09	96.7	34.6	77.6	1.48
Hybrid GA/Powell	6.93	96.7	38.7	76.3	0.49
Hybrid GA/Simplex	7.05	93.3	41.3	78.2	0.76
Target Value	7	100	75	100	0

Table 6. Deviation Between Optimal Recipe Predictions and Experimental Results

	A. Model Name			Deviation of Predictions from Experiment	
Individual Models	Ablated Thickness			1.32%	
	Top Via Diameter	30 μm		1.92%	
		40 μm		2.22%	
		50 μm		0.90%	
	Via Wall Angle	30 μm		3.35%	
		40 μm		0.33%	
		50 μm		1.36%	
	Via Resistance	40 μm		400 Ω	
		50 μm		60 Ω	
			Via Diameter	Via Wall Angle	Via Resistance
Composite Model		40 μm	6.58%	1.88%	380 Ω
		50 μm	0.76%	1.04%	150 Ω

Table 7. Supervisory Control Results (81)

Response	Without Control	With Control	% Improvement
Film Thickness (μm)	7.533	7.190	64.4%
Via Yield (%)	84.667	97.333	82.6%
Non-uniformity (%)	1.931	1.596	17.3%
Film Retention (%)	73.297	72.888	−1.5%

Table 8. VCO Input Parameters with Normal and Nonnormal Distributions (97)

Input Parameter	Range for Mean		Distribution Type	Standard Deviation
	Low	High		
Emitter Length (μm)	2	10	Normal	0.4
I_{bias} (mA)	2	10	Uniform random	0.8
C_{vardim} (μm)	10	30	Normal	1
$L_{package}$	50	100	Uniform random	5

Table 9. VCO Desing Centering with Normal and Nonnormal Distributions (97)

	Initial Value (Yield=0.01%)		Final Value (Yield=71%)	
Input Parameters	Mean	Std	Mean	Std
Emitter Length	5.22	0.4	4.14	0.4
Ibias	5.86	0.8	7.99	0.8
Cvardim	21.69	1	24.91	1
$L_{package}$	68.12	5	60.59	5
Output Parameters	Mean	Std	Mean	Std
Tuning Range	1.92	0.068	2.27	0.106
Phase Noise	−92.58	0.912	−93.63	0.W70271
Output Power	56.64	0.218	54.94	3.885

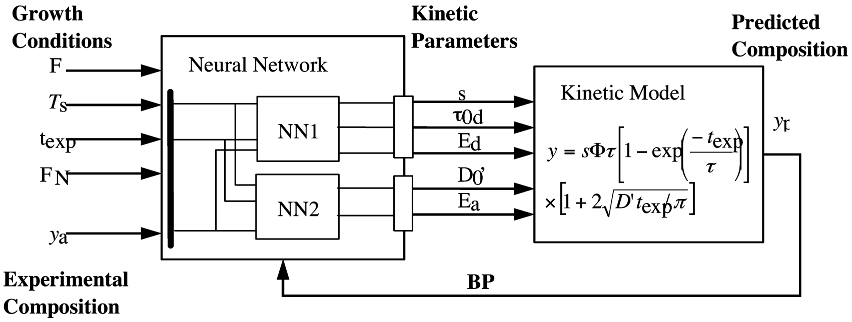


Figure 13. Hybrid neural network used to estimate the kinetic parameters for growth of the As/P heterostructures (55).

erostructures is illustrated in Fig. 13. The neural network component of the hybrid model has the MBE process conditions as its inputs. The outputs of the neural network component are the unknown parameters required to implement the kinetic model. The neural network component consists of two back-propagation neural networks in parallel.

The forward and back-propagation phases of the BP algorithm proceed in a manner similar to that of standard neural networks. Network training occurs by means of a modified error gradient that takes into account the error contribution from each kinetic parameter determined by the partial derivatives of the kinetic model (56). The forward propagation phase begins by initializing the neural network and presenting the input vectors. The outputs of the neural network component are the kinetic parameters. These unknown physical constants are used to compute the predicted phosphorus composition at the interfaces of the GaAsP/GaAs heterostructures.

Evaluation of the trained hybrid neural network model is performed in terms of the root mean squared error (*RMSE*), computed as the square root of the network prediction error (*E*). The hybrid neural network implemented for samples with an As-stabilizing flux $P_{As4} = 4 \times 10^{-6}$ torr demonstrated a training RMSE of 0.028% and a prediction RMSE of 0.574%. And the kinetic parameters derived are provided in Table 2.

Figure 14 is a comparison of diffusion coefficients predicted by the hybrid neural network with diffusion constants for Sb and P diffusion and As self-diffusion in GaAs. The hybrid neural network model accurately predicts the contribution of each of the microscopic processes occurring at the interfaces of the mixed anion III–V heterostructures.

As an another example of the application of semi-empirical neural process modeling, Kuan et al. at the Northern Taiwan Institute of Science and Technology developed hybrid neural network to the predict temperature distribution in semiconductor chips with multiple heat sources (57). In general, computational fluid dynamics (*CFD*) simulation is very popular for heat sink design because it can reduce the cost and time of the design cycle, but the thermal designers still need several trials to reach acceptable results. To solve this problem, Kuan et al. used CFD and standard BP networks. According to a comparison of the standard BP neural network and CFD results, the maximum error was about 16.43% and the RMSE was about 7.63%. After training and testing using CFD data, the BPNN model provided a quick temperature distribution as well as maximum die surface temperature under several heat sources at different locations.

OPTIMIZATION

In semiconductor manufacturing applications, neural network-based optimization has been undertaken from two fundamentally different viewpoints. The first uses statistical methods to optimize the neural process models themselves. The goal here is determining the proper network structure and set of learning parameters to minimize network training error and training time and to maximize network prediction capabilities. The second approach to optimization focuses on using neural process models to optimize a given semiconductor fabrication process or to determine specific process recipes for a desired response. Process recipe optimization may be viewed as an example of off-line process control where the objective is to estimate optimal operating points (58). Recipe optimization is designed to produce desired target output responses based on the functional relationship between controllable input parameters and process responses supplied by the process model. To satisfy (often conflicting) process objectives, search schemes are needed to find optimal process recipes.

Network Optimization

Kim and Bae developed a plasma process model using a back-propagation neural network (*BPNN*) and GAs (59). Constructing a BPNN model is complicated by the presence of several training factors, including the hidden neurons, training tolerance, initial weight distribution, and function gradients. In most applications, training factor effects are typically optimized by experimentally tuning each factor individually. However, a better predictive model might be achieved by adequately accommodating complex effects among the training factors. In this work, GAs were used to optimize training factors simultaneously as an extension of previous work (60). Depending on the number of hidden neurons (*NHN*), the BPNN prediction performance can vary significantly. The activation level (or firing strength) of a neuron in the hidden layer was determined by a bipolar sigmoid function denoted as

$$\mathrm{out_{i,k}} = \frac{1 - \mathrm{e}(-\frac{\mathrm{in_{i,k}}}{\mathrm{g_b}})}{1 + \mathrm{e}(-\frac{\mathrm{in_{i,k}}}{\mathrm{g_b}})} \tag{22}$$

where $in_{i,k}$ and $out_{i,k}$ indicate the weighted input to the *i*th neuron in the *k*th layer and output from that neuron, respectively. The parameter g_b represents the gradient of the bipolar sigmoid function. The linear function adopted in the output layer is expressed as

$$\mathrm{out_{i,k}} = \mathrm{in_{i,k}} \cdot \mathrm{g_l} \tag{23}$$

where g_l represents the gradient of the linear function. Apart from the three training factors (NHN, g_b, and g_l), the initial weight distribution and the training tolerance also influence BPNN prediction considerably. As a consequence, the total number of training factors to optimize is five. The size of the initial population of chromosomes was set to 200. Each chromosome was coded with a real value, resulting in a total chromosome length of five slots corresponding to five training factors. In each slot, random values were generated within the given experimental ranges. The performance of each chromosome was evaluated with the fitness function

$$\mathrm{F} = \frac{1}{1 + \mathrm{RMSE_{TR}}} \tag{24}$$

where $\mathrm{RMSE_{TR}}$ indicates the error calculated with nine training experiments. A selection mechanism is subsequently activated to choose the best chromosome with the highest fitness for genetic manipulation. The crossover probability was specified as 0.9, and the mutation probability was 0.01. As the termination criterion, the number of generation was set to 100.

As an illustration, this method was applied to profile angle data for a semiconductor feature. At each generation, one best model with the smallest $\mathrm{RMSE_{TR}}$ was determined, and the corresponding RMSE and fitness are shown in Fig.

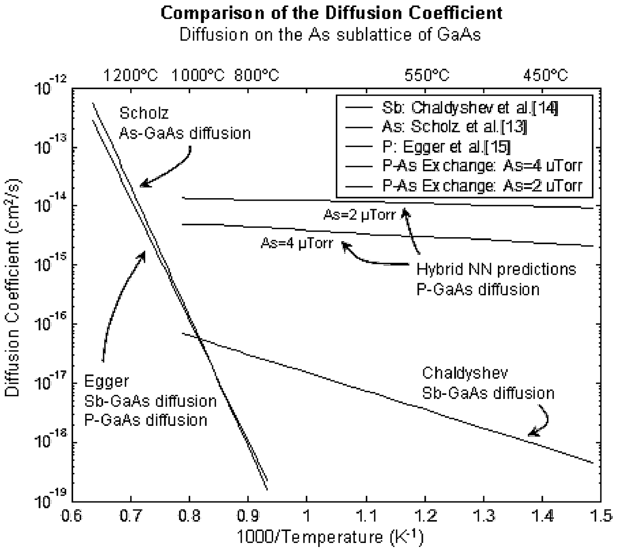


Figure 14. Comparison of diffusion coefficients predicted by the hybrid neural network with diffusion constants for Sb and P diffusion and As self-diffusion in GaAs (55).

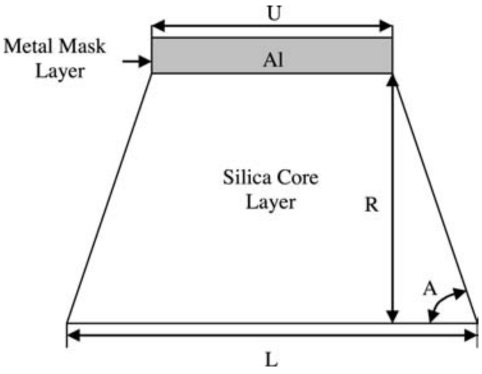


Figure 15. Schematic diagram of etched pattern for measurements (59).

15 as a function of the generation number. The best model was obtained at the 22nd generation, and the corresponding RMSE and fitness were 2.22° and 0.405, respectively.

Compared with the RMSE for the same model in Table 3, the GA-BPNN model demonstrated an improvement of about 22% in predicting the profile angle. The GA was applied to other etch outputs, and the results are shown in Table 4. The improvements calculated over the BPNN models in Table 3 are shown in the last column. As shown in Table 4, all GA-BPNN models yield better prediction performance than conventional BPNN models. More than 20% improvement was achieved for all etch outputs except the DC bias. The improvement was most significant for the Al etch rate model (more than 65%). These improvements indicate that a simultaneous optimization of the training factors is more effective in improving BPNN prediction performance than a sequential optimization of individual factor.

The percent improvement was calculated over the RMSEs for the conventional models contained in Table 3.

Process optimization

Process optimization is designed to produce desired target output responses based on the functional relationship between controllable input parameter's process responses supplied by the process model. Kim and May presented a process optimization approach for via formation in dielectric layers composed of photosensitive benzocyclobutene (*BCB*) for high density interconnect (*HDI*) in MCM-L/D substrates (61). It is known that via formation is a critical process sequence in MCM manufacturing as it greatly affects yield, density, and reliability. Therefore, to achieve low cost manufacturing, optimization of the via formation process to improve yield is crucial. For yield improvement, accurate modeling of via formation is important because it provides the basic information necessary for optimization.

In Reference (61), neural networks were used to model via formation from experimental data. Process models were developed to characterize film thickness, via yield, via geometry, film retention, and film uniformity as a function of various process parameters, including spin speed, pre-bake time, pre-bake temperature, exposure dose, development time, cure time, cure temperature, plasma de-scum power, and plasma de-scum pressure. To reduce the num-

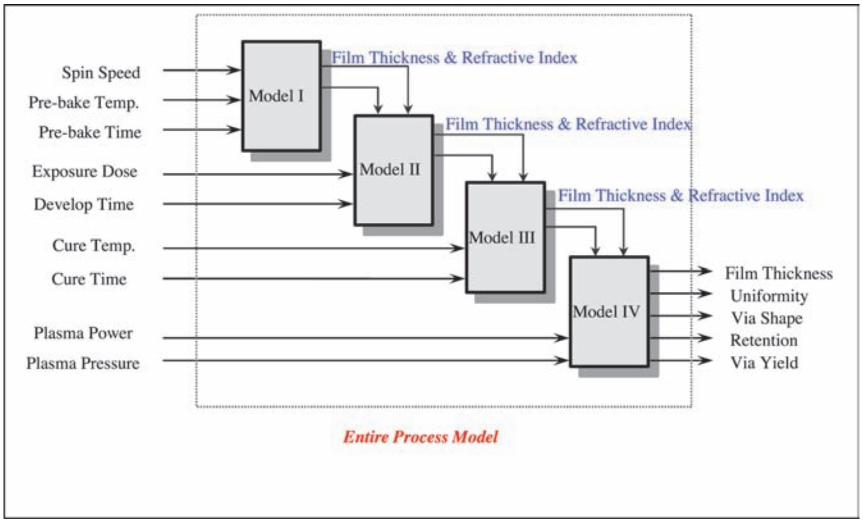


Figure 16. Block diagram of sequential neural process modeling scheme (61).

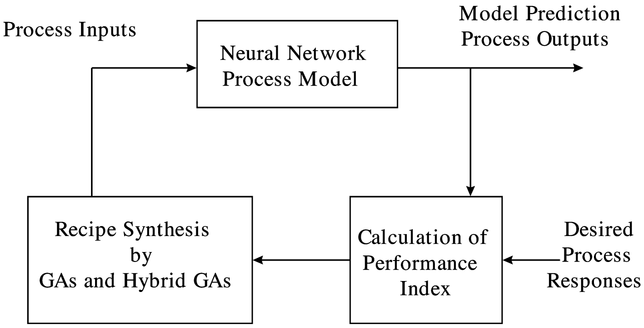


Figure 17. Diagram of recipe synthesis procedure (61).

ber of experimental trials required for process characterization, the via formation process was divided into four subprocesses (spin and pre-bake, exposure and development, cure, and de-scum). Each sub-process was modeled individually using neural networks. In each model, the input layer of neurons corresponds to the process conditions for each sub-process, and the output layer corresponds to the response variable being modeled. These particular models are unique in that the outputs of each sub-process are used as inputs to the next sub-process. For example, to model film thickness after the exposure and development sub-process, the sub-process outputs of the preceding spin and pre-bake sub-process (i.e, the film thickness and refractive index after pre-bake) were used together with exposure dose and development time as model inputs. This approach is illustrated in Fig. 16. Afterwards, the neural process models were used for optimal recipe generation using hybrid genetic algorithms as shown Fig. 17 (62).

GAs are very useful in finding optimal recipes for semiconductor manufacturing processes (60), and global search by GAs is very effective for recipe optimization problems and much less dependent on the initial search point. However, GAs require long computational time. Therefore, hybrid combinations of genetic algorithms with the other two algorithms (Powell's and simplex) can offer improved results in terms of both speed and accuracy (63). Hybrid algorithms simply consist of a global search by GAs, followed by a local search by one of the other methods. In other words, after some number of generations, the best point found using the GA is handed over to the other algorithm as a starting point. With this initial point, both Powell's algorithm and the simplex method can quickly locate the optimum.

Optimal process recipes were found based on neural process models. Five responses (film thickness, via yield, via angle, film retention, and film uniformity) were used as outputs, and the nine processing conditions are used for process setting parameters.

To quantify the search performance, a performance index was defined, and it is expressed by

$$F = \frac{1}{1 + r\sum|K_r(y_d - y)|} \tag{24}$$

where r is the number of process responses, K_r are the weights of process responses, y_d are the desired process responses, and y are the process outputs dictated by the current choice of input parameters. The process outputs are predicted by the neural process models. For genetic search, F was calculated, and strings in a given population were chosen that maximized F in each generation. The GA was stopped after 200 generations when used alone and after 100 generations when used in the hybrid methods. For the other methods, optimization was stopped when F was within a predefined tolerance. The performance of each approach was compared by simulation and experiment, and the hybrid GA/simplex algorithm showed superior results, as shown in Table 5.

Setia and May developed the present modeling and optimization via formation process for another type of material and process: laser ablation for polyimide dielectrics (17). Laser ablation is an effective process for forming vias in dielectric layers during the fabrication of system-on-package (*SOP*) multilayer substrates. Laser ablation is a material removal process that uses localized thermal energy caused by stimulated radiation. The laser ablation technique has several advantages over other via formation techniques, including the lowest number of process steps, the most desirable via shape for subsequent metallization steps (i.e., trapezoidal), and the capability of tight control over the via wall angle and production of vias with a high aspect ratio. However, some uncertainty exists regarding the quality of laser processing in via fabrication. This uncertainty is associated with the complex interactions between the dielectric polymer characteristics and those of the laser. To solve this problem, Setia and May used neural networks technique to model the ablation process and optimize the process using GAs to achieve specific target responses. For laser ablation, Anvik HexScan 2150 SXE excimer laser operating at 308 nm was used.

A 2^{5-1} fractional factorial experimental design was conducted to determine the significance of laser fluence, shot frequency, number of pulses, and the vertical and horizontal positions of a debris removal system. The first three factors are quantitative, whereas the other two are qualitative. The responses were the top via diameter, via wall angle, via resistance, and the ablated thickness of the dielectric. The via resistance measurement was conducted on the metal deposited in the ablated vias for test as shown in Fig. 18, and the measured data was used to study the effect of the debris generated (in the form of carbon residue) during the via fabrication.

Neural networks were then trained using the BP algorithm to model the ablation process using the measurement data collected from the experiment. The prediction error for nearly all responses, with the exception of ablated thickness and via resistance, was less than 5%. The prediction error for the average value of the ablated thickness was 5.5%, and that of via resistance was less than 15%.

The interrelationships between the process set points and responses can graphically illustrated using neural network models, and Fig. 19 shows the effect of laser fluence and frequency on wall angle for 50-μm vias. For these vias, steeper wall angles can be fabricated with fluence in the range of 180–188 mJ/cm^2/pulse and frequency in the range of 128–140 Hz. As the via size is larger, the wall angle does not vary as much.

Genetic algorithms were used to find optimal set points that give the desired output from the neural network models. The quantitative input factors (i.e, laser fluence, shot frequency, and number of pulses) were coded to a 10-bit string, whereas the qualitative factors were encoded in a single bit. Thus, 32-bit chromosomes were required to find the desired value(s) for the individual response models (ablated thickness and via resistance), as well as the combined response model (top via diameter, via wall angle, and via resistance) because all five inputs were significant in affecting at least one response. In this study, the desired ablated thickness, top via diameter, and via resistance were set to 25 μm, 30/40/50 μm, and 0 Ω, respectively. After recipes for the desired process set points were synthesized, experimental verification of these optimized recipes was conducted. The neuro-genetic approach adequately provided suitable process recipes. Table 6 summarizes deviations between the experimental results and the neuro-genetic model predictions. The improvement achieved from the non-optimized recipes (i.e., those recipes used during the designed experiment) and the optimized recipes was as large as 40% for the ablated thickness response, 30% for top via diameter (individual response and composite models), 9% for via wall angle (individual and composite models), and more than 100% for via resistance (individual and composite models). These improvements clearly demonstrate the effectiveness of the genetic optimization approach.

PROCESS MONITORING AND CONTROL

As consistent and cost-effective demands on semiconductor manufacturers to produce integrated circuits with higher density and complexity are prevalent, stringent process control is an issue of growing importance. Efficient and robust process control techniques require accurately monitoring the ambient process conditions for a given fabrication step. Historically, statistical process control (*SPC*) has been used to achieve the necessary level of control. This method is designed to minimize costly misprocessing by applying control charts to monitor fluctuations in critical process variables (64). Although SPC techniques detect undesirable process shifts, they are usually applied off-line. These techniques, therefore, cannot detect shifts until after the process step in question is complete. This delay results in fabricating devices that do not conform to specifications.

The objective of real-time SPC is to take advantage of available on-line sensor data from semiconductor fabricating to identify process shifts and out-of-control equipment states and generate real-time malfunction alarms, which offers the benefit of on-line process monitoring for generating at the very onset of a shift. The application of real-time SPC is complicated, however, by the correlated nature of the sensor data. SPC is based on assuming that the data to be monitored in controlling a process are identically independent and normally distributed (*IIND*). This assumption is not valid, however, when applied to real-time data. These data are often non-stationary (subject to mean and variance shifts), auto-correlated (dependent on data from previous time points), and cross-correlated (dependent on the values of other concurrently measured parameters).

In previous research efforts, Baker et al. addressed these difficulties by employing neural networks to develop time series models that filter cross-autocorrelation from real-time sensor data (65). Neural network-based control charts also previously demonstrated significant performance improvement over traditional Shewhart control charts in preventing Type II errors (i.e., missed alarms)

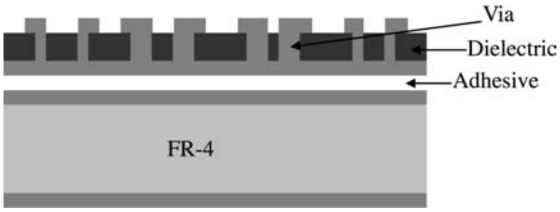


Figure 18. Test structure diagram (cross-sectional view) (17).

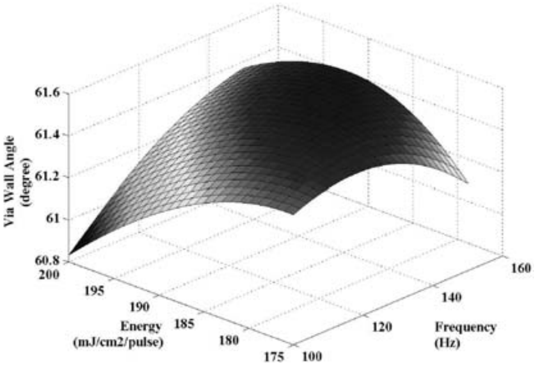


Figure 19. Effect of laser fluence and frequency on wall angle for 50-μm vias (17).

and detecting small process shifts (66). Such superiority was attributed to the ability of neural networks to learn arbitrary mappings of complex nonlinear data sequences, handle noisy and corrupted data, and simultaneously monitor multiple process variables. Furthermore, neural networks have been applied to predict the behavior of chaotic time series. Nelson et al. successfully employed ontogenic neural networks (i.e., those that modify their own topology during training) to predict continuously valued aperiodic functions, such as the Mackey–Glass equation (67). Mori and Ogasawara showed that recurrent neural networks model time series in short-term load forecasting of electrical power systems when statistically based models prove inadequate (68). Finally, wavelet neural networks (or "wavenets") have been used as a modified version of the wavelet transform to predict time series in signal processing (69). In applying this methodology to semiconductor manufacturing, Baker et al. (65)developed a real-time equipment monitoring system that transfers data from an reactive ion etching (*RIE*) system to a remote workstation.

Since neural networks excel in modeling processes with complex dynamics, they are also successfully applied to closed-loop control of a diverse array of such processes, including machining operations (70), lithographic color printing (71), plasma ion source control (72), and linear accelerator beam positioning (73). Recently, adaptive neuro-fuzzy neural networks were used as a technique for run-to-run process malfunction detection and diagnosis for an excimer laser ablation process (74). Neural nets are well suited to process control because they can be used to build predictive models from multivariate sensor data generated by process monitors.

In this section, the issues for process monitoring and control are addressed from two different perspectives: 1) monitoring the variation in manufacturing process conditions for real-time SPC using time-series data; and 2) process control schemes including run-by-run, real-time, and supervisory control schemes, which use *in situ* process sensors for on-line adjustments in process set points.

Time Series Modeling

Conventional SPC techniques are based on the assumption that the data generated by a controlled process is IIND. The IIND assumption, however, is not valid for applying control charts directly to data acquired in real time, because real-time data are non-stationary, auto-correlated, and cross-correlated. Time series modeling accounts for correlation in real-time data. The purpose of a time series model is to describe the chronological dependence among sequential samples of a given variable. Passing raw data through time series filters results in residual forecasting error that is IIND. Therefore, once an adequate time series model is developed, it can legitimately be used for SPC. One of the most basic time series models is the univariate Box–Jenkins autoregressive moving average (*ARMA*) model (75).

Data collected from modern semiconductor manufacturing equipment can also be represented by means of time series models, and Baker et al. showed that neural networks may be used to generalize the behavior of a time series (65). They referred to this new genre of time series model as the neural time series (*NTS*) model. Like statistical time series models such as ARMA, once an NTS model is developed, the forecast data can be used on conventional control charts. However, unlike the ARMA family of models, the NTS model simultaneously filters both auto- and cross-correlated data. In other words, the NTS model accounts for correlation among several variables being monitored simultaneously.

The neural network used to model the RIE process was trained off-line on data acquired when the process was un-

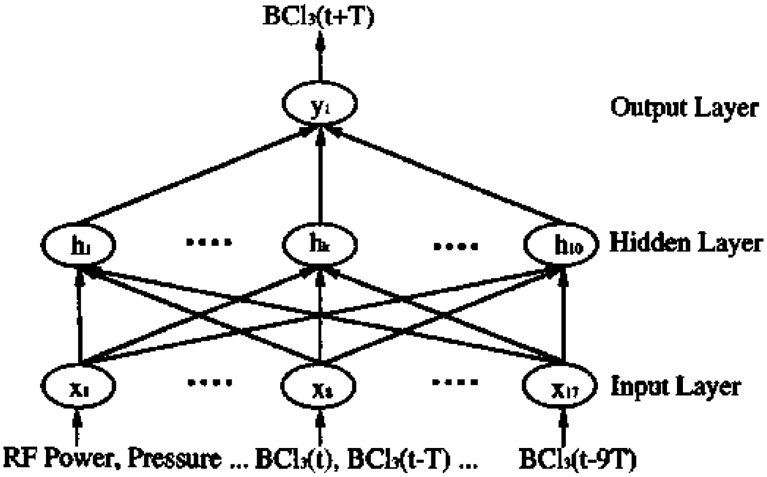


Figure 20. NTS network structure (65).

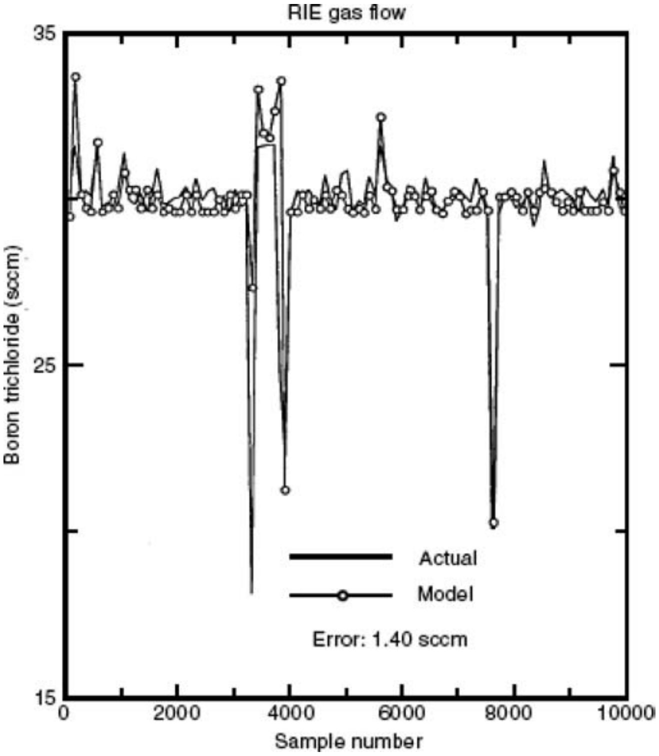


Figure 21. Measured BCl_3 flow and NTS model predictions (65).

der control. The parameter of interest was BCl_3, but the same methodology could be extended to any other process variable. The NTS network was trained to model BCl_3 flow by a unique sampling technique that involved training the network to forecast the next BCl_3 value from the behavior of 10 past values. The network was trained on a subset of the total auto-correlated data that consisted of the first 11 of every 100 samples. It was then tested on 11 midrange samples (samples 51–61, 151–161, etc.) of every 100 to quantify the performance of the trained network. Auto-correlation among consecutive BCl_3 measurements was accounted for by simultaneously training the network on the present value of the BCl_3 and 10 past values. Cross-correlation among the BCl_3 and the other six parameters was modeled by including as inputs to the NTS network the present values of the temperature, pressure, incident and reflected RF power, chlorine, and BCl_3 itself. The resulting network topology, therefore, had 17 input neurons, 10 hidden neurons, and a single output neuron (see Fig. 20). The future value of the BCl_3 at time (t + T) was forecast at the network output (where T is the sampling period). Figure 21 shows the measured and NTS model predictions of the BCl_3 data. Each point on the graph represents one out of every 100 samples, beginning with sample 61. (Recall that samples 51–61, 151–161, etc., were used as test data for the trained network). The NTS model very closely approximates the actual value. Even when drastic changes in the BCl_3 occur, the NTS network quickly adapted. This technique yielded an excellent root mean square error (*RMSE*) of 1.40 standard cm^3/min. This small error indicates that the sampling rate of 50 Hz was probably higher than actually required. In fact, because only 10% of this data was needed to build very accurate NTS models, the sampling rate could theoretically have been reduced as low as 5 Hz.

Run-By-Run Control

The main objective in run-by run control is to adjust fabrication process conditions on a wafer-by-wafer basis. These adjustments are made by comparing measured wafer characteristics and a predictive model of these characteristics. Smith and Boning integrated neural networks into the run-by-run control of chemical-mechanical polishing (*CMP*), a process in which semiconductor wafers are planarized using a slurry of abrasive material in an alkaline or acidic solution (76). CMP exhibits unique characteristics (such as drift in removal rate, memory effects, and varying amounts of process noise) that make this process ideal for control applications. Smith and Boning trained a neural network to map CMP process disturbances to optimal values for the coefficients in an exponentially weighted moving average (*EWMA*) controller (64). Statistical experimental design was used to generate a linearized multivariate model of the form

$$y_t = Ax_t + c_t \tag{25}$$

where t is the run number, y_t is a vector of process responses, A is a constant gain matrix, x_t is vector of process inputs, and c_t is an offset vector, which is calculated recursively by an EWMA controller from the following relationship

$$c_t = \alpha(y_t - Ax_t) + (1-\alpha)c_{t-1} \tag{26}$$

The coefficient α is dynamically estimated from the neural network mapping according to the algorithm outlined in Fig. 22. In designing this system, these researchers developed a self-tuning EWMA controller that dynamically updates its parameters by estimating the disturbance using the neural network mapping, which resulted in an adaptive run-by-run controller that virtually eliminates the need for an experienced engineer to provide EWMA tuning.

The neural network enhanced run-by-run control strategy was also pursued by Wang and Mahajan of the University of Colorado, who similarly integrated neural nets and SPC for the control of a chemical vapor deposition (*CVD*) process (32). These authors also trained a neural network to map the input–output relationships of this process with data from a designed experiment. Then a controller model was extracted from the neural network mapping by using the EWMA technique to filter process output noise and detect process shifts or drift. The controller used feedback to tune the CVD input settings to compensate for the shift/drift detected. Wang and Mahajan showed that this approach outperforms other run-by-run control systems that do not involve neural networks, such as that proposed by Butler and Stefani (77).

Real-Time Control

The next evolutionary step in neuro-control involves using neural nets to continuously correct process conditions, as opposed to making run-by-run adjustments. This real-time control approach has been pursued by Rietman et al. of Bell Laboratories, who designed a neural network to compute in real time the over-etch time for a plasma gate etch step (78). This time computation was based on a neural network mapping of the mean values of fluctuations about control variable set points and an in situ optical emission monitor. By monitoring a single optical emission wavelength during etching, these researchers inferred information about etch rate, etch uniformity, pattern density, and cleanliness of the reaction chamber. In neural network training, vectors representing process "signatures" inherent in the emission trace and set points were mapped to the ideal etch time for a desired oxide thickness. This training procedure is illustrated in Fig. 23. The BP network for the control operation consisted of 36 input nodes, five hidden neurons, and one output.

This system was learning on-line from 1993 until about 1998. During this time, the network was trained on many thousands of wafers. After months of close observation, the network was eventually allowed independent control of a production etcher, which eliminated the need for human intervention in determining the proper over-etch time. In the opinion of the Bell Labs engineers, in addition to reducing process variation, increasing yield, and reducing manufacturing cost, this functional adaptive controller can potentially extend the useful life of the processing equipment because design rules continue to shrink and greater demands are constantly being placed on equipment performance.

Recently, May and Stokes at Georgia Tech developed a real-time, model-based feedback control scheme for reactive ion etching (*RIE*) using neural networks (9, 10). This scheme was pursued to construct a predictive model for RIE systems that can be approximately inverted to achieve the desired control using indirect adaptive control (*IAC*) strategy. The IAC structrure shown in Fig. 24 includes a neural control (*NC*) and plant emulator (*PE*), which are impementd as two separate back-propagation neural networks. In the IAC approach, the plant emulator is trained off-line with experimental data, whereas the controller is trained on-line with feedback from the plant emulator. Conventional IAC schemes require direct feedback of process variables from the plant to adjust the plant emulator. The neural controller (*NC*) adjusts the PE's inputs in real time to optimally match the output of the PE (y_e) to the control target (y^*).

To train the neural controller, the control target is first fed through the neural controller to the plant emulator to obtain the process output $y_e(t)$. Second, using the generalized delta rule, the error between the control target and PE output $[e_2 = y^*(t) - y_e(t)]$ is back-propagated through the plant emulator to calculate the weight adjustments for each layer of the PE. Next, the computed changes in the plant emulator's inputs are used to estimate the output error of the neural controller. Finally, the neural controller's weights are updated using the BP. This cycle is repeated for each successive control target $[y^*(t+1), y^*(t+2)$, etc.].

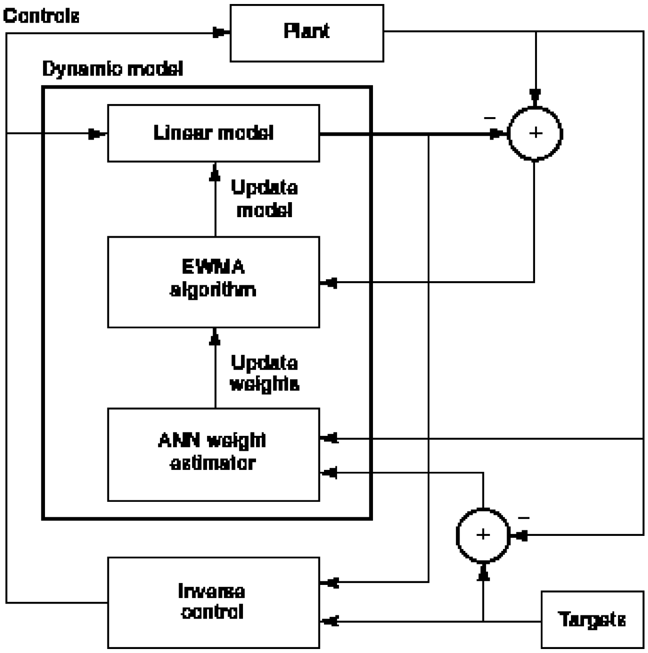


Figure 22. EWMA controller with neural network weight estimator (76). .

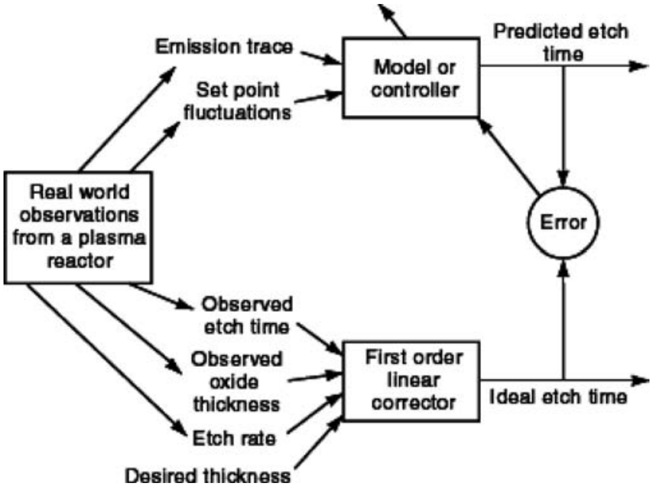


Figure 23. Illustration of training method for wafer-to-wafer neural network control of a plasma gate etch (78).

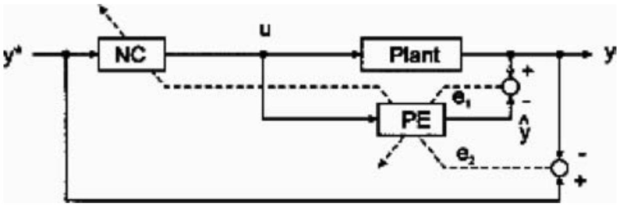


Figure 24. Illustration of adaptive process control scheme using two back-propagation neural networks: a plant emulator (PE) and a neural controller (NC) (9).

To evaluate this scheme, Stokes and May performed real-time control simulations for a SiO_2 plasma etch experiment using a simplified IAC structure (9). It was shown that the neural controller can be adjusted to quickly track changes in target values, effectively inverting (or approximately inverting) the model for the RIE plant. Based on this previous success, the neural network controller was applied to the etching of a GaAs/AlGaAs heterostructure in a BCl_3/Cl_2 plasma by a Plasma Therm 700 SLR series RIE system (10). A multiple-input, multiple-output (*MIMO*) approach to simultaneously control etch rate and DC bias was investigated. Real-time sensor feedback in the form of pro-

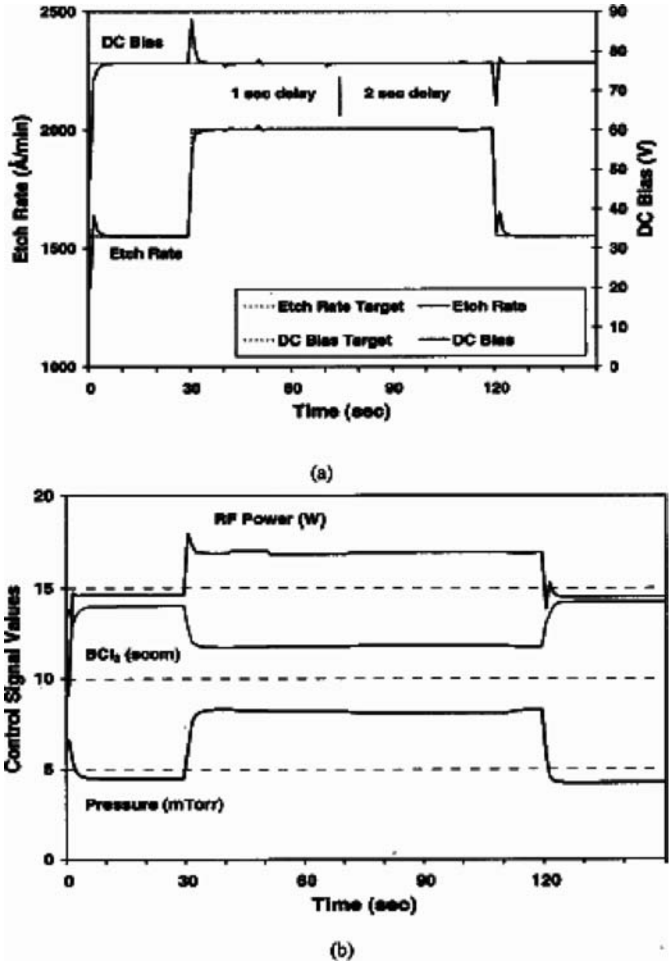


Figure 25. The performance of IAC scheme under variable dead-time: (*a*) plant output and (*b*) control signals (9, 10)).

cess conditions and residual gas analysis (*RGA*) was collected to facilitate control over the etch depth. An example of the performance of the IAC scheme under variable dead-time is shown in Fig. 25. The results in Fig. 25 show that a time delay of 1 second had very little effect on the performance of the system, whereas a 2-second delay caused only a slight delay in the recovery time. The control signals from the IAC were nearly identical to those in the set-point control case with some slight overshoot in the RF power as it rises to return the DC bias to its target when a delay of 2 seconds was present. Overall, this neural network controller exhibited improved set-point tracking, disturbance rejection, response to changes in RIE dynamics, and response to variable dead time. These results indicate that in every case, the neural controller converges very quickly, providing evidence that the dynamic characteristics of the RIE process are indeed learned by on-line training. The controller also adjusted the plant emulator's inputs under noisy conditions to approximately match the target. The methodologies developed are generally applicable to semiconductor manufacturing processes.

Supervisory Control

A run-by-run control system that involves both feed-forward and feedback control schemes is known as a supervisory control system. Control of semiconductor processes can be examined at several levels (79). Supervisory control is the highest level of the hierarchy shown in Fig. 26. At this level, the progression of a wafer is tracked from unit process to unit process, and adjustments can be made to subsequent steps to account for variation in preceding steps. Both feedback and feed-forward adjustments are made in a supervisory control system. As an example, Patel et al. presented a scheme for supervisory control of deposition time and temperature for low pressure chemical vapor deposition (*LPCVD*) grown silicon nitride off product wafer using a Kalman filter-based estimation (80). During the response model was constructed, stability of the feedback loop to modeling error was quantified and an iterative algorithm was proposed for tracking batch data and updating data from batch to batch. Finally, the controller is applied for high volume 300-mm manufacturing on a TEL Alpha 3031 vertical furnace.

The concept of intelligent modeling techniques such as neural networks can also be applied to supervisory control systems. As an example of such a system, Kim has devel-

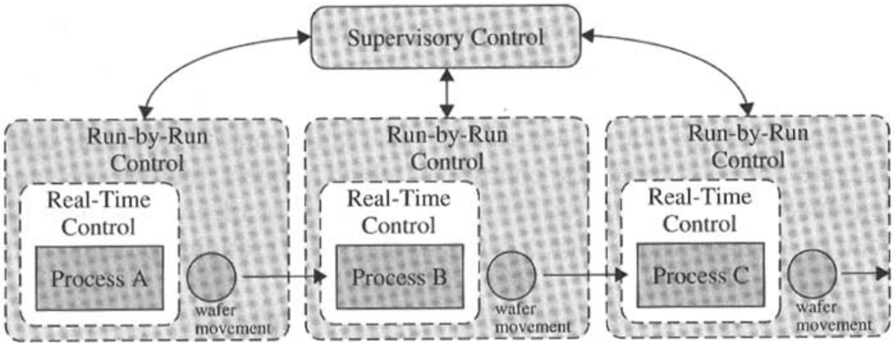


Figure 26. Process control hierarchy (79).

oped a model-based supervisory control algorithm based on computational intelligence techniques and applied this approach to reduce undesirable behavior resulting from various process disturbances in via formation in a photolithography sequence (81). Kim and May (45)presented a modeling approach for via formation in dielectric layers composed of photosensitive benzocyclobutane (*BCB*) based on the mapping capabilities of neural networks. A series of designed experiments were performed to characterize the complete via formation workcell (i.e., each unit process step for via formation). Using the sequential modeling scheme described earlier, each workcell sub-process is modeled individually, and each sub-process model is linked to previous sub-process outputs and subsequent sub-process inputs (see Fig. 16). The goal in this study was to develop a supervisory process control system for via formation to maintain system reliability in the face of process disturbances. Supervisory control can reduce variability in two ways. The first involves reducing the variability of each contributing step by feedback control. The second requires accounting for the variation of consecutive steps so that their deviations cancel each other by feed-forward control. In this system, dielectric film thickness and refractive index were used as process monitors for each sub-process, and via yield, film retention, and film non-uniformity were added as the final response characteristics to be controlled. Based on appropriate decision criteria, model and recipe updates for consecutive sub-processes were determined.

Figure 27 shows the general flowchart of the supervisory control scheme. Nine neural networks were required: one global process model for optimal process recipe synthesis, four models for each sub-process model, and four for recipe updates to realize the supervisory algorithm. To construct the process supervisor, recipe update modules were developed individually for each sub-process. The neural networks for recipe update modules are trained off-line and updated on-line as necessary. Based on the neural networks used for recipe updates, genetic algorithms generate optimal process recipes for the next sub-process.

When the supervisory control algorithm was applied to a real via formation process, experimental results showed significant improvement in film thickness and via yield control as compared with open-loop operation. Table 7 compares the final responses of the process with and without control. The "% improvement" column in this table is calculated using

$$\%\,\text{Improvement} = \frac{(R_{WOC} - R_{WC})}{(R_{WOC} - T)} \times 100 \quad (5\text{-}3) \qquad (27)$$

where $R_{\text{WOC}}, R_{\text{WC}}$, and T represent process response without control, process response with control, and control target value, respectively. These results showed that the supervisory control system significantly increased via yield and the final film thickness was very close to the control target compared with the result of the experiment without control.

PROCESS DIAGNOSIS

Product quality assurance throughout a semiconductor manufacturing facility requires the strict control of literally thousands of process variables. These variables serve as input and output parameters for hundreds of distinct process steps. Individual process steps are conducted by sophisticated and expensive fabrication equipment. A certain amount of inherent variability exists in this equipment regardless of how well the machine is designed or maintained. This variation is the result of numerous small and essentially uncontrollable causes. However, when this variability becomes large compared with background noise, significant performance shifts may occur. Such shifts are often indicative of equipment malfunctions. When unreliable equipment performance causes operating conditions to vary beyond an acceptable level, overall product quality is jeopardized. Consequently, fast and accurate equipment malfunction diagnosis is essential to the success of the semiconductor product process.

This section presents several approaches for the malfunction detection and diagnosis of IC fabrication equipment. The methodologies discussed here include quantitative malfunction detection and diagnosis using standard methods as well as neural network-based malfunction detection and diagnosis using pattern recognition. The use of malfunction detection and diagnosis in equipment, process, and circuit level can allow us to maintain consistent manufacturing processes, increasing the probability of identifying faults caused by equipment malfunction, and ultimately leading to yield improvement.

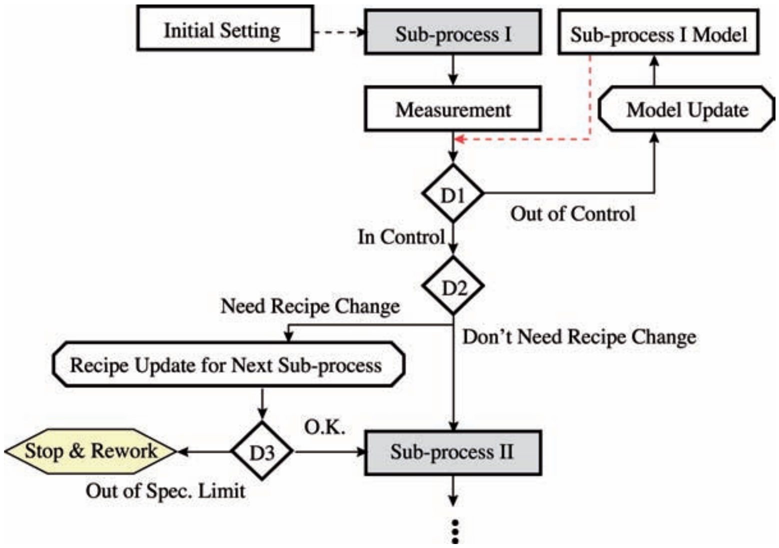


Figure 27. Flowchart of supervisory control algorithm (81).

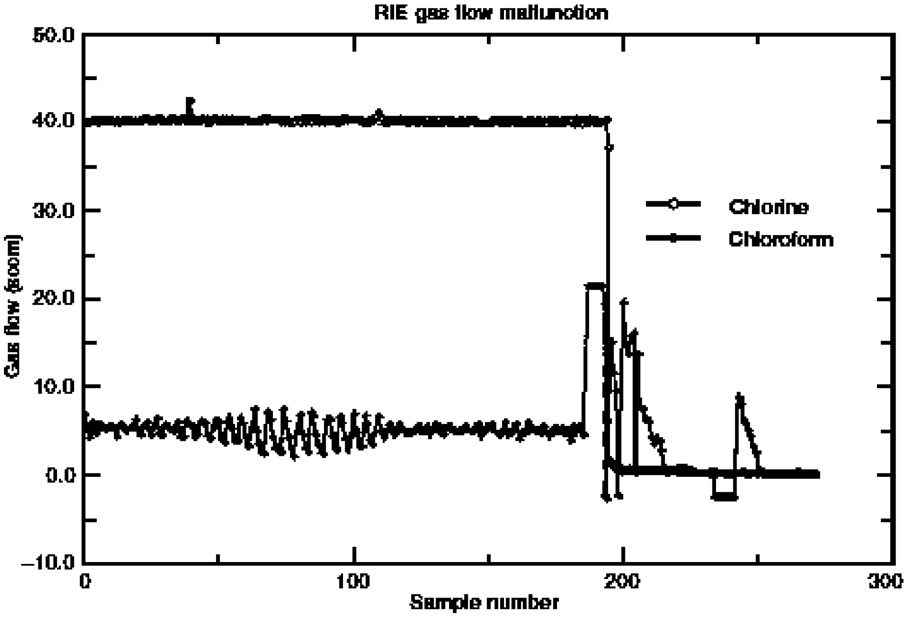


Figure 28. Chlorine and CHF_3 flow rates for Al etch step just before an equipment malfunction (65).

Malfunction Detection

The NTS model (see time series modeling section) is used to generate a real-time alarm signal when sampled process data do not conform to their previously established pattern, indicating a possible equipment malfunction or other out-of-control state. This capability was demonstrated on an actual RIE malfunction (61). In this case, aluminum was etched in a CHF_3 and chlorine gas mixture. The malfunction consisted of an unstable feed condition in the CHF_3 mass flow controller. Figure 28 is a plot of the gas flows during the period leading up to the malfunction. Although the Cl_2 flow appears to fall out of compliance at the 200th sample, it was not the cause of the malfunction. The true cause may be discerned by observing the behavior of the CHF_3 several samples earlier and comparing the instability of its flow with the more stable and consistent readings exhibited by the Cl_2 during the same time span. A careful study of this situation reveals that the CHF_3 mass flow controller was not able to regulate the gas flow correctly, and consequently the RIE control circuitry aborted the process, thus causing the Cl_2 to shut off.

The on-line application of the NTS model was used to generate an alarm signal warning of the impending out-of-control condition of CHF_3 flow even before the RIE aborted itself. Recall that the NTS model acts as a filter to remove auto-correlation and cross-correlation from the raw process data. Thus, the residuals that result from computing the difference between NTS model predictions and the measured values of the CHF_3 flow are IIND random variables. As a result, these residuals can be plotted on a standard Shewhart control chart to identify process shifts, in which case alarm generation was based on the well-known Western Electric Rules, summarized here (64).

1. One data point plots outside of the 3-sigma control limits.
2. Two of three consecutive points plot beyond the 2-sigma warning limits.
3. Four of five consecutive points plot 1-sigma or beyond from the center line.
4. Eight consecutive points plot on one side of the center line.

Although this malfunction eventually broke all of these rules, the violation of Rule 4 was invoked to generate the malfunction alarm. The data from the RIE malfunction was fed into the NTS network with CHF_3 as the forecast parameter. Figure 28 demonstrates that once again the NTS model closely resembled the actual data sequence until the malfunction occurred, at which point the CHF_3 instability became too great and the NTS model predictions diverged from the measurements. Figure 29 shows the measurement residuals resulting from the difference between the NTS model predictions and the actual sensor data. When eight consecutive points in the data sequence plotted on one side the center line (which occurred at the 18th sample), the NTS network immediately responded by signaling an alarm.

At the point where the NTS alarm is generated, the value of the mean shift in CHF_3 flow is merely 0.25σ, which indicates that the NTS model is quite sensitive to small shifts. For the same malfunction, the internal RIE process control circuitry did not respond until significantly later (at about the 170th sample). The rapid NTS response time can be instrumental in identifying incipient equipment faults and preventing subsequent misprocessing, which illustrates an important tradeoff that occurs when the proper data sampling rate is chosen. Although the chosen rate of 50 Hz proved unnecessary to build an accurate NTS model, this high rate ensures that malfunction detection is nearly immediate.

Malfunction Diagnosis

Neural networks have been widely used in process monitoring and diagnosis (82), primarily in mechanical machining operations, such as cutting or injection molding. For example, Burke and Rangwala discussed a neural network approach for tool conditioning in metal cutting (83). Wasserman et al. used neural networks to detect and measure small cracks in rotating machine shafts (84). Recently, neural nets have also begun to find use in electronics systems diagnosis. Murphy and Kagle used the pattern identification capabilities of neural networks to recognize electronic malfunctions (85). Using neural nets for process diagnosis in semiconductor manufacturing has also started to gain attention. The approaches undertaken by researchers in this area include diagnosis at three distinct levels of the manufacturing process: 1) the equipment level, 2) the process level, and 3) the circuit level.

Equipment Level. Kim and May successfully employed a hybrid scheme that involves neural networks in tandem with traditional expert systems to develop a working prototype for real-time, automated malfunction diagnosis of IC fabrication equipment. Hybrid techniques effectively offset the weaknesses of each individual method by itself (86). Traditional expert systems excel at reasoning from previously viewed data, whereas neural networks extrapolate analyses and perform generalized classification for new scenarios. Kim and May's system has been implemented on a Plasma Therm 700 series RIE to outline general diagnostic strategy applicable to other rapid single-wafer processes. Diagnostic systems that rely on post-processing measurements and electrical test data alone cannot rapidly detect process shifts and also identify process faults. As unreliable equipment jeopardizes product quality, it is essential to diagnose the root causes for the malfunctions quickly and accurately. May and Spanos have previously developed a real-time diagnostic system that integrates evidence from various sources using the Dempster–Shafer rules of evidential reasoning (87).

Extending this work, Kim and May integrated neural networks into this knowledge-based expert system (88). Diagnosis is conducted by this system in three chronological phases: the maintenance phase, the on-line phase, and the in-line phase. Neural networks were used in the maintenance phase to approximate the functional form of the failure history distribution of each component in the RIE system. Predicted failure rates were subsequently converted to belief levels. For on-line diagnosis of previously encountered faults, hypothesis testing on the statistical mean and variance of the sensor data was performed to search for similar data patterns and assign belief levels. Finally, neural process models of RIE figures of merit (such as etch or uniformity) were used to analyze the in-line measurements and identify the most suitable candidate among potentially faulty input parameters (i.e., pressure, gas flow, and so on) to explain process shifts. Hybrid neural expert systems offer the advantage of easier knowledge acquisition and maintenance and extracting implicit knowledge (through neural network learning) with the assistance of explicit expert rules. The only disadvantage in neural expert systems is that, unlike other rule-based systems, the somewhat non-intuitive nature of neural networks makes it difficult to provide the user with explanations about the way diagnostic conclusions are reached (3). However, these barriers are lessening as more and more successful systems are demonstrated and become available. It is anticipated that the coming decade will see neural networks integrated firmly into diagnostic software in newly created fabrication facilities.

More recently, Hong and May explored a methodology for real-time malfunction diagnosis of RIE employing optical emission spectroscopy (*OES*) and residual gas analysis (*RGA*) data (89). Based on this metrology data, time series neural networks (TSNNs) were trained to generate evidential belief for potential malfunctions in real time, and Dempster-Shafer theory was adopted for evidential reasoning. Modeling using the TSNN was accomplished in two steps: fault detection for a single faulty component and fault detection on multiple components (39). The struc-

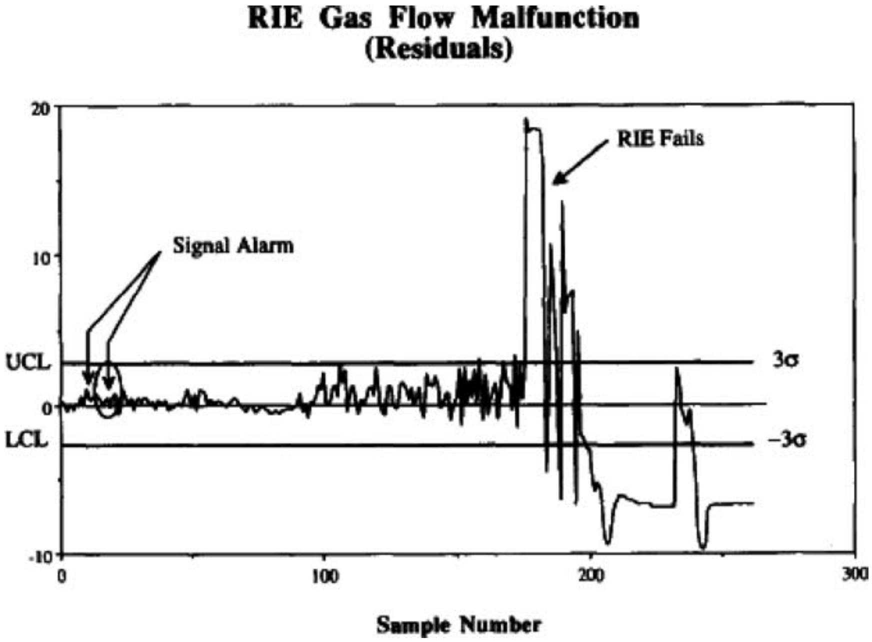


Figure 29. Measurement residuals from NTS model before RIE malfunction plotted on 3-sigma control chart (65).

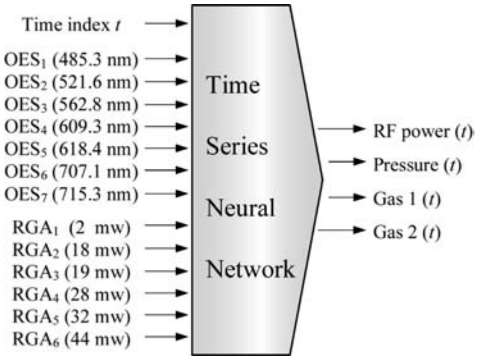


Figure 30. A schematic structure of the TSNN for fusion and prediction of data from OES and RGA sensors (39).

ture of the TSNN is shown in Fig. 30. The TSNN models trained using OES and RGA sensor data were shown to be effective for monitoring and diagnosis for RIE systems. This approach contributed to maintaining a consistent RIE process by successfully detecting faults with only a single missed alarm and a single false alarm occurring out of 21 test runs when both sensors were used in tandem.

Process Level. In another diagnostic application, Sikka of Intel's Artificial Intelligence Laboratories in Santa Clara, California used BP neural networks for wafer map analysis (90). To do so, a technique was developed to detect and characterize spatial features on gray-scale cumulative wafer maps acquired at the final wafer sort step. These cumulative maps are obtained by summing the contents of several individual wafer maps, each consisting of the pass/fail status of each tested die on the wafer. Defects from certain process steps produce characteristic spatial features on the cumulative maps. The Intel wafer map analyzer (*WMA*) software combines standard image processing (to enhance features and extract specific attributes) with neural networks (to determine categories and locations of the extracted attributes) to reduce the need for impractical and lengthy visual wafer inspection. In so doing, this system, accurate to nearly 100%, assists with diagnostic troubleshooting by providing warning signs of potential equipment failures in key process steps.

Recently, Setia and May investigate in-line fault detection and diagnosis of excimer laser ablation process using computational intelligent methodologies such as a combination of feed-forward neural networks and Dempster-Shafer theory and adaptive neuro-fuzzy networks (73). Both methodologies employ response data originating directly from the laser equipment and characterization of microvias formed by the ablation process, which serves as evidence of equipment malfunctions affecting process parameters. The system based on neural networks operating in conjunction with Dempster–Shafer theory performed more accurately in the failure detection task (i.e., 100% detection in 19 possible scenarios[9]) as compared with the neuro-fuzzy networks, which generated one false alarm. Furthermore, both neural networks in conjunction with

Dempster–Shafer theory and neuro-fuzzy networks thus achieve approximately 95% and 90% success in diagnosis, respectively.

Circuit Level. At the integrated circuit level, Plummer has developed a process control neural network (*PCNN*) to identify faults in bipolar operational amplifiers (or op-amps) based on electrical test data (91). The PCNN exploits the capability of neural nets to interpret multidimensional data and identify clusters of performance within such a data set, which provides enhanced sensitivity to sources of variation that are not distinguishable by observing traditional single-variable control charts. Given a vector of electrical test results as input, the PCNN can evaluate the probability of membership in each set of clusters, which represent different categories of circuit faults. The network can then report the various fault probabilities or select the most likely fault category.

Representing one of the few cases in semiconductor manufacturing in which back-propagation networks are not employed, the PCNN is formed by replacing the output layer of a probabilistic neural network with a Grossberg layer (Fig. 31). In the probabilistic network, input data is fed to a set of pattern nodes. The pattern layer is trained using weights developed with a Kohonen self-organizing network. Each pattern node contains an exemplar vector of values corresponding to an input variable typical of the category it represents. If more than one exemplar represents a single category, the number of examples reflects the probability that a randomly selected pattern is included in that category. The proximity of each input vector to each pattern is computed, and the results are analyzed in the summation layer.

The Grossberg layer functions as a lookup table. Each node in this layer contains a weight corresponding to each category defined by the probabilistic network. These weights reflect the conditional probability of a cause belonging to the corresponding category. Then outputs from the Grossberg layer reflect the products of the conditional probabilities. Together, these probabilities constitute a Pareto distribution of possible causes for a given test result (which is represented in the PCNN input vector). The Grossberg layer is trained in a supervised manner, which requires that the cause for each instance of membership in a fault category must be recorded beforehand.

Despite its somewhat misleading name, Plummer applied the PCNN in a diagnostic (as opposed to a control) application. The SPICE circuit simulator was used to generate two sets of highly correlated input/output operational amplifier test data, one representing an in-control process and the other a process grossly out of control. Although the second data set represented faulty circuit behavior, its descriptive statistics alone gave no indication of suspicious electrical test data. Training the Kohonen network with electrical test results from these data sets produced four distinct clusters (representing one acceptable and three faulty states).

With the Kohonen exemplars serving as weights in the pattern layer, the PCNN then was used to identify one of the three possible out-of-control conditions: 1) low npn β; 2) high npn β and low resistor tolerance; or 3) high npn β and high resistor tolerance. The summation layer of the PCNN reported the conditional probability of each of these conditions and the probability that the op amp measurements were acceptable for each input pattern of electrical test data. The PCNN was 93% accurate in overall diagnosis, and correctly sounded alarms for 86% of the out-of-control cases (no false alarms were generated).

YIELD MODELING

Yield modeling is of the highest importance in semiconductor manufacturing. The technical metrics of manufacturing performance typically include product yield, functional performance, parametric performance, facility throughput, and average cycle time. Continuous improvements in manufacturing yield require a strong commitment to quality management as well as equipment maintenance. Optimizing each of these creates a benchmark for properly executing complex manufacturing processes. However, modeling each also brings corresponding technical and scientific challenges.

Parametric Yield

Declining manufacturing yields have been attributed to increasing complexity and stringent process restrictions. With newly developed or highly specialized processes, parametric yield loss, which can be attributed to defects, foreign particles, and random variations in the fabrication process, is particularly important. Parametric yield, or the percentage of devices that meet a set of reasonable constraints, can be challenging to improve even in a defect-free manufacturing environment. Although subtle process fluctuations may not always cause catastrophic failures, they often prevent devices from meeting certain performance specifications. ICs are often categorized according to specific performance criteria. Therefore, it remains critical to develop methodologies for modeling parametric performance (92). Methodologies that take advantage of artificial intelligence tools offer promising solutions to key manufacturing issues.

The Monte Carlo method has been a common approach for evaluating parametric yield. It uses a large number of pseudo-random sets of values of circuit parameters that are generated according to the distribution drawn from measured data. Using the Monte Carlo approach, a simulation is performed for each set of parameters, and information is extracted regarding the predicted performance of a circuit. Then, the performance distribution from the set of simulations can be determined. Unfortunately, the Monte Carlo approach has several drawbacks. The most obvious shortcoming is the large number of simulations it requires, which makes this approach computationally expensive. In a purely random Monte Carlo simulation, each device parameter is varied independently, which subsequently ignores the correlated nature of device parameters. Monte Carlo simulations also assume a specific statistical distribution *a priori* to randomly generate sets of device and/or process parameters. Although it may be suitable for a large, well-characterized fabrication process, newly developed processes can exhibit nonstandard statistical be-

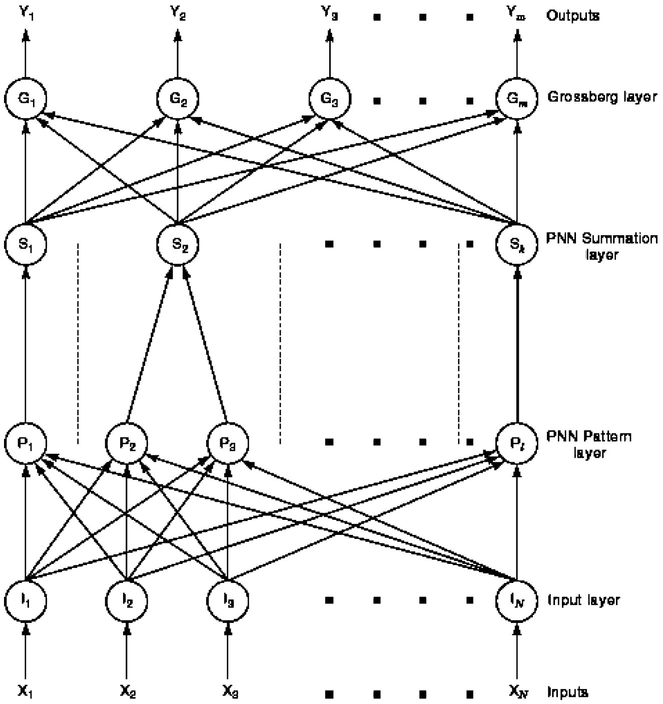


Figure 31. Process control neural network from Reference (91).

havior. In fact, the distribution of parameters from newly developed processes can possess significant skew or extreme deviations, or may not be normal at all (93).

An alternative systematic methodology that allows device yield prediction before high volume manufacturing to evaluate the impact of both design decisions and process capability has been demonstrated (92). This methodology computes the circuit parametric yield numerically from integrals of the form

$$\int p(y)dy \tag{31}$$

where y is a particular device performance characteristic and $p(y)$ is its probability density function (pdf). This pdf is derived by: 1) measuring or simulating a significant sample of device parametric data; 2) using neural networks to encode the probability density function of all marginal pdfs of the measured parameters; and 3) computing $p(y)$ directly from the joint pdf using a standard mathematical transformation.

Yun and May used this technique to model parametric yield for avalanche photodiodes (*APD*) grown by MBE. The input factors were the following variables: device diameter, mean doping, standard deviation of doping, and barrier width. The performance parameters were APD gain and noise. The input factors were chosen because of their potential for variation in a manufacturing setting leading to possible impact on yield. For example, device diameter could vary because of photolithographic variations including misalignment, and the other input factors can fluctuate in the molecular beam epitaxy system used to synthesize the APD structures.

In this study, two BP neural networks were used to train and predict APD gain and noise. Inputs to the neural network models were the four manufacturing process variables. Afterward, the functional form of the overall joint parameter distribution directly from measured data was determined using neural networks. In this case, the network inputs were the manufacturing parameter values, and the network output was their corresponding relative frequency. As neural networks are useful for input–output mapping, the functional form of the joint pdf was encoded in the neural network. Once the pdf of the device parameters was computed, the joint pdf for functions of these parameters were derived. The systematic methodology is detailed below.

Let us consider two sets of random variables X_j (representing the manufacturing parameters) and Y_i (representing the performance metrics), where the Y_is are functions of the X_js

$$x_1 = A; \quad x_2 = B; \quad y_1 = G; \quad y_2 = N \tag{32}$$

The functional relationship between the manufacturing process variables and performance parameters can be expressed as

$$\begin{aligned} y_1 &= H_1(x_1, x_2) \\ y_2 &= H_2(x_1, x_2) \end{aligned} \tag{33}$$

where H_1 and H_2 are continuous, differentiable functions. Now x_1 and x_2 can be solved in terms of y_1 and y_2 to obtain

$$\begin{aligned} x_1 &= G_1(y_1, y_2) \\ x_2 &= G_2(y_1, y_2) \end{aligned} \tag{34}$$

where G_1 and G_2 are also continuous and differentiable. The joint pdf of random variables Y_1 and Y_2, $u(y_1, y_2)$ is given by

$$u(y_1, y_2) = f(x_1, x_2)|J(x_1, x_2)| \tag{35}$$

where is the joint pdf of x_1 and x_2, and is the Jacobian transformation. The Jacobian is given by the following determinant:

$$J(y_1, y_2) = [\begin{matrix} \frac{dx_1}{dy_1} & \frac{dx_1}{dy_2} \\ \frac{dx_2}{dy_1} & \frac{dx_2}{dy_2} \end{matrix}] \tag{36}$$

Once $u(y_1, y_2)$ was calculated, the marginal densities of the device performance parameters (noise and gain) can be calculated as follows:

$$\begin{aligned} I_1(y_1) &= \int u(y_1, y_2)dy_2 \approx y_2 \sum u(y_1, y_2) \\ I_2(y_2) &= \int u(y_1, y_2)dy_1 \approx y_1 \sum u(y_1, y_2) \end{aligned} \tag{37}$$

where $I_1(y_1)$ and $I_2(y_2)$ are the marginal pdfs of the performance characteristics and the numerical integration is performed by the trapezoid rule.

Using this methodology, the parametric yield of gain and noise in APD was predicted based on the variation of the manufacturing parameters. The results from this method were compared with Monte Carlo results (see Figs. 32 and 33).

The Monte Carlo method performed without considering that the variety of the input parameter distributions could not accurately predict the parametric yield. Alternatively, the results from this alternative approach employed by Yun and May were comparable with results achieved using the Monte Carlo method that does consider different input distributions and were also obtained with significantly fewer simulations.

Design Centering

Design centering is essentially an approach to optimize yield (94). In a production line with severe process variations, the number of unqualified circuits is great, which is disadvantageous for the foundry and the consumer. The obvious goal is to maximize the number of qualified circuits whose performance meets the specifications of the customer. As circuits get smaller, design centering requires tighter control in the manufacturing process, which means precise tool alignment and nominal values and tolerances of layout parameters.

In an effort to maximize yield, it is beneficial to explore the parameter space for the optimum designed layout. However, devices are smaller and more complex; therefore, the budget for exploring the large parameter space is expectedly inadequate. Monte Carlo simulations (95)and geometric methods (96)have been previously employed, and each has it benefits. As described above, Monte Carlo simulations are computationally intensive and require a large number of simulations. Geometric methods become a major undertaking as the dimension of the problem increases. Given the time constraints and other restrictions at a production facility, alternative approaches that quickly located optimal layouts are essential. For example, Pratap et al. (97)employed a two-stage neuro-genetic design centering scheme: 1) *parametric yield estimation* through the use of neural networks and 2) *design centering* using genetic algorithms.

In stage 1, the parametric yield estimation was performed using Monte Carlo simulations based on neural network models, which began with a random sample generator that used Monte Carlo runs to generate a large number of input vectors based on the mean, variance, and distribution of the input variables (i.e., heterojunction bipolar transistor emitter length, collector doping, base doping, and emitter doping). Neural network models were used to calculate output parameters: maximum gain (β) and peak cutoff frequency (f_T). Once the output values are determined for each run, the yield was determined using a yield calculator. Parametric yield was calculated based on the upper and lower specifications for each output. The yield for each individual output is defined by

$$\begin{aligned} Y_1 &= \{y|y_{1\min} \le y_1 \le y_{1\max}\} \\ Y_2 &= \{y|y_{2\min} \le y_2 \le y_{2\max}\} \\ &\vdots \\ Y_i &= \{y|y_{i\min} \le y_i \le y_{i\max}\} \\ &\vdots \\ Y_n &= \{y|y_{n\min} \le y_n \le y_{n\max}\} \end{aligned} \tag{38}$$

where Y_i is the partial yield of the ith output, $Y_{i\min}$ and $Y_{i\max}$ are lower and upper specifications, respectively, and n is the number of output values. The total yield of the device is defined by

$$Y = Y_1 \cap Y_2 \mathcal{K} \cap Y_i \mathcal{K} \cap Y_n \tag{39}$$

Here, fixed mean values, variances, and distribution types of each process variable are provided to the parametric yield calculator along with desired specification limits of the outputs.

The second stage of the algorithm used the parametric yield estimator in conjunction with GAs to determine the means and variances of the input parameters that result in the maximum yield. In this step, 1) the distribution of the input variables is assumed to be independent of its mean and variances; 2) the input parameters are assumed to be statistically independent; and 3) the variance is assumed to be independent of means. The GA begins with an initial population of means and variances of input parameters. The parametric yield estimator calculates the yield for each member of the population. If the yield of any population exceeds the desired maximum yield, that particular sample is deemed the design center, and the algorithm ceases. Alternatively, the population of means and variances is provided to the GA block along with the corresponding parametric yield values, and the algorithm performs genetic manipulations to obtain a new population of means and variances. During genetic manipulation, the samples with higher yield are assigned greater fitness values, leading to a higher probability of survival in the new population set. The process continues iteratively until a suitable design is achieved. Results from this methodology are illustrated in Figs. 34 and 35.

Figure 34 shows the yield histogram of f_T before design centering, and Fig. 35 illustrates the improvement in parametric yield for 30-GHz devices (from 25% to 75%). The maximum gain improved in a comparable manner. Similar results (Figs. 36 and 37) were also obtained for 30-GHz voltage controlled oscillators (*VCO*). In Fig. 36, a large pro-

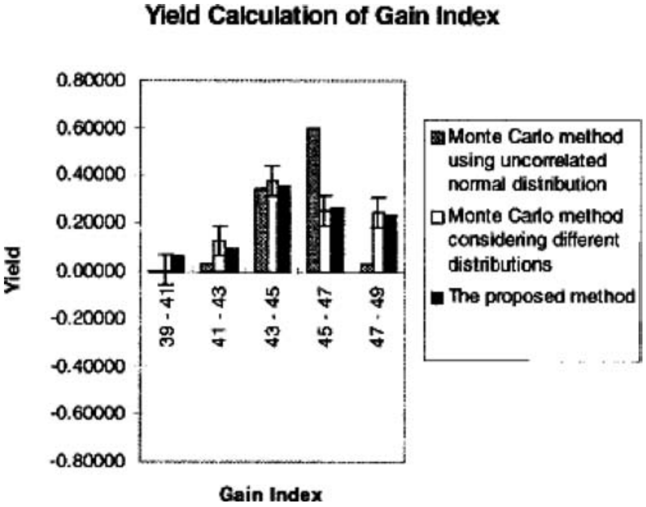


Figure 32. Comparison of yield calculations of gain index obtained from Monte Carlo and method employed by Yun and May (92).

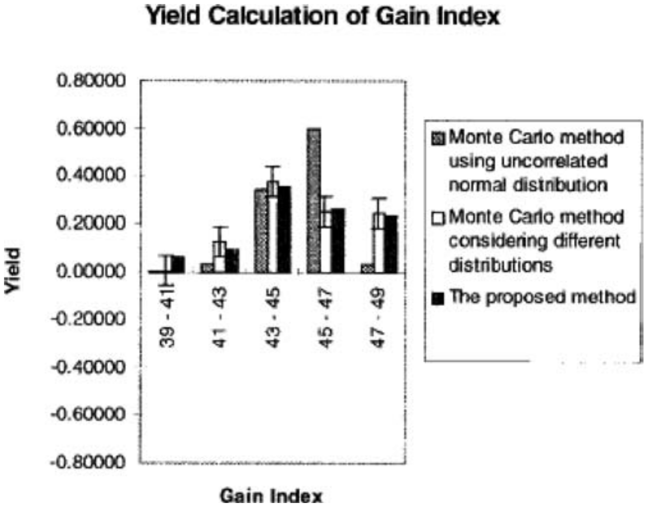


Figure 33. Comparison of yield calculations of noise index obtained from Monte Carlo and method employed by Yun and May (92).

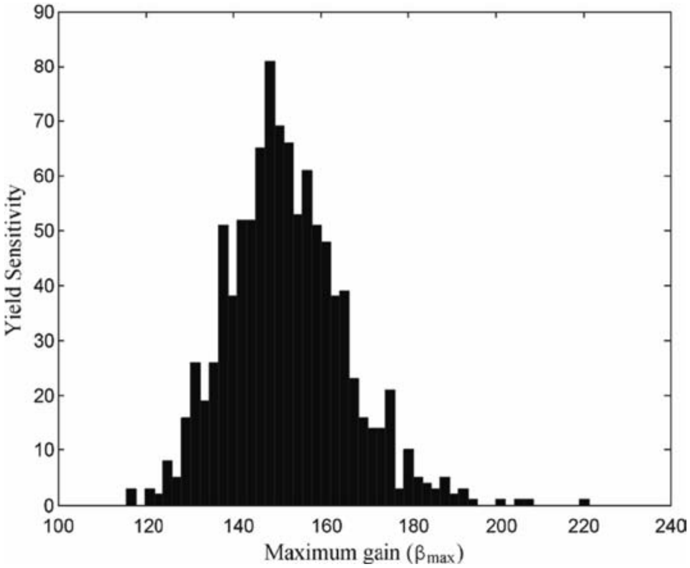


Figure 34. Yield histogram of peak cutoff frequency before design centering (97).

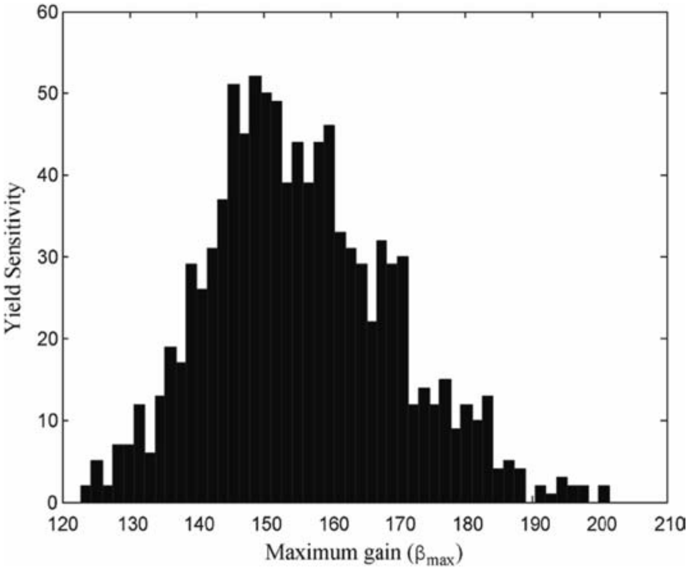


Figure 35. Yield histogram of peak cutoff frequency after design centering (97).

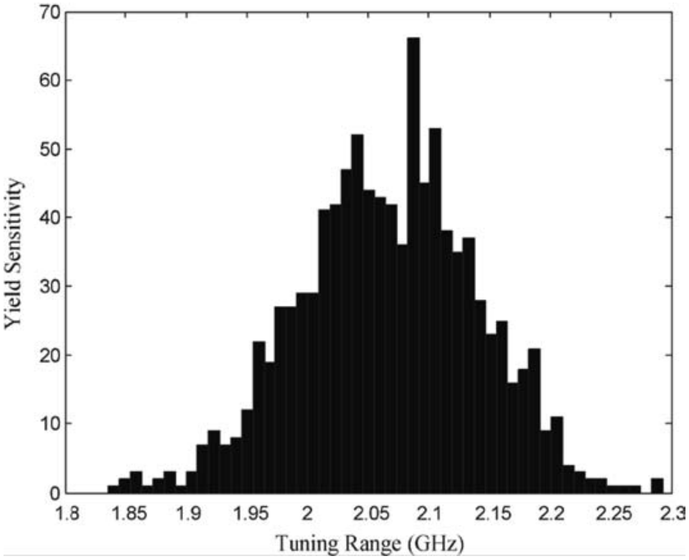


Figure 36. Yield histogram of tuning range before design centering (97).

portion of the devices exhibited a tuning range below 2.15 GHz before design centering. After design centering, parametric yield improved from 8% to 85%.

Although the distribution of input parameters and the electrical performance parameters for those devices (Figs. 34–37) were normal, Pratap et al. demonstrated using VCOs that the neuro-genetic design centering method was also effective when the distributions of process and layout parameters are non-normal (97). To test the effectiveness of the neuro-genetic scheme on nonnormal distributions, design centering was performed using the values in Table 8, and results are summarized in Table 9.

After design centering for the non-normal case, the yield improved from 0.01% to 71% in just 38 iterations. This neuro-genetic approach demonstrates the advantages of artificial intelligence tools in yield maximization modeling.

CONCLUSION

In semiconductor manufacturing, process and equipment reliability directly influence cost, throughput, and yield. Significant process modeling and control efforts are required to reach projected targets for future generations of microelectronics devices and integrated circuits. Computer-assisted methods will provide a strategic advantage in undertaking these tasks, and among such methods, neural networks, genetic algorithms, expert systems, and fuzzy logic have certainly proven to be viable techniques.

Thus far neural networks have impacted semiconductor manufacturing at the process engineering level. In fact, the use of neural networks now is probably at a point in its evolution comparable with that of statistical experimental design or Taguchi methodology a decade or two ago, and now statistical methods such as these have become perva-

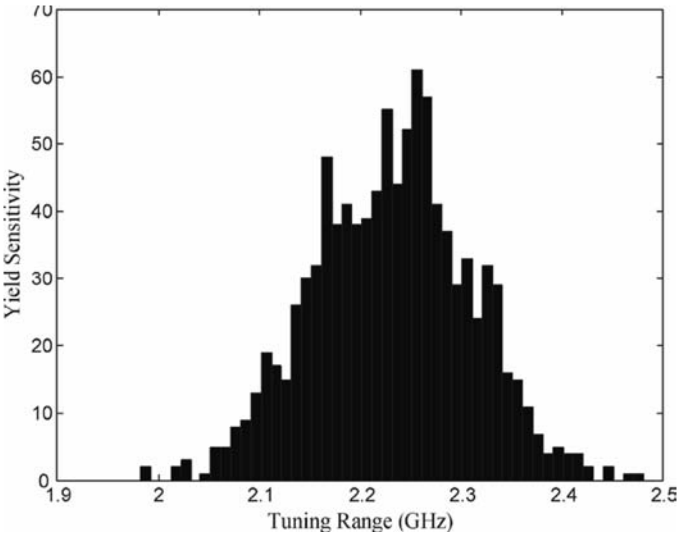


Figure 37. Yield histogram of tuning range after design centering (97).

sive in the industry. The outlook for neural nets is therefore similarly promising. New applications are appearing and software is constantly being developed to meet the needs of these applications. The overall impact of neural network techniques in this field depends primarily on awareness of their capabilities and limitations, coupled with a commitment to their implementation. With each new successful application, neural networks are coupled with other intelligence tools and continue to gain acceptance, and thus their future is bright.

BIBLIOGRAPHY

1. May, G. S. Manufacturing ICs the Neural Way. *IEEE Spectrum* 1994, **31**, pp. 47–51.
2. Losleben, P. Semiconductor Manufacturing in the 21st Century: Capital Investment vs. Technological Innovation, *Proc. 9th IEMT Symposium*; 1990.
3. Huang, S. H.; Zhang, H. C. Artificial Neural Networks in Manufacturing—Concepts, Applications, and Perspectives. *IEEE Trans. Components Packaging Manufacturing Technol. A* 1994, **17**, pp. 212–228.
4. Haykin, S. *Neural Networks: A Comprehensive Foundation*, 2nd ed. Prentice-Hall: Upper Saddle River, 1999.
5. Hopfield, J. J.; Tank, D. W. Neural Computation of Decisions Optimization Problems. *Biologic. Cybernet.* 1985, **52**, pp 141–152.
6. Hsieh, K. L.; Tong, L. I. Optimization of Multiple Quality Responses Involving Qualitative and Quantitative Characteristics in IC Manufacturing Using Neural Networks. *Comput. Industry* 2001, **46**, pp 1–12.
7. Liao, D. Y.; Wang, C. N. Neural-Network-Based Delivery Time Estimates for Prioritized 300-mm Automatic Material Handling Operations. *IEEE Trans. Semiconductor Manufactur.* 2004, **17**, pp 324–332.
8. Su, C. T.; Chiang, T. L. Optimizing the IC Wire Bonding Process Using a Neural Networks/Genetic Algorithms Approach. *J. Intell. Manufactur.* 2003, **14**, pp 229–238.
9. Stokes, D.; May, G. S. Real-time Control of Reactive Ion Etching Using Neural Networks. *IEEE Trans. Semiconductor Manufactur.* 2000, **13**, pp 469–480.
10. Stokes, D.; May, G. S. Indirect Adaptive Control of Reactive Ion Etching Using Neural Networks. *IEEE Trans. Robot. Automation* 2001, **17**, pp 650–657.
11. Wang, K. J.; Chen, J. C.; Lin, Y. S. A Hybrid Knowledge Discovery Model Using Decision Tree and Neural Network for Selecting Dispatching Rules of a Semiconductor Final Testing Factory. *Production Planning Control* 2005, **16**, pp 665–680.
12. Irie, B.; Miyake, S. Capabilities of Three-Layered Perceptrons. *Proc. IEEE Int'l Conf. Neural Networks*; 1988.
13. Rietman, E. A.; Frye, R. C. A Genetic Algorithm for Low Variance Control in Semiconductor Device Manufacturing: Some Early Results. *IEEE Trans. Semiconductor Manufactur.* 1996, **9**, pp 223–229.
14. Holland, J. H., *Adaptation in Natural and Artificial Systems*. University of Michigan Press: Ann Arbor, MI, 1975.
15. Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley: Reading, MA, 1989.
16. May, G. S. Intelligent SOP manufacturing. *IEEE Trans. Advanced Packag.* 2004, **27**, pp 426–437.
17. Setia, R.; May, G. S. Modeling and Optimization of Via Formation in Dielectrics by Laser Ablation Using Neural Networks and Genetic Algorithms. *IEEE Trans. Electron. Packaging Manufactur.* 2004, **27**, pp 133–144.
18. Zadeh, L. A. Fuzzy Sets. *Inform. Control* 1965, **8**, pp 338–353.
19. Geisler, J. P.; Lee, C. S. G.; May, G. S. Neurofuzzy Modeling of Chemical Vapor Deposition Processes. *IEEE Trans. Semiconductor Manufactur.* 2000, **13**, pp 46–60.
20. Chang, P. C.; Liao, T. W. Combining SOM and Fuzzy Rule Base for Flow Time Prediction in Semiconductor Manufacturing Factory. *Appl. Soft Comput.* 2006, **6**, pp 198–206.
21. Chang, P. C.; Hieh, J. C.; Liao, T. W. Evolving Fuzzy Rules for Due-Date Assignment Problem in Semiconductor Manufacturing Factory. *J. Intel. Manufactur.* 2005, **16**, pp 549–557.
22. Fargher, H. E.; Kilgore, M. A.; Kline, P. J.; Smith, R. A. A Planner and Scheduler for Semiconductor Manufacturing. *IEEE Trans. Semiconductor Manufactur.* 1994, **7**, pp 117–126.

23. Yang, T. H.; Tsai, T. N. Modeling and Implementation of a Neurofuzzy System for Surface Mount Assembly Defect Prediction and Control. *IEE Trans.* 2002, **34**, pp 637–646.

24. Yu, C.-Y.; Huang, H.-P. Priority-Based Tool Capacity Allocation in the Foundry Fab. *Proc. Int'l Conf. Robotics and Automation (ICRA)*; 2001.

25. Kinnaird, C.; Khotanzad, A. Wire Bonding Process Control Using Fuzzy Logic. *Proc. IEEE Int'l Symp. Semic. Manufac. Conf.*; 1997.

26. Azzaro, C.; Floquet, P.; Pipouleau, L.; Domenech, S. A Fuzzy Simulation Model for Production Control in a Semiconductor Wafer Manufacturing. *Proc. 3rd IEEE Conf. Cont. App.*; 1994.

27. Cheng, M.-H.; L, H.-S.; Lin, S.-Y.; Liu, C.-H.; Lee, W.-Y.; Tsai, C.-H. Fault Detection and Isolation for Plasma Etching Using Model-based Approach. *Proc. IEEE/SEMI Adv. Semic. Manufac. Conf. Workshop*; 2003.

28. Setia, R.; May, G. S. Run-to-Run Failure Detection and Diagnosis Using Neural Networks and Dempster-Shafer Theory: An Application to Excimer Laser Ablation. *IEEE Trans. Electron. Packaging Manufactur.* 2006, **29**, pp 42–49.

29. Rosen, I. G.; Parent, T.; Cooper, C.; Chen, P.; Madhukar, A. A Neural-Network-Based Approach to Determining a Robust Process Recipe for the Plasma-Enhanced Deposition of Silicon Nitride Thin Films. *IEEE Trans. Control Syst. Technol.* 2001, **9**, pp 271–284.

30. Kim, B.; Hong, W.-S. Use of Neural Network to Characterize a Low Pressure Temperature Effect on Refractive Property of Silicon Nitride Film Deposited by PECVD. *IEEE Trans. Plasma Sci.* 2004, **32**, pp 84–89.

31. Kim, B.; Han, S.-S.; Kim, T. S.; Kim, B. S.; Shim, I. J. Modeling Refraction Characteristics of Silicon Nitride Film Deposited in a SiH/sub 4/-NH/sub 3/-N/sub 2/ Plasma Using Neural Network. *IEEE Trans. Plasma Sci.* 2003, **31**, pp 317–323.

32. Wang, X. A.; Mahajan, R. L. Artificial Neural Network Model-Based Run-to-Run Process Controller. *IEEE Trans. Components, Packaging, Manufactur. Technol. C* 1996, **19**, pp 19–26.

33. Han, S. S.; Cai, L.; May, G. S.; Rohatgi, A. Modeling the Growth of PECVD Silicon Nitride Films for Solar Cell Applications Using Neural Networks. *IEEE Trans. Semiconductor Manufactur.* 1996, **9**, pp 303–311.

34. Bhatikar, S. R.; Mahajan, R. L. Artificial Neural-Network-Based Diagnosis of CVD Barrel Reactor. *IEEE Trans. Semiconductor Manufactur.* 2002, **15**, pp 71–78.

35. Salam, F. M.; Piwek, C.; Erten, G.; Grotjohn, T.; Asmussen, J. Modeling of a Plasma Processing Machine for Semiconductor Wafer Etching Using Energy-Functions-Based Neural Networks. *IEEE Trans. Control Syst. Technol.* 1997, **5**, pp 598–613.

36. Kim, B.; May, G. S. Reactive Ion Etch Modeling Using Neural Networks and Simulated Annealing. *IEEE Trans. Components, Packaging, Manufactur. Technol. C* 1996, **19**, pp 3–8.

37. Kim, B.; Kwon, K. Modeling Magnetically Enhanced RIE of Aluminum Alloy Films Using Neural Networks. *IEEE Trans. Semiconductor Manufactur.* 1998, **11**, pp 692–695.

38. Hong, S. J.; May, G. S.; Park, D.-C. Neural Network Modeling of Reactive Ion Etching Using Optical Emission Spectroscopy Data. *IEEE Trans. Semiconductor Manufactur.* 2003, **16**, pp 598–608.

39. Hong, S. J.; May, G. S. Neural-Network-Based Sensor Fusion of Optical Emission and Mass Spectroscopy Data for Real-Time Fault Detection in Reactive Ion Etching. *IEEE Trans. Industrial Electron.* 2005, **52**, pp 1063–1072.

40. Hettwer, A.; Benesch, N.; Schneider, C.; Pfitzner, L.; Ryssel, H. Phi-scatterometry for Integrated Linewidth and Process Control in DRAM Manufacturing. *IEEE Trans. Semiconductor Manufactur.* 2002, **15**, pp 470–477.

41. Cardarelli, G.; Palumbo, M.; Pelagagge, P. M. Use of Neural Networks in Modeling Relations Between Exposure Energy and Pattern Dimension in Photolithography Process [MOS ICs]. *IEEE Trans. Components, Packaging, Manufactur. Technol. C* 1996, **19**, pp 290–299.

42. Choi, J. Y.; Do, H. M. A Learning Approach of Wafer Temperature Control in a Rapid Thermal Processing System. *IEEE Trans. Semiconductor Manufactur.* 2001, **14**, pp 1–10.

43. Yi, J.; Sheng, Y.; Xu, C. S. Neural Network Based Uniformity Profile Control of Linear Chemical-Mechanical Planarization. *IEEE Trans. Semiconductor Manufactur.* 2003, **16**, pp 609–620.

44. Thongvigitmanee, T.; May, G. S. Modeling Ceramic Filled Polymer Integrated Capacitor Formation Using Neural Networks. *IEEE Trans. Electron. Packaging Manufactur.* 1999, **22**, pp 314–318.

45. Kim, T. S.; May, G. S. Sequential modeling of via formation in photosensitive dielectric materials for MCM-D applications. *IEEE Trans. Semiconductor Manufactur.* 1999, **12**, pp 345–352.

46. Su, C. T.; Chiang, T. L. Optimal Design for a Ball Grid Array Wire Bonding Process Using a Neuro-Genetic Approach. *IEEE Trans. Electron. Packaging Manufactur.* 2002, **25**, pp 13–18.

47. Pratap, R. J.; Staiculescu, D.; Pinel, S.; Laskar, J.; May, G. S. Modeling and Sensitivity Analysis of Circuit Parameters for Flip-Chip Interconnects Using Neural Networks. *IEEE Trans. Advanced Packag.* 2005, **28**, pp 71–78.

48. Yu, C.-Y.; Huang, H.-P. On-line Learning Delivery Decision Support System for Highly Product Mixed Semiconductor Foundry. *IEEE Trans. Semiconductor Manufactur.* 2002, **15**, pp 274–278.

49. Davis, C.; Hong, S.; Setia, R.; Pratap, R.; Brown, T.; Ku, B.; Triplett, G.; May, G. An Object-Oriented Neural Network Simulator for Semiconductor Manufacturing Applications. *Proc. 8th World Multi-Conference on Systemics, Cybernetics and Informatics*, 2004.

50. Muller, J.; Pindo, M.; Ruping, S. Process Improvements by Applying Neural Networks. *Semiconductor International*, 2002.

51. Neumath. http://www.neumath.com/index.htm.

52. Nadel, L.; Cooper, L.; Culicover, P. *Neural Connections, Mental Computation*. MIT Press: Cambridge, MA, 1989.

53. White, D. A.; Boning, D.; Butler, S. W.; Barna, G. G. Spatial Characterization of Wafer State Using Principal Component Analysis of Optical Emission Spectra in Plasma Etch. *IEEE Trans. Semiconductor Manufactur.* 1997, **10**, pp 52–61.

54. Hong, S. J.; May, G. S. Neural Network Modeling of Reactive Ion Etching Using Principal Component Analysis of Optical Emission Spectroscopy Data. *Advanced Semiconductor Manufacturing 2002 IEEE/SEMI Conference and Workshop*, 2002.

55. Brown, T. D.; May, G. S. Hybrid Neural Network Modeling of Anion Exchange at the Interfaces of Mixed Anion III–V Heterostructures Grown by Molecular Beam Epitaxy. *IEEE Trans. Semiconductor Manufactur.* 2005, **18**, pp 614–621.

56. Nami, Z.; Misman, O.; Erbil, A.; May, G. S. Semi-Empirical Neural Network Modeling of Metal-Organic Chemical Vapor

Deposition. *IEEE Trans. Semiconductor Manufactur.* 1997, **10**, pp 288–294.

57. Kuan, Y. D.; Hsueh, Y. W.; Lien, H. C.; Chen, W. P. Integrating Computational Fluid Dynamics and Neural Networks to Predict Temperature Distribution of the Semiconductor Chip with Multi-Heat Sources. *Advances in Neural Networks—ISNN 2006, Pt 3, Proceedings*; 2006, vol. 3973, pp 1005–1013.
58. Majahan, R.; Hopper, P.; Atkins, W. Neural Networks and Fuzzy Logic for Semiconductor Manufacturing, Part II. *Semiconductor Int.* 1995, **8**, pp 111–118.
59. Kim, B.; Bae, J. Prediction of Plasma Processes Using Neural Network and Genetic Algorithm. *Solid-State Electron.* 2005, **49**, pp 1576–1580.
60. Han, S.-S.; May, G. S. Using Neural Network Process Models to Perform PECVD Silicon Dioxide Recipe Synthesis via Genetic Algorithms. *IEEE Trans. Semiconductor Manufactur.* 1997, **10**, pp 279–287.
61. Kim, T. S.; May, G. S. Optimization of Via Formation in Photosensitive Dielectric Layers Using Neural Networks and Genetic Algorithms. *IEEE Trans. Electron. Packaging Manufactur.* 1999, **22**, pp 128–136.
62. Han, S. Modeling and Optimization of Plasma Enhanced Chemical Vapor Deposition Using Neural Networks and Genetic Algorithm.Georgia Institute of Technology: Atlanta, GA, 1996.
63. Yen, J.; Liao, J. C.; Lee, B.; Randolph, D. A Hybrid Approach to Modeling Metabolic Systems Using a Genetic Algorithm and Simplex Method. *IEEE Trans. Systems, Man Cybernet. B* 1998, **28**, pp 173–191.
64. Montgomery, D. C., *Introduction to Statistical Quality Control*. Wiley: New York, 1991.
65. Baker, M. D.; Himmel, C. D.; May, G. S. Time Series Modeling of Reactive Ion Etching Using Neural Networks. *IEEE Trans. Semiconductor Manufactur.* 1995, **8**, pp 62–71.
66. Yazici, H.; Smith, A. E. Neural Network Control Charts for Location and Variance Process Shifts. *Proc. Congr. Neural Net.*, vol. **I**; 1993, pp 265–268.
67. Nelson, D. E.; Ensley, D. D.; Rogers, S. K. Prediction of Chaotic Time Series Using Cascade Correlation: Effects of Number of Inputs and Training Set Size. *Proc. SPIE Conf. Appl. Neural Net.*; 1992.
68. Mori, H.; Ogasawara, T. A Recurrent Neural Network Approach to Short-Term Load Forecasting in Electrical Power Systems. *Proc. 1993 World Congr. Neural Net.*; Seattle, WA, 1993.
69. Rao, S. S.; Pappu, R. S. Nonlinear Time Series Prediction Using Wavelet Networks. *Proc. 1993 World Congr. Neural Net.*, Seattle, WA, 1993.
70. Hattori, S.; Nakajima, M.; Katayama, Y. Fuzzy Control Algorithms and Neural Networks for Flatness Control of a Cold Rolling Process. *Hitachi Rev.* 1992, **41**, pp 31–38.
71. Lam, M. S.; Lin, P.; Bain, L. J. Modeling and Control of the Lithographic Offset Color Printing Process Using Artificial Neural Networks. *Neural Net. Manufac. Robot.*, 1992, **57**, pp 1–10.
72. Mead, W. C.; Brown, S. K.; Jones, R. D.; Bowling, P. S.; Barnes, C. W. Adaptive Optimization and Control Using Neural Networks. *Nuclear Instruments Methods Phys. Res. Section α-Accelerators Spectrometers Detectors Associated Equipment* 1994, **352**, pp 309–315.
73. Nguyen, D.; Lee, M.; Sass, R.; Shoaee, H. Accelerator and Feedback Control Simulation Using Neural Networks. *Proc. Particle Accelerator Conference (PAC)*; 1991.
74. Setia, R.; May, G. S. In-line Failure Detection and Diagnosis of Excimer Laser-Based Microvia Fabrication Using Computational Intelligence. *J. Laser Applications* 2006, **18**, pp 258–266.
75. Box, G. E. P.; Jenkins, G. *Time Series Analysis: Forecasting and Control*, Holden Day: San Francisco, 1976.
76. Smith, T.; Boning, D. A Self-tuning EWMA Controller Utilizing Artificial Neural Network Function Approximation Techniques. *Proc. IEEE/CPMT Int'l 19th Elec. Manufac. Techn. Symp.*; 1996.
77. Butler, S. W.; Stefani, J. A. Supervisory Run-to-Run Control of Polysilicon Gate Etch Using In Situ Ellipsometry. *IEEE Trans. Semiconductor Manufactur.* 1994, **7**, pp 193–201.
78. Rietman, E. A.; Patel, S.; Lory, E. Neural Network Control of a Plasma Gate Etch: Early Steps in Wafer-to-Wafer Process Control. *Proc. IEEE/CPMT Int'l 15th Elec. Manufac. Techn. Symp.*; 1993.
79. May, G. S.; Spanos, C. J., *Fundamentals of Semiconductor Manufacturing and Process Control*, Wiley-Interscience: Hoboken, NJ, 2006.
80. Patel, N. S.; Rajadhyaksha, A.; Boone, J. D. Supervisory Control of LPCVD Silicon Nitride. *IEEE Trans. Semiconductor Manufactur.* 2005, **18**, pp 584–591.
81. Kim, T. S.; May, G. S. Intelligent Control of Via Formation by Photosensitive BCB for MCM-L/D Applications. *IEEE Trans. Semiconductor Manufactur.* 1999, **12**, pp 503–515.
82. Sorsa, T.; Koivo, H. N.; Koivisto, H. Neural Networks in Process Fault Diagnosis. *IEEE Trans. Systems, Man Cybernet.* 1991, **21**, pp 815–825.
83. Burke, L. I.; Rangwala, S. Tool Condition Monitoring in Metal-Cutting—a Neural Network Approach. *J. Intell. Manufactur.* 1991, **2**, pp 269–280.
84. Wasserman, P. D.; Unal, A.; Haddad, S. Neural Networks for On-Line Machine Condition Monitoring. In *Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, 1991.
85. Murphy, J. H.; Kagle, B. J. Neural Network Recognition of Electronic Malfunctions. *J. Intell. Manufactur.* 1992, **3**, pp 205–216.
86. Hillman, D. V. Integrating Neural Nets and Expert Systems. In *AI Expert*, 1990, pp 45–59.
87. May, G. S.; Spanos, C. J. Automated Malfunction Diagnosis of Semiconductor Fabrication Equipment—a Plasma Etch Application. *IEEE Trans. Semiconductor Manufactur.* 1993, **6**, pp 28–40.
88. Kim, B.; May, G. S. Real-time Diagnosis of Semiconductor Manufacturing Equipment Using Neural Networks. *Proc. IEEE/CPMT Int'l Elec. Manufac. Techn. Symp.*, 1995.
89. Hong, S. J.; May, G. S. Neural Network-Based Real-time Malfunction Diagnosis of Reactive Ion Etching Using In Situ Metrology Data. *IEEE Trans. Semiconductor Manufactur.* 2004, **17**, pp 408–421.
90. Sikka, D. Automated Feature Detection and Characterization in Sort Wafer Maps. *Proc. Int. Joint Conf. Neural Net.* 1991–1994. 1993.
91. Plummer, J. Tighter Process Control with Neural Networks. In *AI Expert*, 1993, pp 49-55.
92. Yun, I.; May, G. S. Parametric Manufacturing Yield Modeling of GaAs/AlGaAs Multiple Quantum Well Avalanche Photodiodes. *IEEE Trans. Semiconductor Manufactur.* 1999, **12**, pp 238–251.
93. Page, M. Analysis for Nonnormal Process Distributions. *Semiconductor Int.* 1994, **17**, pp 88–96.

94. Meehan, M. Understanding and Maximising Yield Through Design Centering [Microwave Circuits]. *Proc. IEE Colloquium on Computer Based Tools for Microwave Engineers*; 1991.

95. Antreich, K.; Koblitz, R. Design Centering by Yield Prediction. *IEEE Trans. Circuits Syst.* 1982, **29**, pp 88–96.

96. Director, S.; Hachtel, G. The Simplicial Approximation Approach to Design Centering. *IEEE Trans. Circuits and Syst.* 1977, **24**, pp 363–372.

97. Pratap, R. J.; Sen, P.; Davis, C. E.; Mukhophdhyay, R.; May, G. S.; Laskar, J. Neurogenetic design centering. *IEEE Trans. Semiconductor Manufactur.* 2006, **19**, pp 173–182.

Gary S. May
Tae Seon Kim
Gregory Triplett
Ilgu Yun
Georgia Institute of Technology,
Atlanta, GA
Catholic University of Korea,
Bunchon City, Korea
University of Missouri,
Columbia, Missouri
Yonsei University, Seoul, Korea