

Neural Network Based Uniformity Profile Control of Linear Chemical–Mechanical Planarization

Jingang Yi, *Member, IEEE*, Ye Sheng, *Student Member, IEEE*, and C. Shan Xu

Abstract—In this paper, a neural network based uniformity controller is developed for the linear chemical–mechanical planarization (CMP) process. The control law utilizes the metrology measurements of the wafer uniformity profile and tunes the pressures of different air-bearing zones on Lam linear CMP polishers. A feedforward neural network is used to self-learn the CMP process model and a direct inverse control with neural network is utilized to regulate the process to the target. Simulation and experimental results are presented to illustrate the control system performance. Compared with the results by using statistical surface response methods (SRM), the proposed control system can give more accurate uniformity profiles and more flexibility.

Index Terms—Chemical–mechanical planarization (CMP), neural networks, uniformity, run-to-run control.

I. INTRODUCTION

CHEMICAL–MECHANICAL PLANARIZATION (CMP) is an important enabling technology in microelectronics manufacturing. Control of the CMP process is an active research area in both academia and industry. Due to the fact that the CMP process is not completely understood and there is a lack of *in situ* sensors, real-time control of the CMP process is difficult. As a result, run-to-run (R2R) control strategies are used to monitor CMP processes.

The controlled CMP process parameters could be belt speed, head speed, slurry rate, conditioner pressure, etc., and the process responses are material removal rate, wafer nonuniformity, and planarization, etc. In [1], a generic cell CMP controller scheme was discussed, and a regression model was proposed for a R2R controller. The linear process model in [1] is given as

$$y = Ax + c$$

where $A \in \mathbb{R}^{4 \times 2}$ is a model coefficient matrix and $c \in \mathbb{R}^2$ is an offset vector. A recursive adaptation scheme was developed to estimate the vector c with an assumption that model matrix A is fixed during the process. In [2], an exponentially weighted moving average (EWMA) model was used for modeling semiconductor manufacturing processes such as etching. Chen and Guo [3] then presented an aged-based double EWMA scheme for the CMP process to accommodate process drifts

and noises. The controller developed by [3] extended the predictor-corrector control (PCC) scheme in [2] and was applied to CMP processes. In [4], a concept of “sheet film equivalent” (SFE) was used to design a R2R CMP control scheme for dielectric applications, and the experimental results from Texas Instruments fab were compared with the EWMA control scheme.

Statistical surface response methods (SRM) are widely used to model the relationship between process parameters and responses in semiconductor manufacturing. Recently, SRM models have been developed to model the relationship between CMP removal rate and uniformity and process parameters [5]–[9]. Although the SRM models can give a fairly accurate CMP process model, they cannot predict precisely complex CMP processes in real-time under different environmental conditions. In [10], a neural network model was introduced to study the plasma etching processes, and superior performance has been demonstrated compared with the SRM approach. Even though the neural network method has been applied to other semiconductor manufacturing processes such as CVD [11], [12], plasma and ion etchings [10], [13], [14], a small amount research has been performed for CMP processes. Lin and Liu [15] used an adaptive neuro-fuzzy interface system to analyze CMP process parameters on rotary CMP tools; however, very limited experiments and polishing parameters have been studied.

All of the CMP research mentioned above discussed the rotary CMP process. For a typical rotary CMP equipment, the polishing platen is rotating as well as the wafer carrier. The linear CMP process provided by Lam Research Corporation on the other hand has been proven to have better uniformity and edge exclusion performance [16]. The objective of this study is to investigate the use of neural networks to model and control the polishing uniformity of the linear CMP process.

The remainder of this paper is organized as follows. Section II introduces the Lam linear planarization technology (LPT). In Section III we discuss the modeling of CMP process uniformity using a neural network technique. Section IV describes the uniformity profile optimal control using a direct inverse neural network strategy. Simulation and experimental results of the neural network based modeling and control of linear CMP processes are discussed in Section V. Comparisons with the SRM are also presented in this section. Concluding remarks are given in Section VI.

II. LPT CHEMICAL–MECHANICAL PLANARIZATION

CMP processes use the chemical and mechanical interactions among the wafer, polishing pad, and the slurry to planarize the

Manuscript received October 25, 2002; revised June 30, 2003.

J. Yi and C. S. Xu are with the Lam Research Corporation, CMP/Cleaning Technology, Fremont, CA 94538 USA (e-mail: jingang.yi@lamrc.com; shan.xu@lamrc.com).

Y. Sheng is with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: yesheng@me.berkeley.edu).

Digital Object Identifier 10.1109/TSM.2003.818987

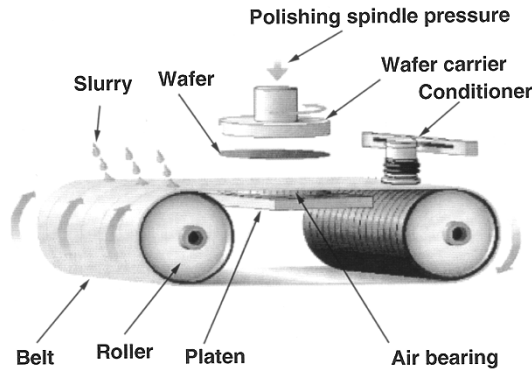


Fig. 1. Schematic of Lam linear chemical-mechanical planarization systems.

wafer surface. For widely used rotary CMP polishers, the polishing pad is on a rotating table with a relatively large radius. The wafer carrier is rotating against the polishing pad table. The polishing pad table is normally rigid, and the wafer head carrier sometimes can be tilted or controlled by the inside air zones.

The LPT polisher provided by Lam Research Corporation uses a different mechanism. Fig. 1 shows the schematic of the Lam LPT setup. The polishing pad is moving linearly while the wafer carrier is rotating against the pad. An air-bearing supports the polishing pad from underneath the air platen [see Fig. 2(a)]. Compared with regular rotary CMP polishers, the LPT design can provide a wide range of polish pad speeds and polish pressures, and therefore increases the throughput as well as the planarization performance.

The air zones on the platen are sets of co-centered small holes located at different radii as shown in Fig. 2(b). From edge to center on the circular platen, the air zones are named from zones A to F. By tuning the air pressures of different air zones on the platen and adjusting the platen height, we can change the polish pad deformation and therefore control the wafer polishing uniformity. The use of the proprietary air-bearing platen intrinsically improves the within-wafer nonuniformity (WIWNU) and decouples uniformity control from removal rate control.

In this paper, we concentrate on how to model the relationship between the air-bearing settings and the polish uniformity profiles. Based on this model, we also investigate the control of the air zone pressures in order to achieve a given uniformity profile under a set of other given polish parameters such as polish pad speed, polish pressure, conditioners, etc. Since air zones B to E and platen height are often used in CMP processes, in this study we only consider tuning the pressures of air zones B to E and the platen height to achieve a certain given uniformity profile.

III. MODELING OF LPT UNIFORMITY WITH NEURAL NETWORKS

Neural networks have been widely used for high-dimensional problems of regression, identification, system modeling and control. Simply speaking, neural networks mimic the way the human brain works: the training algorithms learn patterns of systems through the training process and then predict the system's performance without having to know statistics. The main advantage of neural networks is in their ability to approximate statically or dynamically functional relationships,

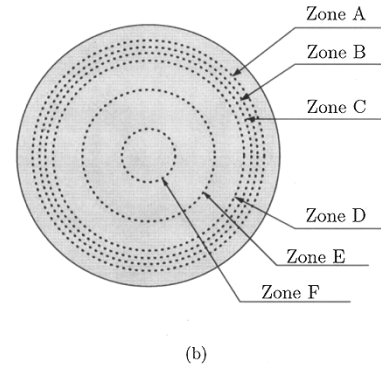
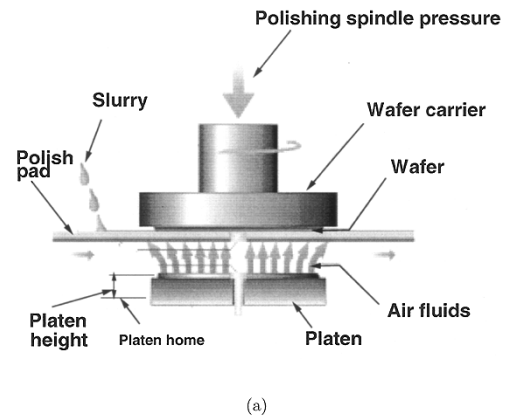


Fig. 2. LPT air-bearing systems. (a) Polishing platen assembly. (b) Platen air zones.

particularly nonlinear relationships. There are however a few challenges associated with neural networks, such as training time, network structures, and training algorithms.

In this section, neural networks are utilized to model the relationship between the uniformity profile and the air zone pressure distributions in the LPT CMP process. Particularly, we will discuss the neural network structures and training algorithms. Fortunately the training convergence rate of neural network models is not a problem for the CMP process since each run takes a relatively long time. We have enough time to train the neural network models between each run. We will discuss training time in Section V-D.

A. Structure of the Neural Networks Based Uniformity Model

Since there is no fully understanding of underlying physics for the CMP polishing process, it is difficult to exactly capture the dynamics of CMP processes. We therefore assume that the relationship between the wafer uniformity profile and the polishing conditions are nonlinear. For LPT CMP processes, these polishing conditions are air-bearing zone pressures and platen height. The CMP processing time is long relative to the aerodynamics of the air-bearing system. Therefore, it is appropriate to consider that a static relationship exists between the input polishing conditions (air zone pressures and platen height) and the process output (removal rate uniformity). A static neural network model is thus chosen and applied to such processes.

The architecture of a neural network model with one hidden layer is shown in Fig. 3. The reason that only one hidden layer neural network is chosen in the model is that this multilayer

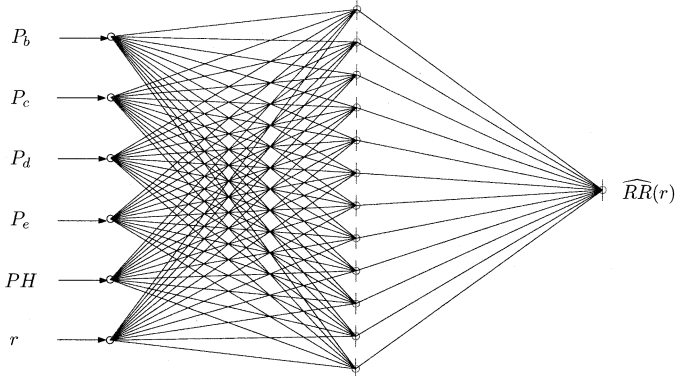


Fig. 3. Architecture of a static neural network model with one hidden layer (bias nodes are not shown in the figure.)

neural network has been proved to be able to approximate any form of nonlinear functions [17], [18]. The number of the neurons of the hidden layer depends on the training process.

A static neural network model for the linear CMP process can be represented as

$$\widehat{RR}(r) = \mathcal{F}(P_b, P_c, P_d, P_e, PH, r) = \mathcal{F}(\mathbf{u}, r) \quad (1)$$

where $\widehat{RR}(r)$ is the estimate of removal rate (RR) at position r , $\mathbf{u} = [P_b \ P_c \ P_d \ P_e \ PH]^T$ is the process tuning variables, P_b, P_c, P_d, P_e are the pressures of platen air-bearing zones B to E respectively, and PH is the platen height. r is the distance from the wafer center. For each neuron, the activation function can be either a linear or hyperbolic tangent (\tanh) or logistic sigmoid function. Empirically, it is often found that “ \tanh ” activation functions give rise to a faster convergence of training algorithms than logistic sigmoid functions. Thus, in this paper, the hidden neurons employ a “ \tanh ” activation function, i.e.,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

We use a linear neuron for the output layer so that it has an unlimited range. The model of (1) can be further represented as

$$\begin{aligned} \widehat{RR}(r) &= \sum_{j=1}^{M_2} W_j \tanh \left(\sum_{i=1}^{M_1} w_{ji} \phi_i + w_{j0} \right) + W_0 \\ &= f(\Phi, \Theta_{ff}) \end{aligned} \quad (2)$$

where $\Phi = (\phi_i) = [P_b \ P_c \ P_d \ P_e \ PH \ r]^T$ and $\Theta_{ff} = (W_j \ w_{ji})$, $j = 0, 1, \dots, M_2, i = 0, 1, \dots, M_1$, is a vector in \mathbb{R}^p , $p = (M_1 + 2)M_2 + 1$, which combines hidden-to-output layer and input-to-hidden layer weights of the feedforward neural network. M_1 and M_2 are the number of inputs to the network and the number of hidden neurons of the network, respectively. For the feedforward neural network process model, $M_1 = 6$, and for the neural network used for the uniformity control in the next section, $M_1 = 5$ since radius r is not a process control variable. Initial weights $\Theta_{ff}(0)$ of the neural networks are selected randomly from a uniform distribution between -0.5 and $+0.5$.

B. Neural Network Training

After determining the neural network structure, we need to select a proper data set $\{P_b^k, P_c^k, P_d^k, P_e^k, PH^k, r_i^k, RR^k(r_i), k = 1, \dots, N, i = 1, \dots, M\}$ to train the model, where we use superscript k to denote the k th run, N is the total number of polishing runs for training, and M is the number of measurement points across the wafer diameter, respectively. In order to obtain good identification results, the input air-bearing pressures and platen height signals P_b, P_c, P_d, P_e, PH must be chosen properly. Since the system is nonlinear, values of training data sets should be varying for different operational conditions. Furthermore, it is important to make the excitation signal cover the operating range. The trained neural network model will be more accurate in the range of the training data [19]. Outside the training range, the neural network model will be less accurate. So it is of great importance to know the working range of the real system. Experimental design for the response surface methods is a good source for the training data set. In this paper, we use the data for the design of experiments (DOE) during the period of tool qualifying and process tuning as the initial neural network training data set.

In order to avoid saturation of activation functions in the neural network, input and output data are scaled as

$$\begin{aligned} P_{is} &= \frac{P_i}{P_{i_{\max}}}, \quad i = b, c, d, e, \quad PH_s = \frac{PH}{PH_{\max}}, \\ RR_s &= \frac{RR}{RR_{\max}}, \quad r_s = \frac{r}{r_{\max}} \end{aligned}$$

where $P_{b_{\max}} = 30$ psi, $P_{c_{\max}} = 45$ psi, $P_{d_{\max}} = 70$ psi, $P_{e_{\max}} = 30$ psi, $PH_{\max} = 40$ mil, $RR_{\max} = 7000$ Å/min, and $r_{\max} = 100$ mm.

Normally, the calculation of weights Θ_{ff}^k of the feedforward model at k th run is obtained by iteratively (run-to-run) minimizing the estimation error function $E_1^k(\Theta_{ff}^k)$ as

$$\begin{aligned} E_1^k(\Theta_{ff}^k) &= \frac{1}{2} \sum_{i=1}^M (RR^k(r_i) - \widehat{RR}^k(r_i))^2 \\ &= \frac{1}{2} \sum_{i=1}^M \left(RR^k(r_i) - f(\Phi^k(r_i), \Theta_{ff}^k) \right)^2 \\ &= \frac{1}{2} \|\epsilon^k\|^2 \end{aligned} \quad (3)$$

where $\epsilon_i^k = RR^k(r_i) - \widehat{RR}^k(r_i)$ is the estimate error evaluated at r_i at k th run and $\epsilon^k \in \mathbb{R}^M$ is a vector with elements ϵ_i^k , $1 \leq k \leq N$, and $1 \leq i \leq M$. Using the first order Taylor expansion of $f(\Phi^k(r_i), \Theta_{ff}^k)$ around the initial values Θ_{ff}^k , we obtain

$$f(\Phi^k(r_i), \Theta_{ff}^k + \Delta\Theta_{ff}^k) \approx f(\Phi^k(r_i), \Theta_{ff}^k) + J\Delta\Theta_{ff}^k \quad (4)$$

for a small $\Delta\Theta_{ff}^k$, where $J \in \mathbb{R}^{M \times p}$ is the Jacobian matrix $(\partial f(\Phi^k(r_i), \Theta_{ff}^k)) / (\partial \Theta_{ff}^k)$. Thus $E_1(\Theta_{ff}^k + \Delta\Theta_{ff}^k)$ can be represented as

$$\begin{aligned} E_1(\Theta_{ff}^k + \Delta\Theta_{ff}^k) &\approx E_1(\Theta_{ff}^k) + \mathcal{G}^T \Delta\Theta_{ff}^k \\ &\quad + \frac{1}{2} \Delta\Theta_{ff}^{kT} J^T J \Delta\Theta_{ff}^k \end{aligned} \quad (5)$$

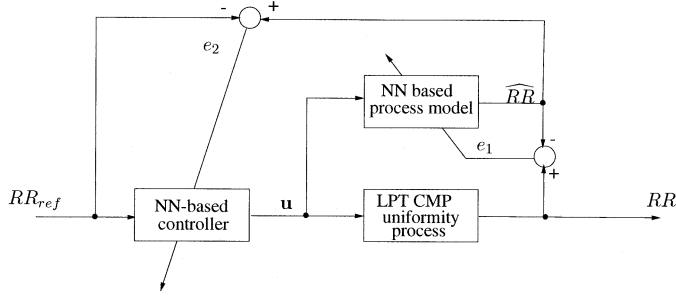


Fig. 4. A schematic of neural network based RR uniformity control systems.

where $\mathcal{G} = \nabla E_1(\Theta_{ff}^k) = J^T \epsilon^k$. Letting $((\partial E_1(\Theta_{ff}^k + \Delta\Theta_{ff}^k))/(\partial\Delta\Theta_{ff}^k)) = 0$, $\Delta\Theta_{ff}^k$ becomes

$$\Delta\Theta_{ff}^k = -(J^T J)^{-1} J^T \epsilon^k. \quad (6)$$

Equation (6) is known as the Gauss–Newton algorithm. One disadvantage with such a method is that the step size given by (6) could be relatively large, in which case the linear approximation in (4) would no longer be valid. In this paper, we can use the Levenberg–Marquardt algorithm [20], a modified error function to solve this problem, which is of the form

$$E_1(\Theta_{ff}^k + \Delta\Theta_{ff}^k) \approx E_1(\Theta_{ff}^k) + \mathcal{G}^T \Delta\Theta_{ff}^k + \frac{1}{2} \Delta\Theta_{ff}^{kT} J^T J \Delta\Theta_{ff}^k + \frac{1}{2} \lambda \|\Delta\Theta_{ff}^k\|^2 \quad (7)$$

where the parameter λ governs the step size. Through this strategy, we can minimize the error function while at the same time keeping the step size small so as to ensure that the linear approximation remains valid [20], [21]. Letting $((\partial E_1(\Theta_{ff}^k + \Delta\Theta_{ff}^k))/(\partial\Delta\Theta_{ff}^k)) = 0$, we obtain the adaptation of neural network weights $\Delta\Theta_{ff}^k$ as

$$\Delta\Theta_{ff}^k = -(J^T J + \lambda I)^{-1} J^T \epsilon^k. \quad (8)$$

The above iteration algorithm can be used in training the feedforward neural network model with a set of measurement data of N runs (total number of $M \times N$ input data). Moreover, it can also be utilized to update the neural network weights once new measurement data set (totally number of M input data) is ready after the weights have been well-trained.

IV. NEURAL NETWORK BASED UNIFORMITY PROFILE CONTROL

In Section III, we discussed the feedforward neural network model of the removal uniformity profile. In this section, a direct inverse neural network based control system is used to regulate the RR uniformity to follow a desired profile $RR_{ref}(r)$.

A. Model Reference Neural Network Based Control

The neural network based model reference control strategy is shown in Fig. 4. The process must be invertible in order to apply this strategy. The input to the control systems is the reference removal rate profile $RR_{ref}(r)$, and the control system goal is to regulate the output of the process uniformity profile $RR(r)$ closest to the input $RR_{ref}(r)$ from run to run. A neural network feedforward process model is first designed by the method discussed in Section III. At the same time, a neural network based

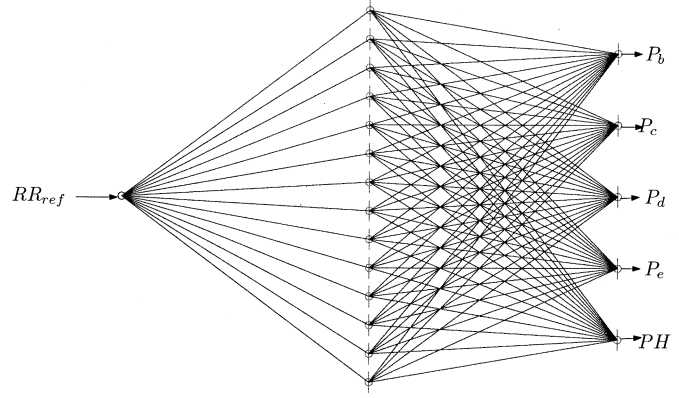


Fig. 5. Architecture of the neural network controller with one hidden layer (bias nodes are not shown in the figure).

controller is also designed and trained in order to tune and optimize the air-bearing pressures and platen height.

The architecture of a neural network based controller with one hidden layer is shown in Fig. 5. Similar to the feedforward neural network for process modeling, the neural network used for the controller has one hidden layer. However, the input and output of the neural network is completely different from the feedforward neural network shown in Fig. 3. The input layer to the neural network of the controller is the removal rate profile reference $RR_{ref}(r)$, and the output layer consists of air-bearing pressures $P_i, i = b, c, d, e$, and platen height PH. The estimate of the uniformity profile $\widehat{RR}(r)$ is then feedback to train and update the neural network weights of the controller.

The training procedure for the neural network weights of the controller can be similarly obtained as the feedforward neural network model. The neural network should be able to tune the plant input u under which the process output $RR(r)$ is driven to the desired value $RR_{ref}(r)$. This can be accomplished by using the error between the desired and actual responses of the process, $[RR_{ref}(r) - RR(r)]$, to adjust the weights of the inverse neural network in the descent direction. If the process output $RR(r)$ can be measured, an estimate $\widehat{RR}(r)$ from the feedforward network model is first updated. Then we use $\widehat{RR}(r)$ to construct an adaptation law for the weights of the inverse neural network, $\Theta_{inv} \in \mathbb{R}^q, q = (M_1 + 2)M_3 + 1$, where $M_1 = 5$ is the number of the nodes in the output layer and M_3 is the number of the neurons in the hidden layer. In a real product process, we are not going to measure the post-polishing removal rate profile for each wafer due to the throughput constraints. Instead, a few wafers from a lot have been measured by either in-line or off-line metrology tools. These data sets can be fed into a feedforward neural network process model and then used to update the neural network based controller.

B. Training Algorithm

In order to design the training algorithms for the run-to-run uniformity controller, without loss of generality, it is assumed that at each run the profile measurements are available¹. First

¹We here consider a “run” in a general sense. It could be a wafer, a lot, or even several wafers, depending on the process design and requirements.

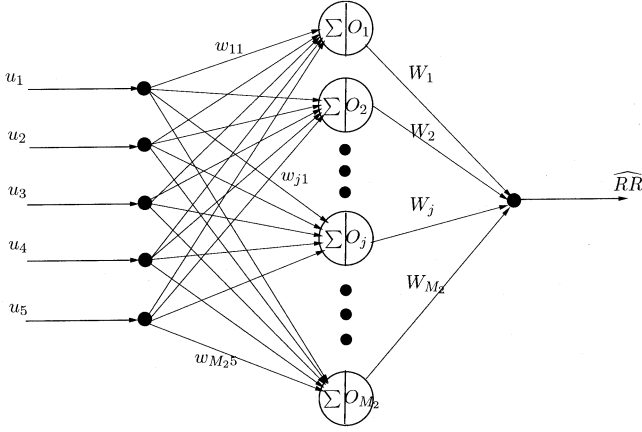


Fig. 6. Partial architecture of training the neural network based controller. A particular radius r is needed as an additional input to the neural network process model to obtain an estimated removal rate profile $\widehat{RR}(r)$.

a rough neural network process model is trained by the Levenberg–Marquardt method off-line. A recursive error backpropagation (BP) method is then employed to train the neural network controller at each training run to drive $\widehat{RR}(r)$ to $RR_{\text{ref}}(r)$. The choice of BP algorithm for the neural network control results from the existence of the feedforward neural network process model, and the fact that Levenberg–Marquardt method is not suitable for the model reference adaptation scheme due to the computational complexity. It will be shown in Section V-D that a recursive BP algorithm is fast enough for the uniformity profile control.

The following error at k th run is minimized

$$E_2^k(\Theta_{\text{inv}}^k) = \frac{1}{2} \sum_{i=1}^M \left[RR_{\text{ref}}^k(r_i) - \widehat{RR}^k(r_i) \right]^2 \quad (9)$$

where $\widehat{RR}^k(r_i)$ is the estimate of $RR^k(r_i)$ by updating the feedforward neural network process model using the measurement at the k th run. Here we assume that the estimated removal rate $\widehat{RR}^k(r_i)$ is approximately equal to the real removal rate $RR^k(r_i)$, and $RR_{\text{ref}}^k(r_i)$ is the reference removal rate profile at k th run². The effectiveness of this assumption is guaranteed by the training accuracy of the proposed neural network model.

Following the same idea of designing feedforward neural network weights, we can derive the updating laws of weights Θ_{inv}^k for the neural network based controller. Notice that the weights of the inverse network can affect the process output $RR^k(r)$ only through the control signal, i.e., \mathbf{u}^k . The error gradient can be computed using the chain rule as

$$\frac{\partial E_2^k}{\partial \Theta_{\text{inv}}^k} = \sum_{i=1}^M \frac{\partial E_2^k}{\partial \widehat{RR}^k} \frac{\partial \widehat{RR}^k}{\partial \mathbf{u}^k} \frac{\partial \mathbf{u}^k}{\partial \Theta_{\text{inv}}^k}. \quad (10)$$

The first partial derivative in (10) can be calculated using (9) as $(\widehat{RR}^k(r_i) - RR_{\text{ref}}^k(r_i))$, and the third is calculated using a standard BP algorithm [20]. To calculate the second term, $(\partial \widehat{RR}^k / \partial \mathbf{u}^k)$, we need to recall the structure of the neural network based process model as shown in Fig. 6. In Fig. 6, we

denote the input layer as $\mathbf{u} = [P_b, P_c, P_d, P_d, \text{PH}]^T$, and the output layer is the estimated uniformity profile \widehat{RR} . A particular radius r_i is needed as an additional input in order to use the neural network process model to obtain the estimated profile $\widehat{RR}(r_i)$. $O_j = \tanh(\cdot)$, $j = 1, 2, \dots, M_2$, are the output of the hidden layer neurons of the neural network process model.

From Fig. 6, we can calculate the second derivative term $(\partial \widehat{RR}^k / \partial \mathbf{u}^k)$ in (10) as

$$\frac{\partial \widehat{RR}^k}{\partial u_i^k} = \sum_{j=1}^{M_2} \frac{\partial \widehat{RR}^k}{\partial O_j^k} \frac{\partial O_j^k}{\partial u_i^k}, \quad (11)$$

where we can obtain

$$\begin{aligned} \frac{\partial \widehat{RR}^k}{\partial O_j^k} &= W_j, \\ \frac{\partial O_j^k}{\partial u_i^k} &= (1 - O_j^{k2}) w_{ji}, \quad i = 1, \dots, 5, \quad j = 1, \dots, M_2. \end{aligned}$$

In the last term, we use the mathematical relationship $\tanh'(x) = 1 - \tanh^2(x)$.

The learning rate of the BP algorithm for the neural network based controller is chosen as $\eta = 10^{-5}$ in this study. In order to speed the parameter convergence, a smoothing filter with a momentum $\alpha = 0.1$ is used to update the neural network parameter $\Delta \Theta_{\text{inv}}^k$ by $\Delta \Theta_{\text{inv}}^k$ at k th run,

$$\Delta \Theta_{\text{inv}}^k = \Delta \Theta_{\text{inv}}^k + \alpha \Delta \Theta_{\text{inv}}^{k-1}. \quad (12)$$

Similar to the recursive Levenberg–Marquardt algorithm (8), the above adaptation laws can be applied to both the multiple run data set training and the weights updating with one new data set.

The overall procedure of the control scheme can be summarized as follows:

- **Step 1:** A feedforward neural network based process model is first trained off-line using N experimental data sets from various DOEs as described by the adaptation law (8) in Section III.
- **Step 2:** A neural network based uniformity controller is trained off-line by a BP algorithm by (12) through the use of the feedforward neural network model obtained in step 1.
- **Step 3:** The trained neural network based uniformity controller is used to tune the air-bearing pressures and platen height at the k th run for the CMP process.
- **Step 4:** At $(k + 1)$ th run, update the feedforward neural network process model by adaptation law (8) with the new data set.
- **Step 5:** After the feedforward process model has been updated, use adaptation law (12) to calculate the new weights for the uniformity controller based on the estimate of removal rate profile \widehat{RR}^{k+1} and then go to step 3.

V. EXPERIMENTAL RESULTS

In this section, we use some simulation and experimental examples to illustrate the design of neural network based process modeling and run-to-run uniformity profile control. First, we

²We treat the reference removal rate profile $RR_{\text{ref}}^k(r_i)$ in a general form assuming that it could be varying for different runs.

TABLE I
EXPERIMENTAL PARAMETERS OF CMP PROCESS #1

Slurry rate	Head pressure	Belt speed	Head speed	Pad conditioning
250ml/min	5 psi	350 ft/min	25 rpm	Linear/50%/6 psi/7 sec. per sweep

TABLE II
REMOVAL RATE ($\text{\AA}/\text{min}$) PROFILES BY DESIGN OF EXPERIMENTS

Run #	Pattern	P_b	P_c	P_d	P_e	PH	$r = 99\text{mm}$	$r = 98\text{mm}$...	$r = 5\text{mm}$	$r = 0\text{mm}$
1	-+--+	8	35	40	22	30	1846	2489	...	3935	3841
2	0	0	25	50	15	20	3273	3439	...	3748	3973
...
31	+--+	22	15	60	8	30	3956	4115	...	3702	3986
32	0	15	25	50	15	20	3314	3581	...	3832	4015

demonstrate the effectiveness and correctness of the neural network modeling. Neural network based uniformity profile tuning is then illustrated by some experimental results. Due to the limited data sets, we will demonstrate the efficiency and effectiveness of the proposed run-to-run profile controller by some numerical simulation results.

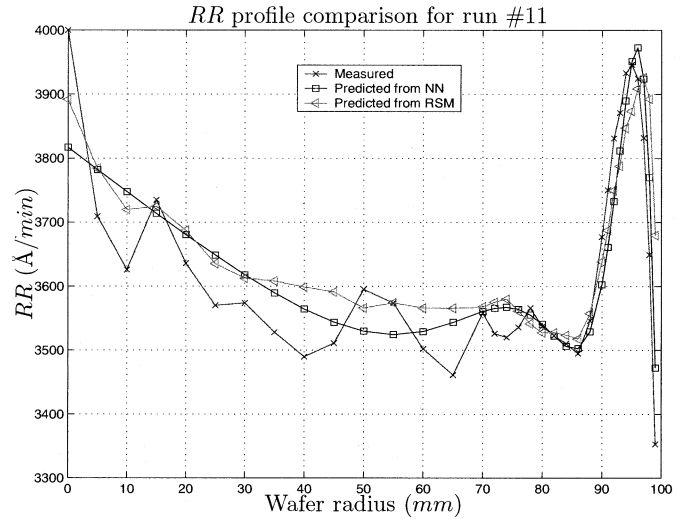
A. Experimental Design

Some experiments have been carried out at the Lam Research Corporation CMP Division using the Lam Teres CMP polisher. Lam's XD Platen was used to regulate the air-bearing pressures. Thermal oxide wafers (200 mm) were polished using Cabot SS12 slurry. Table I shows the polishing parameters except the air-bearing pressures and platen height for the first set of design experiments.

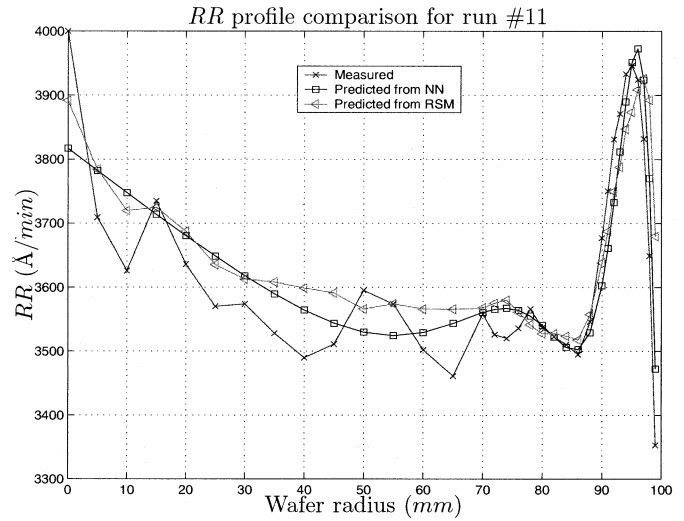
Design of experiments were originally carried out to model the process uniformity profile using the response surface method. Since the polishing parameters listed in Table I have little impact on the process uniformity profiles, we do not change them and focus more on the effect of variations of air-bearing pressures and platen height on uniformity profiles. A set of 32 polishing runs have been designed. Table II shows a few examples of these experiments. In Table II, we illustrate the removal rate profile ($\text{\AA}/\text{min}$) at different radii from 0 to 99 mm. The metrology measurements are 67 points across the wafer diameter: for radii between 0–70 mm, we measure one point at a distance of 5 mm; for radii between 70–90 mm, we measure one point at a distance of 2 mm; for radius between 90–100 mm, we take one point at a distance of 1 mm. Since the polishing head is rotating and the pad is moving linearly, we can obviously establish that the cross-diameter removal rate profile is symmetric with respect to the wafer center. We therefore only list the measurement for radius changes from 0 to 100 mm, and, for radii from -100 to 0 mm, the removal rate profile is symmetric.

B. Performance of the Neural Network Process Model

In order to validate the neural network process model, we have to pick up some experimental data as the training data set. Based on the trained neural network model, we can verify the neural network process model by the other experimental data



(a)



(b)

Fig. 7. Removal rate profiles predicted by a feedforward neural network model and a response surface method for a CMP process with parameters by Table I. (a) Predicted uniformity profile. (b) Predict errors.

sets. We randomly pick up 16 runs of data among the 32 DOE data sets as the training data set. Specifically we pick runs 1, 2, 4, 6, 8, 10, 13, 15, 16, 17, 18, 21, 23, 25, 26, and 28. By trying different neural network structures, we finally used one hidden layer neural network with 12 hidden neurons for the feedforward neural network process model, i.e., $M_2 = 12$. After training the neural network model for the uniformity profile, we also validated the model by predicting the removal rate profile under different combinations of the air-bearing pressures given by other DOE testing data. The Levenberg–Marquardt training algorithm discussed in Section III has been used to train the neural network models with initial parameter $\lambda = 0.9$ using a Matlab Toolbox [22]. Similar real-time training algorithms can be programmed into run-to-run control software.

In order to compare with the neural network process modeling, a response surface method was also used to determine the relationship between the air-bearing pressures and the uniformity profile by using all these data sets of 32 runs. For this ex-

TABLE III
COMPARISON RESULTS BETWEEN THE NEURAL NETWORK MODEL AND
RESPONSE SURFACE METHODS FOR A CMP PROCESS BY TABLE I

Methods	NN	RSM	Experiment
Maximum error ϵ_{\max} ($\text{\AA}/\text{min}$)	183	327	—
Standard deviation σ ($\text{\AA}/\text{min}$)	69	89	—
WIWNU	3.69%	3.52%	4.45%
k -value	6.15 %	5.57 %	8.90%

ample, a simple linear regression model was determined using the optimization method as

$$\text{RR}(r_i) = A(r_i)P_b + B(r_i)P_c + C(r_i)P_d + D(r_i)P_e + E(r_i)\text{PH} + F(r_i), \quad i = 1, \dots, 34 \quad (13)$$

where $\text{RR}(r_i)$ is the estimated removal rate at the radius r_i by RSM, $A(r_i), B(r_i), C(r_i), D(r_i), E(r_i)$ are the coefficient terms for the air-bearing pressures P_b, P_c, P_d, P_e , and platen height PH, respectively, and $F(r_i)$ is the model constant term.

Fig. 7 shows the predicted removal rate profiles under a feed-forward neural work model and a response surface method with the experimental measurements for run #11. The air-bearing setting of run #11 is: $P_b = 15$ psi, $P_c = 25$ psi, $P_d = 50$ psi, $P_e = 15$ psi, and PH = 20 mil. Fig. 7(a) shows the removal rate profiles and Fig. 7(b) illustrates the model predicted errors of the neural network process model and the linear regression model. From Fig. 7, we can see that the neural network model predicts the real measurements very well. Table III gives the comparisons between the neural network model and the response surface model.

In Table III, we calculated the maximum error ϵ_{\max} and standard deviation σ with $\text{RR}_{\text{ref}}(r_i) = 3550$ $\text{\AA}/\text{min}$ as follows

$$\epsilon_{\max} = \max_{1 \leq i \leq 34} (|\widehat{\text{RR}}(r_i) - \text{RR}_{\text{ref}}(r_i)|)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{34} (\widehat{\text{RR}}(r_i) - \text{RR}_{\text{ref}}(r_i))^2}{34 - 1}}.$$

The WIWNU and k -value are calculated as

$$\text{WIWNU} = \frac{\sigma(\widehat{\text{RR}}(r_i))}{\overline{\widehat{\text{RR}}(r_i)}} \times 100\%$$

$$k\text{-value} = \frac{\max_i(\widehat{\text{RR}}(r_i)) - \min_i(\widehat{\text{RR}}(r_i))}{2\overline{\widehat{\text{RR}}(r_i)}} \times 100\%$$

where $\overline{\widehat{\text{RR}}(r_i)}$ is the mean value of the predicted RR profiles. Both WIWNU and k -value are used to qualify the profile nonuniformity. From the above definitions, we can find that for most cases WIWNU and k -value should have the same trend. Normally, k -value is much larger than WIWNU value. k -value uses the difference of maximum and minimum values of the measurements and WIWNU instead uses the standard derivation of the same measurement set. Therefore, if there is one measurement point out of specifications, k -value can give a direct indication while WIWNU might not reflect such a change significantly. In order to tightly control the profile

TABLE IV
EXPERIMENTAL PARAMETERS OF CMP PROCESS #2

Slurry rate	Head pressure	Belt speed	Head speed	Conditioning
200 ml/min	5 psi	225 ft/min	20 rpm	Linear/100%/5 psi/7 sec. per sweep

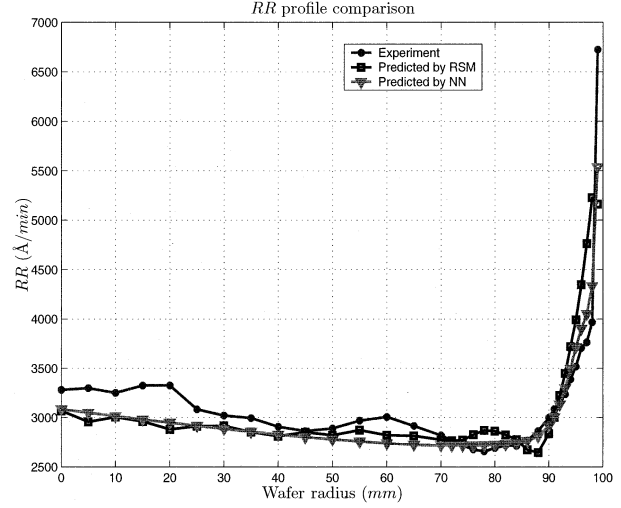


Fig. 8. Removal rate profiles predicted by a trained neural network model and a response surface method for a CMP process with different operating conditions.

TABLE V
COMPARISON RESULTS OF THE NEURAL NETWORK AND RESPONSE SURFACE
METHODS FOR A DIFFERENT CMP PROCESS

Methods	NN	RSM
Maximum error ϵ_{\max} ($\text{\AA}/\text{min}$)	377	1000
Standard deviation σ ($\text{\AA}/\text{min}$)	153	294

nonuniformity across the wafer surface, some semiconductor manufacturers use both WIWNU and k -value to qualify CMP processes.

In order to validate the efficiency of neural network modeling, we applied the proposed neural network model that has been trained by the DOE data set to another experiment with different operating conditions. Table IV lists the operating conditions. The air-bearing settings of the experiment is: $P_b = 15$ psi, $P_c = 50$ psi, $P_d = 5$ psi, $P_e = 10$ psi, and PH = 12 mil. Fig. 8 shows the experimental and the predicted profiles. Table V shows the comparison results of the neural network and response surface methods with different experimental operations.³ From Fig. 8 and Table V, we can see that using the neural networks trained by the DOE experiments under one set of operating conditions (Table I) can predict process profile under another set of operating conditions (Table IV) very well.

Fig. 9 shows the maximum errors and standard deviations of predicted profiles by neural network and response surface methods for another set of 28-run experiments on a different CMP polisher platform and processes. We can clearly find that

³In Table V, 3-mm edge exclusion measurements are used to calculate the maximum errors and mean square errors.

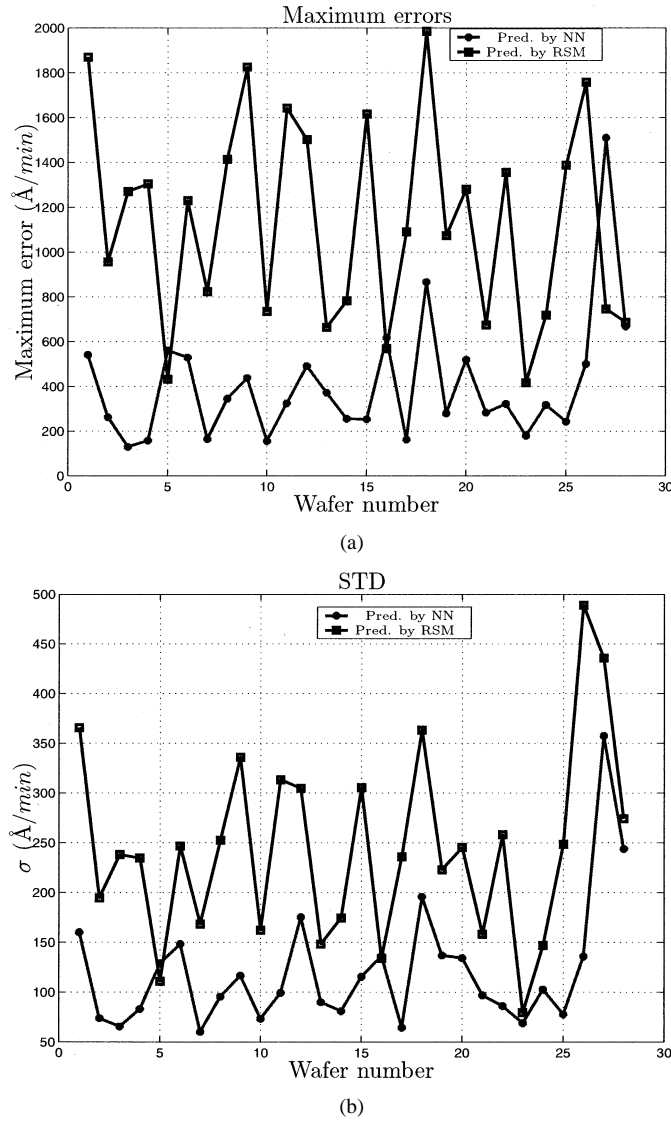


Fig. 9. Comparisons of profile maximum errors and mean square errors over a range of values of air-bearing settings with other process parameters by Table IV. (a) Maximum errors. (b) Standard deviations.

the neural network model shows a superior fitting with experimental results than what the responses surface method does.

From the above examples, we can conclude that the neural network model predicts a better uniformity profile than the response surface method for various operating conditions. We tested but did not demonstrate the comparisons for other DOE runs in detail. We found that in most cases the neural network model can always predict a better CMP removal rate uniformity profile than the response surface method under various CMP process conditions and polisher platforms. Another attractive property of using the neural network model is that it can be updated for each run in real-time and therefore is able to compensate for process parameter variations such as removal rate drift, etc.

C. Performance of the Neural Network Based Controller

Once the feedforward neural network process model is available, we can train the the neural network based controller offline. Then we can use the trained neural network controller to

optimize or tune the uniformity profile recipe for each run. For the neural network used for the uniformity control, we similarly use one hidden layer with 14 hidden neurons, i.e., $M_3 = 14$. The same DOE data set discussed in Section V-A is used here. The training data set for the neural network controller is generated by the feedforward neural network process model with a 16 DOE data set. A more accurate neural network process model trained by all 32 DOE data sets is used for simulation validations.

We compare the optimized process recipes of air-bearing systems given by the neural network based controller and that given by the response surface method. For the linear regression model by response surface method, we can obtain an optimized recipe by minimizing the estimated nonuniformity error with respect to customer requirements $RR_{ref}(r_i)$ as follows:

$$\mathbf{u}_{opt-RSM} = \arg \min_{\mathbf{u}} \{ \sigma [\widehat{RR}_{RSM}(r_i) - RR_{ref}(r_i)] \}$$

$$= [0 \quad 45 \text{ psi} \quad 50 \text{ psi} \quad 14 \text{ psi} \quad 9 \text{ mil}]^T \quad (14)$$

where $\sigma [\widehat{RR}_{RSM}(r_i) - RR_{ref}(r_i)]$, $i = 1, \dots, M$, is the standard derivation of the estimated error of the uniformity profile at measurement points.

For the neural network based controller, we can obtain the optimized recipe by learning law given by (12) as

$$\mathbf{u}_{opt-NN} = [0 \quad 30.5 \text{ psi} \quad 47.2 \text{ psi} \quad 14.1 \text{ psi} \quad 19.2 \text{ mil}]^T. \quad (15)$$

Some experiments using $\mathbf{u}_{opt-RSM}$ were tested. We found that the process performance under recipe $\mathbf{u}_{opt-RSM}$ was not satisfactory. Instead, process engineers have optimized and developed a process recipe \mathbf{u}_{real} , by tuning the process around $\mathbf{u}_{opt-RSM}$, as

$$\mathbf{u}_{real} = [5 \text{ psi} \quad 25 \text{ psi} \quad 50 \text{ psi} \quad 15 \text{ psi} \quad 20 \text{ mil}]^T. \quad (16)$$

In Fig. 10(a), simulation and experimental results of the removal rate profile are illustrated. The experimental results have been obtained under \mathbf{u}_{real} from the real process. For the optimized recipe obtained by RSM, $\mathbf{u}_{opt-RSM}$, we found that the process ("+" line) is far away from the $RR_{ref} = 3550 \text{ Å/min}$. The neural network model predicted profile ("o" line) is closer to the reference profile and fits well with the experimental results ("▽" line). The difference between the profile predicted by the RSM optimized recipe $\mathbf{u}_{opt-RSM}$ and the real recipe \mathbf{u}_{real} again proves that the neural network model is much more accurate than an RSM model in this case. Considering the optimized recipe \mathbf{u}_{opt-NN} given by the neural network based controller, the predicted RR profile ("□" line) is much closer to the real process output. This is not surprising because the optimized recipe \mathbf{u}_{opt-NN} obtained by the neural network based controller (15) is closer to the optimized process recipe \mathbf{u}_{real} (16).

Fig. 10(b) shows the $RR(r)$ error profiles under different recipes. Table VI illustrates the error calculations. From Table VI, we can clearly see that the neural network based controller can predict the best the removal rate profile to the reference RR profile by the smallest maximum error and standard deviation. The real process tuning recipe produces a good RR profile to the reference values, and the RSM recipe generates the worst performance to track the reference profiles. However, if we consider the WIWNU and k -values, there are

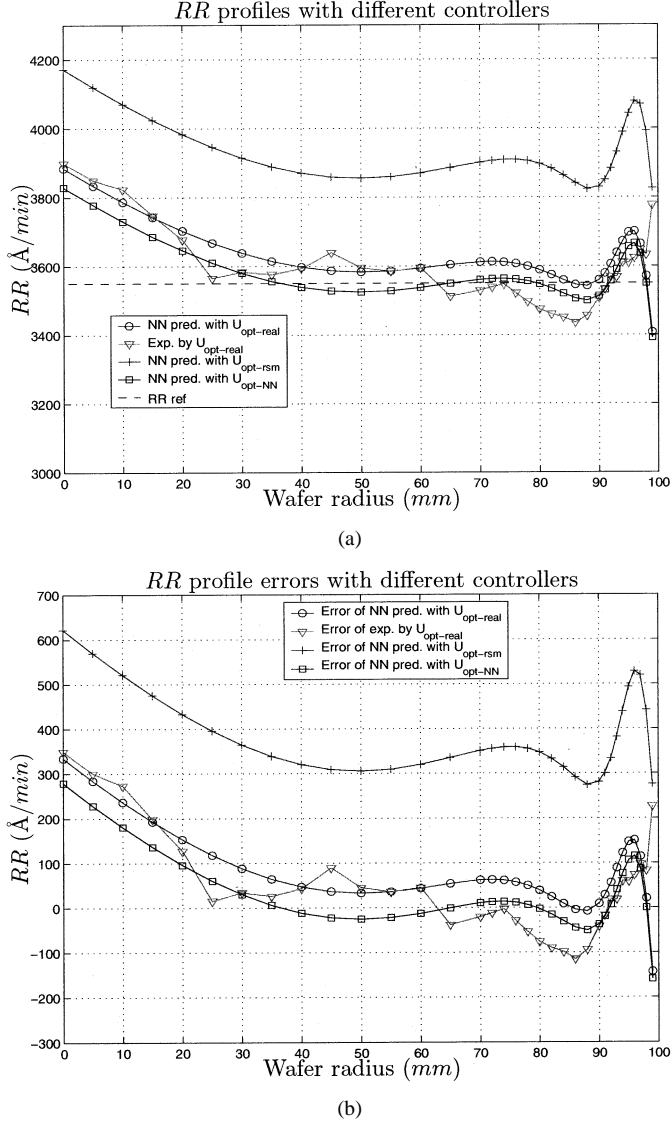


Fig. 10. Comparisons of removal rate profiles under various recipes. (a) RR uniformity profile. (b) Uniformity profile errors.

TABLE VI
COMPARISON RESULTS AMONG DIFFERENT OPTIMIZED RECIPES

Optimized recipes	NN	RSM	Experiment
Maximum error ϵ_{\max} ($\text{\AA}/\text{min}$)	204	622	349
Standard deviation σ ($\text{\AA}/\text{min}$)	86	91	113
WIWNU	2.44 %	2.32%	3.16%
k -value	4.73%	4.42 %	6.45%

almost no significant differences among these three recipes. Even though the uniformity profile given by the RSM is not as good as those obtained by the neural network based model and real process tuning, this method however provides an approximate range of the “optimal” uniformity recipes.

We can use the neural network model to predict the uniformity profile by tuning an individual input parameter channel while keeping other parameters unchanged. This is very useful

in practice as a guidance to tune the process and uniformity profiles. Fig. 11 illustrates the profile variations under varying one tuning parameter around the optimized recipe \mathbf{u}_{real} . From Fig. 11(a), we can find that tuning the air pressure of zone *B* does not have a significant effect on the uniformity profile. Air pressure of zone *C* affects the uniformity profile at all locations across the wafer as shown in Fig. 11(b). Air pressure of zone *D* does not have effect on uniformity profile within the radius of 80 mm but have an effect in the range of 80–100 mm (Fig. 11(c)). Fig. 11(d) shows that varying pressure P_e of zone *E* has minor effects on the removal rate profile across the wafer within radius 90 mm, but has a significant effect around the wafer edge. From Fig. 11(e), we can find that increasing platen height will reduce the removal rate at the locations within radius 85 mm and bring a jump of removal rate around the point 92 mm.

In order to validate the performance of neural network based run-to-run uniformity controller, we simulate a 500 oxide wafer CMP process developed for one of our customers. We assume that the metrology data is available for each 5-wafer run in the process. Material removal rate drifts and random metrology disturbances are added in the model to simulate the realistic processes. For a worst case of our CMP polisher performance in production fabs, 2% run-to-run RR disturbance and a 300–400 $\text{\AA}/\text{min}$ per 1000 wafers RR drop are observed. In the simulation, we add an edge slow drift during each 5-wafer run and 2% metrology white noise at each measurement point. Fig. 12(a) shows the RR profiles of the first (#1) and the last (#500) wafers. The optimized recipe obtained by RSM, $\mathbf{u}_{\text{opt-RSM}}$, is used as the process baseline recipe. The uniformity process model is estimated using a feedforward neural network trained by a 32 DOE data set as shown in Section V-A. Following the run-to-run controller design in Section IV, after k th run, we adjust the feedforward neural network model using the process uniformity profile measurements (with adding disturbances and drifts) and recipe input $\mathbf{u}_{\text{opt-RSM}}$. An updated neural network based control recipe $\mathbf{u}_{\text{opt-NN}}(k+1)$ is used as controlled process input. Fig. 12(b) shows the comparisons of the WIWNU variations with and without neural network based run-to-run control. Because of the profile drift in the process, the WIWNU without run-to-run control increases significantly from 2–3% at wafer #1 to 6–7% at wafer #500. The WIWNU under a run-to-run neural network based control on the other hand maintains at a stable level of 1%–2% throughout the 500-wafer process. Fig. 12(c) shows the material removal rate variations with and without neural network based run-to-run control. Material removal rate under the run-to-run control clearly has been stabilized around the customer requested target (3550 $\text{\AA}/\text{min}$) while a 100–200 $\text{\AA}/\text{min}$ drift exists for the uncontrolled process. Fig. 12(d) illustrates the controlled recipe inputs at each run. We can see that at each run the controller actually tunes the process recipe adaptively based on the estimated process model.

From the above simulation and experimental examples, we found that the linear regression model can provide us with some searching directions for the optimized recipes. The neural network approach can also use the DOE data for the training process. We therefore consider these two methods to

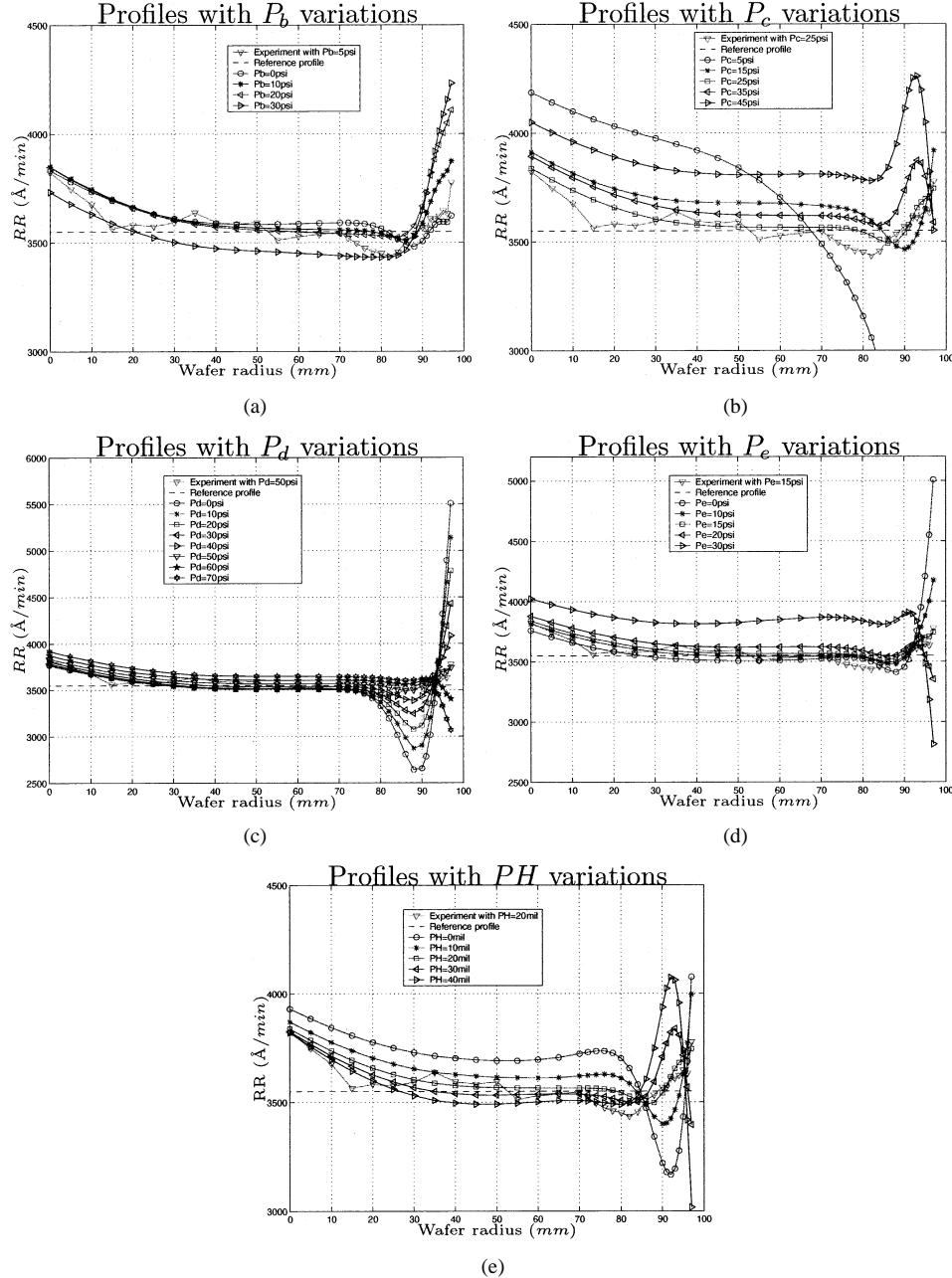


Fig. 11. Comparisons of removal rate profiles by varying one input process parameter around u_{real} . (a) Varying P_b . (b) Varying P_c . (c) Varying P_d . (d) Varying P_e . (e) Varying PH .

be complementary. However, the training algorithm for the neural network process model needs fewer data sets than the RSM approach. Moreover, the neural network approach is more suitable for run-to-run process control since the neural network weights can be tuned on-line from the current run to the next run. Simulation performance demonstrates that the neural network based run-to-run control can potentially tune the process uniformity profiles very well under process drifts and disturbances.

D. Training Time and Convergence of Neural Networks

As we mentioned before, a Levenberg–Marquardt algorithm was used to tune the neural network weights for process modeling, and a standard BP algorithm was used for uniformity pro-

file control. Table VII shows the training time and errors for neural networks used for the feedforward process modeling and uniformity profile control with a 32 DOE data set.

From the results given in Table VII, we can see that the feedforward neural network for process modeling with the Levenberg–Marquardt algorithm converges faster (9 ms per iteration) with small errors compared to the standard BP algorithm, 113 ms per iteration with larger errors. From these calculations, both neural networks can normally be updated on-line during each run once they are well-trained off-line.

VI. CONCLUSION AND FUTURE WORK

In this paper, we discussed a neural network based removal rate uniformity profile model for a linear chemical-mechanical

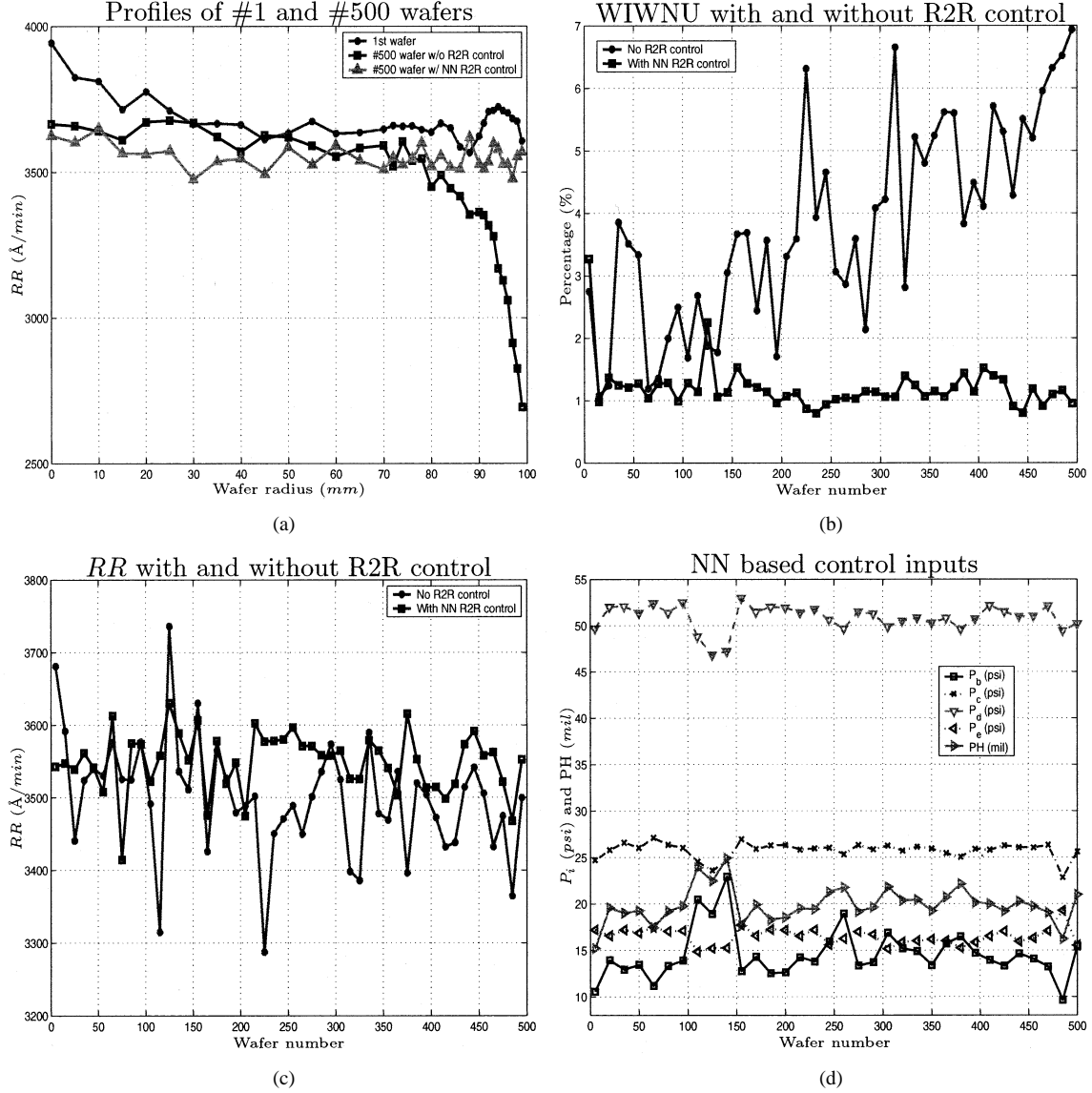


Fig. 12. Run-to-run uniformity control performance. (a) Profiles of the first and the last run wafers. (b) WIWNU with and without control. (c) RR with and without control. (d) Neural network based controller inputs.

TABLE VII
TRAINING TIME AND ERRORS OF TWO TYPES OF NEURAL NETWORKS
TRAINING ALGORITHMS FOR UNIFORMITY PROCESS MODELING AND CONTROL

	Iter #	Training time (sec.)	Errors
L-M algorithm	3200	29	5.9×10^{-5}
BP algorithm	4500	510	1.09×10^{-2}

planarization process. A uniformity run-to-run control scheme was also presented using one neural network model and another neural network controller. The neural network model and controller were trained by a Levenberg–Marquardt algorithm and a BP algorithm, respectively. We first discussed the feedforward neural network based process model and compared the model predictions with the widely used response surface method. Experimental results showed that the trained neural network model can predict the uniformity profile very well even with a smaller DOE data set. The neural network based uniformity

controller was also demonstrated to be able to regulate the process output to the target profile very well. Compared with other model-based run-to-run CMP control schemes, the neural network based controller can compensate for process variations such as removal rate drift, etc.

In this paper, we only consider the uniformity profile control by tuning the air-bearing pressures and platen height with neural network approaches. Fully implementation of such a run-to-run control strategy in production is undergoing. In the future, we can explore the use of the neural network model for removal rate control for the linear CMP process. Since the removal rate and uniformity profile are found roughly decoupled for the linear CMP process, this will simplify the process design and enhance the system performance.

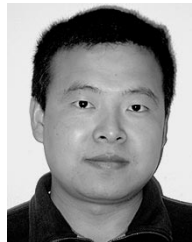
ACKNOWLEDGMENT

One of the authors, J. Yi, would like to thank Dr. D. Wei and Dr. T. Taylor at Lam Research Corporation for various discus-

sions and helpful comments on design of experiments. The authors also thank two anonymous reviewers and the associate editor for their constructive suggestions and comments.

REFERENCES

- [1] D. S. Boning, W. P. Moyne, T. H. Smith, J. Moyne, R. Telfeyan, A. Hurwitz, S. Shellman, and J. Taylor, "Run by run control of chemical-mechanical polishing," *IEEE Trans. Comp., Packag., Manufact. Technol. C*, vol. 19, pp. 307–314, Dec. 1996.
- [2] S. W. Butler and J. A. Stefani, "Supervisory run-to-run control of polysilicon gate etch using *In Situ* ellipsometry," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, pp. 193–201, May 1994.
- [3] A. Chen and R.-S. Guo, "Age-based double EWMA controller and its application to CMP processes," *IEEE Trans. Semiconduct. Manufact.*, vol. 14, pp. 11–19, Feb. 2001.
- [4] N. S. Patel, G. A. Miller, C. Guinn, A. C. Sanchez, and S. T. Jenkins, "Device dependent control of chemical-mechanical polishing of dielectric films," *IEEE Trans. Semiconduct. Manufact.*, vol. 13, pp. 331–343, Aug. 2000.
- [5] R. Telfeyan, J. Moyne, N. Chaudhry, J. Pugmire, S. Shellman, D. Boning, W. Moyne, A. Hurwitz, and J. Taylor, "A multilevel approach to the control of a chemical-mechanical planarization process," *J. Vac. Sci. Technol. A*, vol. 14, no. 3, pp. 1907–1913, 1996.
- [6] S.-Y. Kim and Y.-J. Seo, "Correlation analysis between pattern and non-pattern wafer for characterization of shallow trench isolation-chemical-mechanical polishing (STI-CMP) process," *Microelectron. Eng.*, vol. 60, pp. 357–364, 2002.
- [7] C. E. Chemali, J. Moyne, K. Khan, R. Nadeau, P. Smith, J. Colt, J. Chapple-Sokol, and T. Parikh, "Multizone uniformity control of a chemical-mechanical polishing process utilizing a pre- and post-measurement strategy," *J. Vac. Sci. Technol. A*, vol. 18, no. 4, pp. 1287–1296, 2000.
- [8] E. D. Castillo and J.-Y. Yeh, "An adaptive run-to-run optimizing controller for linear and nonlinear semiconductor processes," *IEEE Trans. Semiconduct. Manufact.*, vol. 11, pp. 285–295, May 1998.
- [9] S.-K. S. Fan, "Quality improvement of chemical-mechanical wafer planarization process in semiconductor manufacturing using a combined generalized linear modeling-nonlinear programming approach," *Int. J. Prod. Res.*, vol. 38, no. 13, pp. 3011–3029, 2000.
- [10] C. D. Himmel and G. S. May, "Advantages of plasma etch modeling using neural networks over statistical techniques," *IEEE Trans. Semiconduct. Manufact.*, vol. 6, pp. 103–111, May 1993.
- [11] X. A. Wang and R. Mahajan, "Artificial neural network model-based run-to-run process controller," *IEEE Trans. Comp., Packag., Manufact. Technol. C*, vol. 19, pp. 19–26, Mar. 1996.
- [12] Z. Nami, O. Misman, A. Erbil, and G. S. May, "Semi-empirical neural network modeling of metal-organic chemical vapor deposition," *IEEE Trans. Semiconduct. Manufact.*, vol. 10, pp. 288–294, May 1997.
- [13] B. Kim and G. S. May, "An optimal neural network process model for plasma etching," *IEEE Trans. Semiconduct. Manufact.*, vol. 7, pp. 12–21, Feb. 1994.
- [14] D. Stokes and G. S. May, "Real-time control of reactive ion etching using neural networks," *IEEE Trans. Semiconduct. Manufact.*, vol. 13, pp. 469–480, Nov. 2000.
- [15] Z.-C. Lin and C.-Y. Liu, "Application of an adaptive neuro-fuzzy inference system for the optimal analysis of chemical-mechanical polishing process parameters," *Int. J. Adv. Manufact. Technol.*, vol. 18, no. 1, pp. 20–28, 2001.
- [16] A. Jensen, P. Renteln, S. Jew, C. Raeder, and P. Cheung, "Empirical-based modeling for control of CMP removal uniformity," *Solid State Technol.*, vol. 44, no. 6, pp. 101–106, 2001.
- [17] K. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, pp. 183–192, 1989.
- [18] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [19] D. Psaltis, A. Sideris, and A. Yamamura, "A multilayered neural network controller," *IEEE Control Syst. Mag.*, vol. 8, no. 2, pp. 17–21, 1988.
- [20] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford Univ. Press, 1995.
- [21] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with Marquardt algorithm," *IEEE Trans. Neural Networks*, vol. 5, pp. 989–993, Nov. 1994.
- [22] M. Nörgaard, "Neural Network Based System Identification Toolbox," Department of Automation, Technical University of Denmark, Tech. Rep. 97-E-851, 1997.



Jingang Yi (S'98–M'02) was born in Hubei Province, China. He received the B.S. degree in electrical engineering from the Zhejiang University, Hangzhou, China in 1993, the M.Eng. degree in precision instruments from Tsinghua University, Beijing, China, in 1996, the M.A. degree in applied mathematics, and Ph.D. degree in mechanical engineering, both from the University of California at Berkeley in 2001 and 2002, respectively.

Since May 2002, he is with Lam Research Corporation as a systems engineer. His research interests include adaptive, nonlinear and intelligent control and its applications to intelligent transportation systems (ITS), mechatronic systems and semiconductor manufacturing systems.

Dr. Yi is a member of ASME.



Ye Sheng (S'99) was born in Jiangsu Province, China. He received the B.Eng. and M.Eng. degrees in precision instruments from Tsinghua University, Beijing, China, in 1993 and 1996, respectively. He is working toward the Ph.D. degree in mechanical engineering at the University of California at Berkeley.

His research interests include dynamic systems modeling and control, adaptive system, neural networks, fuzzy logic, and signal processing.

C. Shan Xu received the B.S. degree from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree from the University of California at Berkeley, both in physical chemistry.

He is a technical director for Lam Research Corporation's CMP/Cleaning Products Group, and has worked on various CMP applications and consumable development.