

# NEURAL NETWORK OPTIMIZATION ROUTINES FOR PLASMA ETCH PROCESS CONTROL AND EFFICIENT PARTS REPLACEMENT

Debbie L. Sniderman <sup>\*1</sup>, Jill P. Card <sup>2</sup>, Casimir Klimasauskas <sup>3</sup>

<sup>1</sup>Digital Equipment Corp., Hudson, MA 01749 <sup>2</sup>Digital Equipment Corp., Hudson, MA 01749

<sup>3</sup>NeuralWare, Inc. Pittsburgh, PA 15275

Described below is a neural-network based dynamic control strategy for a single wafer dielectric Plasma etcher. In an out-of-control situation, temporal data on replaceable parts and process data collected from a SECSII interface are fed into a policy-iteration optimization routine and a modifiedfeasible sequential quadratic programming algorithm (1). Based on a neural network model of etch rates, selectivity, and non-uniformity, the algorithms generate "least effort" and "lowest cost" methods for returning into control. If a part worn, used, or dirty the routine will suggest that it be replaced/calibrated if it returns the system into control and is cost effective to do so. Similarly, process variable setpoint changes will be suggested if the algorithm finds they are operating in a "costly" regime and bring the process back into spec. The two routines successfully suggested methods for returning into control when tested off-line on several down situations.

## INTRODUCTION

Two neural network based optimization routines are developed for dynamic process control of a plasma etch process using maintenance variables as well as traditional plasma etch variables in the process model. Prior plasma etch models using neural networks (2-9) do not capture tool drift or maintenance events on aging parts, which have an impact on process performance. Our model (10), including the lifetime of parts that either wear out, require cleaning, or fail, is able to predict etch rates, selectivity, and non-uniformity. Given the expense entailed in replacing these parts and the impact on wafer cycle time imposed by shutting a tool down, it is important that these actions are performed efficiently and at minimal cost. This problem of optimal action based on an overall cost function expands naturally into one of dynamic control. While dynamic process control using a neural prediction model has not been applied in the literature to plasma etch processes, the petroleum industry has had published successes in development and application of real-time neural control of chemical processes (11). In these studies, time-dependent input parameters in the form of non-linear equilibrium reaction rates are related to our treatment of time dependent parts replacements and calibration due to tool drift. The objective of this work is to develop a cost effective parts replacement and calibration scheduling procedure based on an optimal control scheme, with the work done in two phases. The first is a feasibility analysis of achieving feedback control of a plasma etch process using a neural prediction model and nonlinear optimization control loop, the results of which are presented in (10) and this paper. The second phase will consist of applying the models and control schemes developed in Phase I to a manufacturing environment as illustrated in Figure 1. This will require real-time automated data input and decision processing, which is discussed in (12).

\* currently at Analog Devices Inc., Cambridge, MA 02139.

## METHODS

### Cost Functions

Cost functions are defined for all input and output variables over their operating limits. Two types of methods of representing cost are developed for each type of variable. For continuously varying and controlled variables (ConV and CV), the costs are piecewise continuous functions based on operating limits and are normalized to range from 0=low to 1=high. For input variables requiring replacement or calibration (RepV and CaIV) cost is represented as a decreasing step function of plasma-on time since last replacement or calibration.

For the ConV's and CV's, the first or second standard deviation of operating range histograms are used to define upper and lower soft regions. Safety limits are determined by process control limits on the system, and target regions are determined by recipe setpoints or "normal" operating values. A normalized cost is assigned in each region as shown for the RF power variable in Figure 2. For the RepV's, three time segments are identified as early, mid, and upper (end of) life, based on histograms of typical part lifetime for each variable, and a normalized cost is assigned to each time period. This is shown for the Clamp variable in Figure 3, where a higher cost is incurred if a clamp is to be changed within an unexpectedly short time since last installation, and a lower cost would result if a clamp's lifetime was extended without causing the process to go out of control. The cost is 0 if no replacement or calibration is to occur.

The method for determining the "cost" of a maintenance event for each RepV and CaIV included evaluating a number of considerations important to production, engineering, and equipment staff, such as: time to complete a calibration or replacement, whether the process chamber would be vented/opened, where the calibration or replacement would be performed, number of people needed to calibrate the item, cost to purchase a replacement part, and number of monitor runs necessary after the calibration was complete were evaluated and ranked for the RepV. Similar factors are considered to determine the relative costs of continuous variables as they move outside the target, soft, and safety operating regions.

We define  $h(z)$  as the cost function vector, and  $h(z(i))$  as the  $i$ th element of the cost function vector for  $i=l\dots,m+p$ . We also make the following definitions for  $z$ , the vector of all variables and output variables for a given etch run:

$$\mathbf{z} \in \mathfrak{R}^n = [ \mathbf{z}^{m_1}, \mathbf{z}^{m_2}, \mathbf{z}^p ] \text{ and } \mathbf{z}^L < \mathbf{z} < \mathbf{z}^U \quad (1)$$

Vectors  $\mathbf{z}^L$  and  $\mathbf{z}^U$  represent lower and upper operating ranges for the variables of  $\mathbf{z}$ , and

$m_1$  = the number of continuous input variables

$m_2$  = the number of maintenance (replacement and calibration) variables

$p$  = the number of output variables

$m = m_1 + m_2$ , the total number of input variables

$\mathbf{z}^{m_1} \in \mathfrak{R}^{m_1}$  = vector of  $m_1$  continuous input variables

$\mathbf{z}^{m_2} \in \mathfrak{R}^{m_2}$  = the vector of  $m_2$  maintenance input variables

$\mathbf{z}^p \in \mathfrak{R}^p$  = the vector of  $p$  continuous output variables.

### Optimization I

The first optimization method, referred to as the "Least Cost" (LC) algorithm, focuses on minimizing the cost of operation over the ranges of all input and output variables, producing a

single solution vector,  $\mathbf{z}_{opt}^m$ . The procedure is relatively straightforward for dealing with continuous input and output variables and seeks to minimize the maximum of the operating costs across all input and output variables while maintaining all within acceptable operating regions. However, introducing maintenance variables requires modification of a standard optimization technique to handle the go/no-go maintenance possibilities for each of these variables.

For  $h(\mathbf{z})$ , the cost function vector for all the input and output variables of a given etch record, we define the scalar

$$\max h(\mathbf{z}) = \max \{h(z(I)): i = 1, 2, \dots, m + p\} \quad [2] \quad [2]$$

as the maximum cost value of the set of continuous input variables, maintenance input variables, and output variables. The optimization problem is to find a set of input variables which minimize [2] and contain maintenance activities as discrete events and continuous variables in continuous event space. Simultaneously, the optimization should address the fact that interactions between the costs for maintenance events,  $h(z^{m2})$  and the costs for the continuous output variables,  $h(z^p)$  are correlated and highly non-linear. These problems are addressed by performing the optimization in two parts: a discrete component and continuous component.

The set of all possible sequences of maintenance events is enumerated, including the null (no action) events set. For computational efficiency, a subset of this set may be extracted, and the maintenance-based cost functions are altered slightly from a decreasing step function over time-since-maintenance, to a close approximation which maintains a small non-zero slope at no more than one point. For each possible combination of maintenance events a continuous optimization is performed using a general-purpose nonlinear optimizer, such as dynamic hill climbing (13) or Feasible Sequential Quadratic Programming (1) to find the value of the input variable vector,  $\mathbf{z}_{opt}^m$ , that minimizes the summed total cost of all input and output variables,

$$\min f(\mathbf{z}) = \sum_{i=1}^{m+p} h(z_{opt}(i)). \quad [3]$$

## Optimization II

The second optimization strategy seeks to minimize the weighted objective function:

$$f(\mathbf{z}^j) = \sum_{i=1}^m f(z_i^j) + K(\prod_{i=m+1}^{m+p} f(z_i^j))^{1/p} \quad [4]$$

for K a constant. The last  $p$  terms of  $\mathbf{z}$  are the output variable values computed from the  $n$  inputs. The term  $(\prod_{i=m+1}^{m+p} f(z_i^j))^{1/p}$  is intended to help remove sensitivity to large valued outliers. In this way, the cost incurred when the majority of the output variables lie close to target is preferred as compared to the cost when all variables are the same distance away from target. Values of  $K \gg 3$  represent weighting of the output variables' adherence to target as more important than adjustments of input variables to lower cost structures, but with no resulting improvement in quality. Weighting the input-output variables in this way is the primary difference between the two optimization routines. This routine heavily favors adjusting the variables that have the greatest individual impact on the achievement of in-spec output vector values. Consequently, corrective action solutions tend to involve fewer input variables, and we can refer to this procedure as the "least effort" algorithm (LE).

To select the variables that have the greatest impact on reaching the target output values, criteria are established on the input/output variable vector,  $\mathbf{z}$ , and a vector set  $\Phi$  is formed such that  $\mathbf{z} \in \Phi$ . The vector set  $\Phi$  resembles a fully-crossed main effects model which most aggressively approaches one or more of the targeted output values without violating the operating limits of the remaining output values and is specified by:  $\Phi = \{\mathbf{z}^j \in \mathfrak{R}^n : j \leq s \in I; \text{ an } s \text{ vector set}\}$ . First,  $s$  vectors are formed and then variables are screened as follows.

The index  $j$  in (4) refers to the  $j$ th vector of a total of  $s$  vectors of dimension  $n = m + p$  which is included in the set to be optimized by  $f$ . These  $s$  discrete vectors are obtained from an original vector set of both continuous and maintenance variables by creating a discrete rate change from nominal partitioning. For the calibration or replacement variables, only two partitions are possible: current life time or a reset time = 0. For the  $m$  continuous variables,  $n_m$ , different rate changes ranging from +80% to -80% of the current value are tested and accepted as long as they are within  $(z^L, z^U)$ . The nominal rate change=0 included. Each continuous variable is individually changed from its nominal setting across all rate partition values while the remaining  $m-1$  input variables are held at nominal value. The  $p$  output variables are computed from the inputs, forming  $\mathbf{z}$ .

The condition for accepting the specific variable at a specified rate change for inclusion in the optimization stage is as follows. Define  $\mathbf{z}_{ik}(l) \in \mathfrak{R}$ , with  $l=1,2,\dots,p$ , referring to the  $l$ th output value obtained when the input variable vector is evaluated at nominal variable values with the exception of the  $i$ th input variable which is evaluated at its  $k$ th rate partition. Also define  $\mathbf{z}_{ik} \in \mathfrak{R}$ , to be the value of the  $i$ th input variable at its  $k$ th rate partition from nominal. The target value for the  $l$ th output variable,  $l=1,2,\dots,p$ , is target  $(l)$ , and the  $l$ th output variable value for the nominal input vector values is denoted  $z_o(l)$ . For each  $i \leq m$ , and each  $k \leq n_m$ ,

$$\text{if } |(z_{ik}(l) - \text{target}(l)) / (z_o(l) - \text{target}(l))| < K(I) \\ \text{for } I \leq p, 0 \leq K(I) \leq 1, \text{ and } \mathbf{z}^L \leq \mathbf{z}_i^k \leq \mathbf{z}^U, \text{ then}$$

$$z_{jk} \in \Delta_i = \{\text{acceptable rate partitioned values of the } i\text{th input variable}\}. \quad [5]$$

To each set  $\Delta_i, i=1,\dots,m$ , we add the  $i$ th nominal value. The final set  $\Phi$  of  $n$ -dimension vectors is composed by crossing all elements of sets  $\Delta_i$  of acceptable input variable rate partitioned values. Thus, the total number of vectors  $\mathbf{z} \in \Phi$ , equal the product of the dimensions of the  $\Delta_i$ , as shown in [6]

$$\text{Total vectors} \in \Phi = \left( \prod_{i=1}^M n_i \right) * (2^{m^2}). \quad [6]$$

## RESULTS AND DISCUSSION

### Method of Analysis

Ideally, repeated real-time prediction and closed-loop testing would be the best way to determine how well the optimizer predictions work in an out-of-control situation. This method of testing would have required that automatic data collection for replaceable part lifetimes be implemented and maintained throughout the lifetime of the project, which was not done to save time. One LE solution and the top 20 LC solutions ranked by the value of the objective function are analyzed off-line by performing the following comparisons. First, LE and LC suggestions are compared to identify which variables were suggested by each optimization routine and to check for subsets and combinations. Second, both suggestions are compared to the actual fix implemented in the fab at the time of failure after assessing how well the actual engineering-suggested solution worked to solve the problem. Next, both routines' suggestions are

compared to known physical properties, trends, and prior tool behaviors to estimate their validity from a directional standpoint only. Magnitudes of suggested changes are not analyzed beyond checking that no suggestions are made outside the input realm of the neural model. Finally, the input "cost" of each variable suggested for change is identified in order to check if the model was simply suggesting a lower cost state for that variable.

The eleven cases chosen for analysis span the following range of input/output scenarios (Table 1). Each event chosen precedes or follows a maintenance event. (i.e. part replacement or calibration) to determine if the optimization routines would suggest that event or an alternative. Cases 4 and 5 (all output CV's in spec after maintenance events) are included to test the sensitivity of each routine.

**Table I: Description of Test Cases Chosen for Statistical Analysis**

Case	1	2	3	4	5	6	7	8	9	10	11
<b>Etch Rate</b>	Low	Low	Low	In Spec	In Spec	In Spec	In Spec	In Spec	High	In Spec hi	In Spec hi
<b>Std Dev</b>	In Spec	In Spec	In Spec hi	In Spec	In Spec hi	In Spec	In Spec	High	In Spec	High	High
<b>Selectivity</b>	In Spec	In Spec	Low	In Spec	In Spec	High	High	In Spec	High	In Spec hi	High

**Summary of Optimizer Results**

Table II summarizes the results of the comparisons described above. Up to 20 LE solutions are compared to the LC result for each case.

**Table II: Summary of Optimizer Results**

Case	Subsets?	Actual?	Physical?	Lower Cost?
<b>Low Etch Rate</b>	Yes	Agreed	ER/Temp	Yes
<b>High Etch Rate</b>	Yes	Agreed	ER/MFC	Yes
<b>High Std Dev</b>	*	*	*	*
<b>Low Selectivity</b>	No	No	Yes	Yes
<b>High Selectivity</b>	Yes	Yes	Yes	Yes
<b>Nothing Wrong</b>	No	Agreed	Yes	Yes

\* Poorest model agreement

In general, the LE routines' variables are subsets of one the LC routine, since the LC suggests smaller adjustments to more variables as they shift within processing regimes. The exceptions being in the known cases where output CV's are in spec. The LE routine produces no suggestions, however, the LC algorithm suggests changing six ConV's by less than 4% purely to reduce the total "cost" of operation and move each variable closer to it's target. For the low selectivity case, the LC routine recognized that the std dev. was in spec but at the extreme upper edge of spec, so it's suggestions seem more targeted towards reducing std dev. as well as raising the low etch rates and low selectivity than the LE did, perhaps accounting for the reason that they were not subsets of each other.

The LC routine seemed to agree with the actual fix implemented in the Fab more often than the LE routine. The LE routine suggested fewer variables and more "risk " or higher cost fixes which put more emphasis on maintaining the outputs in control rather than the inputs. In high selectivity case number 6, the LC solution suggested lowering the throttle valve angle (as seen in Figure 4) which was changed through a maintenance event six points later, and the selectivity did not return into spec until this was done. In another instance, both the LE and LC routines predicted that lowering an electrode temperature would lower the selectivity, and indeed the electrode temperature was calibrated soon after this point after several other fixes had been attempted with no luck in the fab. In another example, the high etch rate (and high selectivity) case's LE solution picked up on a "fix" that eventually lowered selectivity back into spec (about five points later) changing a mass flow controller, which hadn't been done since the tool had been installed, but the model understood the impact of the change and suggested that it be implemented before it was done in the Fab. In all 3 cases, both routines seemed to provide insight beyond the scope of what was attempted on-line and anticipate the actual fix to remedy the problem.

Variable suggestions made by both routines made physical sense and agreed with known DOE experimental results on the system. For example, the low etch rate cases produced suggestions to raise electrode temperatures to return to spec just after a clamp change had occurred. The circumstances for this case were that the process cooling water leaked and the chiller temperatures did not return to their previous values after the leak was fixed. Both optimizer routines recognized the shift in electrode temperatures in the input data and suggested raising the temperature to bring etch rates back into spec, as was done in the fab to remedy the problem. Another example of the optimization routines picking up subtle relationships between the output variables occurred while inspecting the top twenty LE solutions where a trade-off in selectivity with std dev. could be seen.

Since the optimizers' results are highly dependent on the costs associated with a part replacement/calibration, the network may not suggest to replace a part, no matter how worn or out of calibration it is, if the cost to change it at that time is too high. In order to avoid taking processing tools down too frequently to make many small adjustments, the next generation optimizer should have flexible cost structures before they are "frozen" in the optimization routine. Even though much effort went into determining the relative variable, more work could have been done to provide costs that were closer together so some variables would have a better chance at being selected by both routines. Almost all of the LC routines suggested new settings to several variables solely for reducing cost as can be seen in Table III below for the high selectivity case discussed above.

**Table III Top 9 Least Effort Solutions for Case 6 High Selectivity**

Throttle Valve Angle	He Clamp Row	Up Elec Temp	Lo Elec Temp	Endpt Cal	Objectiv Function
-0.20	-0.05	0	0.01	0	4.987
-0.20	0	0	0.01	1	5.070
-0.20	-0.05	-0.01	0	0	5.168
-0.20	-0.2	0	0.01	0	5.193
-0.20	0	-0.01	0	1	5.196
-0.20	0	-0.01	0	0	5.199
-0.20	-0.08	0	0.01	0	5.249
-0.10	-0.05	0	0	0	5.251

For the cases analyzed with poor model agreement, the LC routine "hung" and the LE routine provided unreasonable output std dev's, claiming that a 9% non-uniform etch would become a 0.76% etch, a level which had never been achieved in the tool's history. In future implementations, when the model does not agree well the input will be flagged before an optimization is attempted.

## CONCLUSIONS

The dynamic controller described above has been shown to meet expectations in being able to accurately determine an optimal solution for returning an out of specification condition to within specification for all quality metrics, when tested off-line. Both least cost and least effort solutions are reasonable enough to implement for real-time testing. Additional work will be done to the optimization routine to further test the impact of using different objective functions and gain improved precision without sacrificing speed, as well as implementing in a production environment.

The incorporation of binary input variables representing part replacements and/or calibrations adds a functionality to this procedure not previously encountered in current market software products. These variable forms are applicable to be used on dynamic control structures for fabrication tools within the semiconductor industry.

## ACKNOWLEDGMENTS

The authors would like to extend thanks for ongoing support for this project to: Rich Bickford, Mark Naimo, Paul Skowronski, and Teanne Davis at Digital Equipment Corporation, and Bernie Cerasaro as liaison to Neuralware.

## REFERENCES

1. L. Zhou, and A.L. Tits, *SRC Report TR-90-60rb*. (1991).
2. B. Kim, and G. S. May, *IEEE Trans. On Semiconductor Manufacturing*. 7. No. 1. (1994).
3. T-B Koh, S-Y Cha K. B. Woo, D-S Moon K. H. Kwak. and H. S. Chan-, *IEEE/CPMT Int'l. Electronics Manufacturing Technology Symposium*. 218. (1995).
4. E. A. Rietman, and E. R. Lory, *IEEE Trans. on Semiconductor Manufacturing*. 6. No. 4. (1993).
4. C. D. Himmel. B. Kim, and G. S. May, *Proceedings of the 4th Int. Semi. Manufacturing Science Symposium 124* (1992).
6. M. T. Mocella, J. A Bondur, and T. R. Turner, *SPIE Proceedings*. 1594. (1993).
7. S. F. Lee, and C. J. Spanos, *IEEE Transactions on Semiconductor Manufacturing*. 8. 252. (1995).
8. E. A Rietman, *IEEE Transactions or Semiconductor Manufacturing*. 9 No. 1 (1996).
9. B. Kim, and G. S. May, *IEEE/CPMT Int'l Electronics Manufacturing Technology Symposium* 224. (1995).
10. J. P. Card, D. L. Sniderman, and C. Klimasauskas, "Plasma Etch Control with a Neural Network-based Prediction Mode!," *Proceedings of the Electrochemical Society 191st Spring Meeting*, (1997).
11. S. E. Fahlman, *CMU Technical Report. CMU-CS-88-162*. (1988).
12. J. P. Card, D. L. Sniderman, and C. Klimasauskas, to be published *IEEE Transactions on Neural Networks* Special Edition: Everyday Applications of *Neural Networks*. July. (1997).
13. M., De La Maza. and D. Yuret. *AI Expert*. 9. 13. 26. (1994).

Figure 1: Real-Time Process Control Implementation

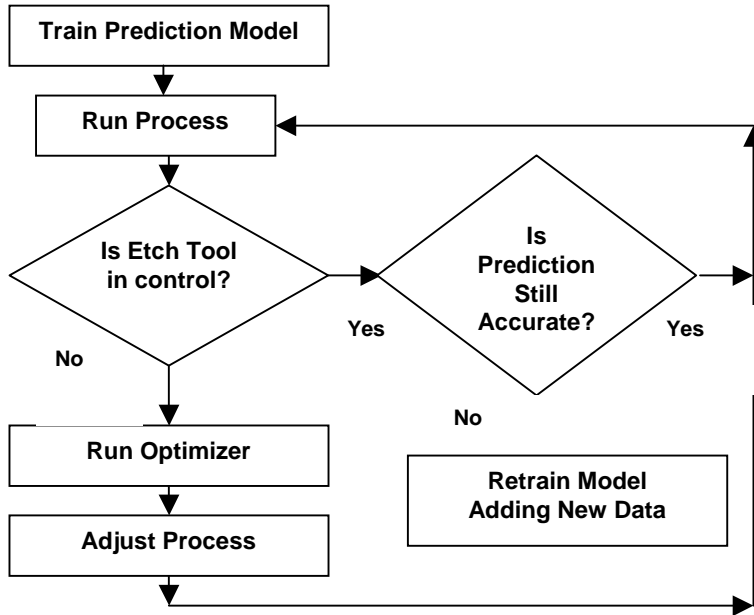


Figure 2: Cost Function for RF Power Continuous Variable

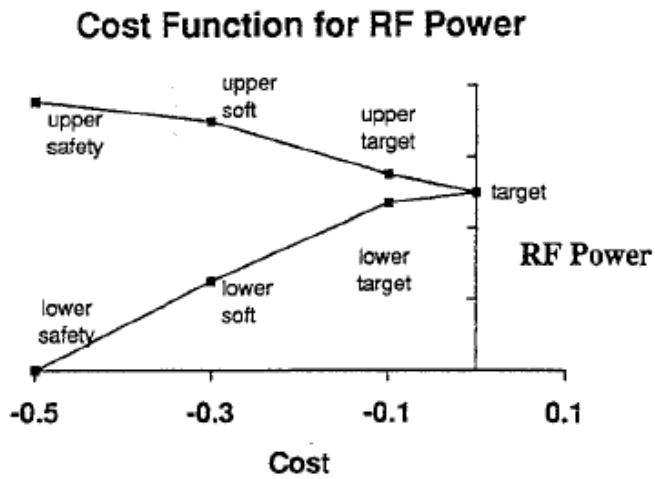


Figure 3: Cost Function for a Replaced/ Calibrated Variable Clamp Replacement

# Cost Function for Clamp Change

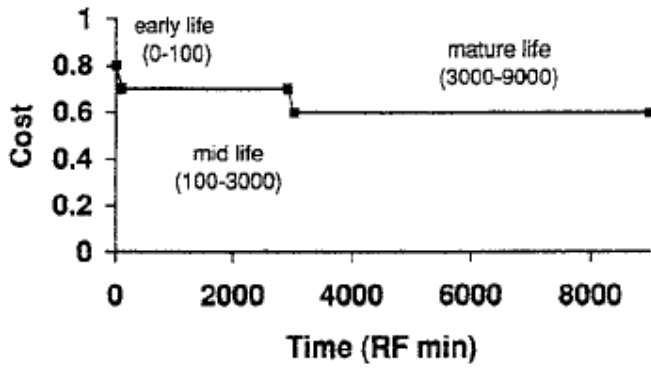


Figure 4: Least Cost Solution for Case 6 High Selectivity

