

TECHNICAL RESEARCH REPORT

Performance Evaluation of Run-to-Run Control Methods in Semiconductor Processes

by Chang Zhang and John S. Baras

CSHCN TR 2001-20
(ISR TR 2001-44)



The Center for Satellite and Hybrid Communication Networks is a NASA-sponsored Commercial Space Center also supported by the Department of Defense (DOD), industry, the State of Maryland, the University of Maryland and the Institute for Systems Research. This document is a technical report in the CSHCN series originating at the University of Maryland.

Web site <http://www.isr.umd.edu/CSHCN/>

Performance Evaluation of Run-to-Run Control Methods in Semiconductor Processes

Chang Zhang*, Hao Deng[†] and John S. Baras[‡]
[§]

Institute for Systems Research and
Department of Electrical and Computer Engineering
University of Maryland, College Park
College Park, MD, 20742

August 31, 2001

Abstract

Run-to-Run (RtR) control plays an important role in semiconductor manufacturing processes. In this paper, RtR control methods are classified and evaluated. The set-valued RtR controllers with ellipsoid approximation are compared with two typical RtR controllers: the Exponentially Weighted Moving Average (EWMA) controller and the Optimizing Adaptive Quality Controller (OAQC) by simulations according to the following criteria: A good RtR controller should be able to compensate for various disturbances, such as small drifts, step disturbances and model errors; moreover, it should be able to deal with bounds, cost requirement and multiple targets that are often encountered in semiconductor processes. Based on our simulation results, suggestions on selection of a proper RtR controller for a semiconductor process are given as conclusions.

*Phone: (301) 405-6578, Fax: (301) 314-9920, Email: zchang@isr.umd.edu

[†]Phone: (301) 228-6672, Fax: (301) 695-8659, Email: hdeng@bechtel.com

[‡]Phone: (301) 405-6606, Fax: (301) 314-9218, Email: baras@isr.umd.edu

[§]This work was supported by the Center for Satellite and Hybrid Communication Networks, under NASA cooperative agreement NCC3-528

1 Introduction

Run-to-Run (RtR) control plays an important role in semiconductor manufacturing processes [1]-[5], [9]-[11], [13]-[18]. A RtR controller is a model-based process control system that combines the advantage of both the statistical process control and the feedback control. The goal of the controller is to reduce the variability of the process outputs, as measured by the Mean Square Deviations (MSDs) between the process outputs and the target values [1]. A good RtR controller should be able to compensate for various disturbances such as small drifts, shifts (step disturbances) and model errors. A drift disturbance, which may be produced by the equipment aging, change of environment or other factors, causes slow and smooth changes of process outputs. Different from a drift disturbance, a shift disturbance causes a large change of process outputs in a few runs. The shift disturbance may be produced by the failure of a component, change of the operator, etc. A model error is often caused by the coarse estimates of process parameters. A good RtR controller should also be able to deal with bounds, cost requirement and multiple targets that are often encountered in real processes.

Generally, a RtR controller is designed in the following way. First, it computes an optimal control based on the initial process model. The initial process model is usually derived from former off-line experiments such as using the Response Surface Model (RSM) method [1]. A typical block diagram of a RtR controller is illustrated in Figure 1. The RtR controller does not modify its recipe during a run because of the following reasons:

1. Cost. It is usually very expensive to obtain real-time information in a semiconductor process. Discrete-time measurements are much cheaper.
2. Variability. Frequent changes of inputs to the process may increase the variability of the process outputs [14].

When the controller is online, the process model within the controller is updated by the model estimator according to the new measurements from run to run. The optimizer then supplies a new recipe according to the updated process model. The cost function used by the optimizer is usually a weighted quadratic function between the process outputs and target values. An important output should be given a large weight. Different model updating methods lead to different kinds of controllers. In this paper, RtR control methods are classified and evaluated in some typical semiconductor processes. Suggestions on how to select a proper RtR controller for a specific semiconductor process are provided.

This paper is organized as follows. Classification of RtR control methods is given in section 2; comparisons of the set-valued RtR controllers with the Exponentially Weighted Moving Average (EWMA) controller are given in section 3.1; in section 3.2, one of the set-valued RtR controllers is compared with the Optimizing Adaptive Quality Controller (OAQC). Finally, conclusions are given in section 4.

2 Classification of RtR Control Methods

Depending on how to update a process model, RtR control methods can be classified into the following categories.

1. The EWMA method [14]. The EWMA approach is widely used in RtR control for its simplicity and efficiency to compensate for smooth drifts and other small disturbances. The EWMA method uses a linear (affine) model to approximate a process:

$$\hat{y}_t = Au_t + \hat{b}_t, \quad (1)$$

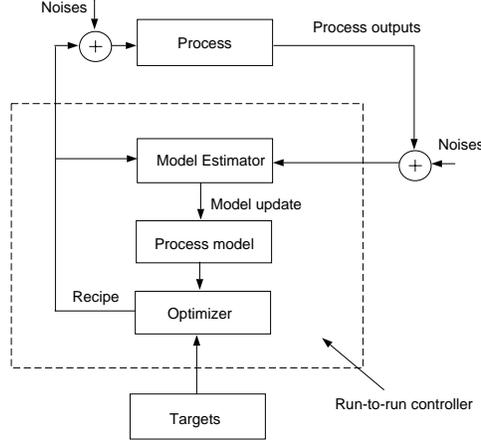


Figure 1: Structure of a RtR controller.

where t is the time index (run number), $\hat{\mathbf{y}}_t$ represents a p by 1 vector of model outputs, A stands for a p by q fixed “gain” matrix, \mathbf{u}_t represents a q by 1 vector of inputs and $\hat{\mathbf{b}}_t$ is a p by 1 vector of model offsets. Let the measured output vector be \mathbf{y}_t . The EWMA method only updates the offset vector in the model:

$$\hat{\mathbf{b}}_{t+1} = (I - W)\hat{\mathbf{b}}_t + W(\mathbf{y}_t - A\mathbf{u}_t), \quad (2)$$

where I is a unit matrix, weight matrix $W = \text{diag}([w_1 \dots w_p])$ and $0 < w_i < 1$, $i = 1, \dots, p$. For more description about the EWMA method, we recommend readers to [2], [9] and [14]. There are many extensions to the EWMA method. For example, the Double Exponential Forecasting Filter method [3] had two EWMA modules. One module was used to update the offset term of the linear model and the other module was used to predict the drift disturbance. The weight matrix W of the EWMA controller is an important factor that affects its performance. An Artificial Neural Network (ANN) was used in [15] to adaptively adjust the weights of the EWMA controller. The neural network had to be trained extensively off-line before it was deployed online.

2. Least Square Estimation (LSE) method. The process model is updated according to a LSE approach. Typical examples are the OAQC [4] and the Kalman filter based approach [10]. The OAQC uses a second-order model to approximate a process:

$$\hat{\mathbf{y}}_t = \hat{N}_t \mathbf{z}_t + \hat{M}_t \mathbf{T}_t + \hat{\mathbf{b}}_t, \quad (3)$$

where \hat{N}_t is a p by $2q + q(q-1)/2$ parameter matrix, $\mathbf{z}_t^T = (\mathbf{u}_t, \mathbf{u}_t^2, u_{i,t}u_{j,t} (i < j))$ is a $2q + q(q-1)/2$ by 1 vector that contains the quadratic expansion of \mathbf{u}_t ($u_{i,t}, u_{j,t}$ are components of \mathbf{u}_t), \hat{M}_t is a p by l parameter matrix, \mathbf{T}_t is a l by 1 vector of time index t , and $\hat{\mathbf{b}}_t$ is the $p \times 1$ offset vector [4]. All the parameters (\hat{N}_t , \hat{M}_t and $\hat{\mathbf{b}}_t$) of the model in the OAQC can be adjusted adaptively using a LSE approach from run to run. For more details about the OAQC, we recommend readers to [4]. Simulations in [13] showed that the OAQC had better performance than the EWMA controller did in controlling a nonlinear process. The Kalman filter based approach uses a linear model to describe a process. Different from an EWMA controller, the Kalman filter based RtR controller can adaptively adjust both the slope and the intercept terms [10]. Therefore, the LSE method based RtR controllers may have stronger tracking ability than the EWMA controller.

3. The set-valued method [1], [5], [17], [18]. Due to measurement errors and environment noises, it is difficult to find an exact process model. The locations of the likely process model parameters for the next optimization run form a set. We could be quite certain that the parameter vector is somewhere in this set. The set-valued approach seeks safe estimates of the process model parameters in the possible parameter set for the next run. The identified process model is insensitive to various noises [1], [17]. The main difficulty of

designing a set-valued RtR controller is the excessive computational time to calculate the feasible parameter set. It is also very hard to solve the optimization problem within an irregular parameter set. An outer-bounding ellipsoid is usually used to approximate the set of likely parameter values; the ellipsoid is used for its simplicity. An ellipsoid is simply characterized by a vector center and a matrix that describes its size; for a convex region, an ellipsoid can be used to obtain a satisfactory approximation; a linear transformation maps an ellipsoid into another ellipsoid. With the ellipsoid approximation, a high-order polynomial model can be used to describe a process:

$$\hat{\mathbf{y}}_t = \Theta_t X_t, \quad (4)$$

where Θ_t is a p by n parameter matrix, X_t is a n by 1 vector of inputs (e.g., $X_t^T = (1, \mathbf{u}_t, \mathbf{u}_t^2, u_{i,t}u_{j,t} (i < j), \mathbf{u}_t^3, \dots)$). Therefore, the ellipsoid algorithm based RtR controllers may describe a nonlinear process more accurately than linear model and second-order model based RtR controllers.

At each iteration, the ellipsoid algorithm returns an outer bounding ellipsoid that contains the true parameter vector with high probability. If the vector center of the ellipsoid is taken as the estimate of the process parameter vector, the explicit model update is implemented and it leads to a model-reference method. If we search for the worst expected output that may be produced by a vector within the ellipsoid and then minimize the worst-case cost, a worst-case controller is obtained. There are mainly two ellipsoid based algorithms available for RtR control: the Modified Optimal Volume Ellipsoid (MOVE) algorithm [17], [18] and the Dasgupta Huang Optimal Bounding Ellipsoid (DHOBE) algorithm [6]. For details of these two algorithm based RtR controllers, we recommend readers to [5], [17] and [18]. The RtR controller based on the MOVE algorithm is called the SVR-MOVE controller [17]. The RtR controller based on the DHOBE algorithm and the model-reference approach is called the DHOBE-MR controller [5]. The RtR controller based on the DHOBE algorithm and the worst-case approach is called the DHOBE-SV controller [5]. Both the SVR-MOVE controller and the DHOBE-MR controller use the center of the ellipsoid as the estimate of the process parameter vector. The DHOBE-SV controller uses the vector within the ellipsoid that produces the worst-expected cost as the estimate of the parameter vector.

4. Other RtR control methods using nonlinear models to describe processes. They include the machine learning method and the neural-network based method, etc. A typical example of the machine learning based approach is the Knowledge Based Interactive Controller (KIRC) [13]. The KIRC uses leaves in a classification decision tree to suggest control actions. The algorithm generates a decision tree by using an information space with attribute tests. The starting operating point is chosen from the largest leaf in the decision tree, where all outputs are inside the target range. A comparative simulation [13] showed that the KIRC was only applicable to processes that could be approximated by linear models. The ANNs have great potential in modeling severe nonlinear semiconductor processes [7], [8], [12]. But a drawback of the ANN method is that it may not supply an explicit model for the process. Thus it may be difficult for one to apply optimal control to the process. A Taylor expansion was used in [16] to find a linear equation to describe the ANN model. The authors believe that a higher-order Taylor extension may produce better results.

3 Performance Evaluation of RtR Controllers by Simulation

In this section, we compared the set-valued RtR controllers with two other popular RtR controllers: the EWMA controller and the OAQC. Because the detailed design of the OAQC was not available to us, we simulated one of the set-valued RtR controllers, the DHOBE-MR controller under the same environment as that of the OAQC. Because the objective of a RtR controller is to maintain the process outputs on targets, the main performance metric is $RMSD(y_i - T_i)$, the square root mean square deviation of the process's i th output y_i from its target value T_i . The smaller its value, the better. In the rest of the paper, we will use the notation $RMSD_i$ for simplicity.

3.1 Comparison of the Set-valued RtR Controllers with the EWMA Controller

The set-valued RtR controllers that we evaluated included the SVR-MOVE controller, the DHOBE-MR controller and the DHOBE-SV controller. They were compared with the EWMA controller. The platform we used was the Chemical Mechanical Polishing (CMP) process. CMP process is of critical importance in semiconductor manufacturing [2]. In the CMP process, a wafer is affixed to a wafer carrier and pressed facedown on a rotating platen holding a polishing pad. A slurry with abrasive material is dripped onto the rotating platen during polish. The typical process goal is to achieve “global” planarization [2].

The underlying process model was given by:

$$y_t = -1382.6 + 50.18u_{1,t} - 6.65u_{2,t} + 163.4u_{3,t} + 8.45u_{4,t} + w_t + \delta t, \quad (5)$$

where t was the time index (run number), y_t was the process output and $u_{i,t}, i = 1, 2, 3$ and 4 were the inputs, w_t was a normally distributed random variable with variance 665.64 and $\delta = -17$ was the drift size at each run. The units were dropped for simplicity. The target value was set as 1700. The inputs were constrained in the range: $0 \leq u_{1,t} \leq 2$, $0 \leq u_{2,t} \leq 200$, $0 \leq u_{3,t} \leq 30$ and $0 \leq u_{4,t} \leq 50$ respectively. The controllers’ objective was to maintain the output y_t as close to the target value as possible.

In the following three scenarios, we tested the performance of RtR controllers with respect to different noises.

Scenario 1

First, we assumed that the controllers had perfect knowledge of the process model parameters at the beginning. The initial process model used by the controllers was:

$$\hat{y}_t = -1382.6 + 50.18u_{1,t} - 6.65u_{2,t} + 163.4u_{3,t} + 8.45u_{4,t}, \quad (6)$$

where \hat{y}_t is the predicted process output. The noises (w_t and δ) were unknown to the controllers. The controllers were fully tuned to compensate for the disturbances based on post-measurements. Because the weight of the EWMA controller played an important role in its performance, we listed the simulation results for different weights of the EWMA controller in the first row of Table 1. The *RMSD* values for the SVR-MOVE controller, DHOBE-MR controller, DHOBE-SV controller and the uncontrolled process were given in the first row of Table 2. The simulation result when the optimal weight value (0.5) of the EWMA controller was chosen is shown in Figure 2. The three horizontal straight dashed lines give the target and the 3σ bounds, where $\sigma = 25.8$ is the standard deviation of the Gaussian noise. The uncontrolled process, denoted by the “*.” symbol, diverged due to the drift disturbance. From Figure 2 and data in Table 2, one can see that the SVR-MOVE controller, DHOBE-MR controller and EWMA controller with a proper weight worked well in this case. The DHOBE-SV controller had larger variability than the other controllers. It was due to the fact that the controller is pessimistic or conservative to produce the recipe.

Scenario 2

In this scenario, a shift disturbance was added to the underlying process (i.e., the disturbances included unknown drift, unknown shift and unknown Gaussian noise). The occurrence of the shift disturbance and its magnitude were unknown to the controllers *a priori*. The initial process model used by the controllers was the same as that in scenario 1. As in scenario 1, we computed the *RMSDs* for different weights of the EWMA controller and listed the results in the second row of Table 1. The *RMSD* values for the SVR-MOVE controller, DHOBE-MR controller, DHOBE-SV controller and the uncontrolled process were given in the second row of Table 2. Figure 3 shows the simulation results when the weight of the EWMA controller was optimal (0.6 in this scenario). Again, the uncontrolled process diverged. The SVR-MOVE controller and DHOBE-MR controller returned the output of the process back to target quickly. The EWMA controller needed more steps to do so. The process controlled by the DHOBE-SV controller still had large variability.

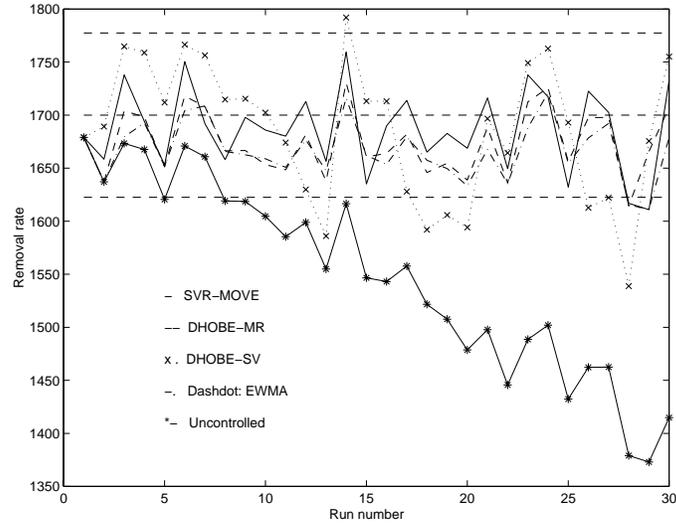


Figure 2: Comparison of the set-valued RtR controllers with the EWMA controller in scenario 1.

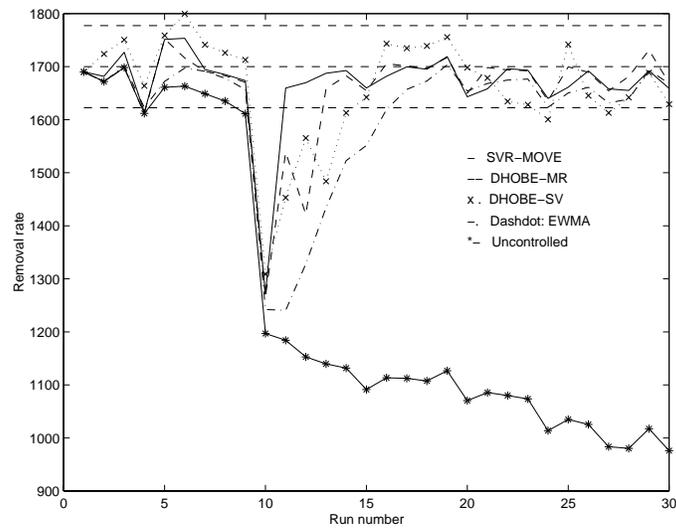


Figure 3: Comparison of the set-valued RtR controllers with the EWMA controller in scenario 2.

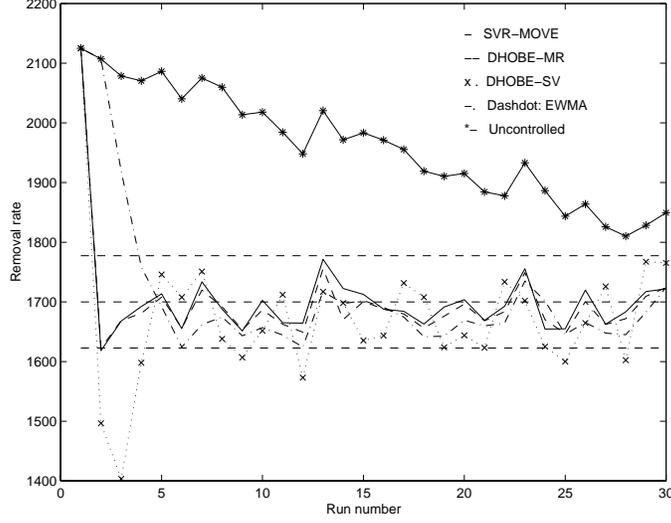


Figure 4: Comparison of the set-valued RtR controllers with the EWMA controller in scenario 3.

Weight	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Scenario 1	82.13	52.60	42.44	37.75	37.29	39.52	41.50	45.37	59.17
Scenario 2	269.42	186.09	157.19	140.62	133.38	132.39	138.51	149.07	160.79
Scenario 3	157.88	132.27	121.24	122.15	133.13	148.59	197.90	366.64	1040.67

Table 1: *RMSD* values of the EWMA controller in three scenarios

The gains of the SVR-MOVE controller and the DHOBE-MR controller over the EWMA controller with the best weight (0.6) were 29.45% and 25.30% respectively.

Scenario 3

In real life, the underlying process model parameters are unknown. To address this model uncertainty, in this scenario, the initial process model parameters used by the controllers were set at 80% of the true parameter values of the underlying process. Hence, the initial process model was:

$$\hat{y}_t = -1106.08 + 40.144u_{1,t} - 5.32u_{2,t} + 130.72u_{3,t} + 6.76u_{4,t} \quad (7)$$

This large model error should cause the output of the process to change abruptly at the beginning of the simulation. The *RMSD* values of the EWMA controller with respect to different weights were listed in the third row of Table 1. The *RMSD* values for the SVR-MOVE controller, DHOBE-MR controller, DHOBE-SV controller and the uncontrolled process were given in the third row of Table 2. The simulation results are shown in Figure 4. The weight of the EWMA controller was 0.3 in this figure. Still, the SVR-MOVE controller and the DHOBE-MR controller performed better than the other two controllers. Even the DHOBE-SV controller performed better than the EWMA controller in this case. The gains of the SVR-MOVE controller and the DHOBE-MR controller over the EWMA controller with the best weight (0.3) were 28.27% and 29.84% respectively.

Scenario	SVR-MOVE	DHOBE-MR	DHOBE-SV	Uncontrolled
1	36.70	39.25	57.36	179.18
2	93.40	98.90	158.92	523.36
3	86.96	85.06	112.98	284.60

Table 2: *RMSD* values of the set-valued RtR controllers in three scenarios

3.2 Comparison of the DHOBE-MR Controller with the OAQC

Detailed descriptions of the OAQC can be found in [4]. To make the comparison between the DHOBE-MR controller and the OAQC fair, we used exactly the same experimental conditions as those described in [4]¹. The platform used was another CMP process. The underlying “real” process was given by [4]:

$$y_{1,t} = 1563.5 + 159.3u_{1,t} - 38.2u_{2,t} + 178.9u_{3,t} + 24.9u_{4,t} - 67.2u_{1,t}u_{2,t} - 46.2u_{1,t}^2 - 19.2u_{2,t}^2 - 28.9u_{3,t}^2 - 12u_{1,t}t' + 116u_{4,t}t' - 50.4t' + 20.4t'^2 + \epsilon_{1,t} \quad (8)$$

$$y_{2,t} = 254 + 32.6u_{1,t} + 113.2u_{2,t} + 32.6u_{3,t} + 37.1u_{4,t} - 36.8u_{1,t}u_{2,t} + 57.3u_{4,t}t' - 2.42t' + \epsilon_{2,t}, \quad (9)$$

where

$t' = (t - 53)/53$, $\epsilon_{1,t} \sim N(0, 60^2)$, $\epsilon_{2,t} \sim N(0, 30^2)$, and

$y_{1,t}$ was the removal rate; its target value was 2000.

$y_{2,t}$ was the with-in wafer non-uniformity; its target value was 100.

$u_{1,t}$ was the platen speed.

$u_{2,t}$ was the back pressure.

$u_{3,t}$ was the polishing down-force.

$u_{4,t}$ was the profile.

The process model was rather complex as it included both quadratic and two factor interaction terms. The inputs $u_{1,t}$, $u_{2,t}$, $u_{3,t}$ and $u_{4,t}$ were scaled to fit in the range $[-1, 1]$. For output $y_{1,t}$, the larger the value, the better the performance; and for $y_{2,t}$, the smaller the value, the better the performance.

Following [4] to approximate the underlying nonlinear process, we used exactly the same two reduced models as in [4], a quadratic form model and a linear form model respectively. These reduced models provided us with an opportunity to test the controllers’ robustness to model errors for nonlinear processes.

1) Approximate the Underlying Process – A Quadratic Model (Scenario 1)

The “real” process model of equations (8) and (9) was unknown to the DHOBE-MR controller. As in [4], the following quadratic model was used to approximate the “real” process:

$$\hat{y}_{1,t} = 1600 + 150u_{1,t} - 40u_{2,t} + 180u_{3,t} + 25u_{4,t} - 30u_{1,t}^2 - 20u_{2,t}^2 - 25u_{3,t}^2 - 60u_{1,t}u_{2,t} - 0.9t \quad (10)$$

and

$$\hat{y}_{2,t} = 250 + 30u_{1,t} + 100u_{2,t} + 20u_{3,t} + 35u_{4,t} - 30u_{1,t}u_{2,t} + 0.05t \quad (11)$$

As easily seen, the approximate model was different from the underlying process model, which meant that there existed a model error at the beginning of the control experiment. Moreover, the noises in equations (8) and (9) were unknown to the DHOBE-MR controller, so that the controller had to compensate for such disturbances by post-measurements. Our simulation results of the process (simulated by equations (8) and

¹Because the parameters of the OAQC were unknown, we could not replicate the OAQC results of [4] in our simulations. Instead, the simulation data of the OAQC performance were used as appeared in [4]

Scenario	Method	\bar{y}_1	\bar{y}_2	S_{y_1}	S_{y_2}	$RMSD_1$	$RMSD_2$
1	OAQC	1719.7	168.4	70.4	40.1	288.9	79.2
1	DHOBE-MR	1754.7	157.3	84.5	35.0	259.7	67.5
2	OAQC	1718.2	165.7	72.1	42.0	291.0	78.2
2	DHOBE-MR	1781.9	165.0	84.5	36.1	234.2	74.8
3	OAQC	1661.2	189.2	89.2	43.5	350.2	99.2
3	DHOBE-MR	1741.4	189.1	108.7	35.6	280.8	96.0

Table 3: Performance measure ($RMSD$ values) of the OAQC and the DHOBE-MR controller for the three scenarios. The response target values for $y_{1,t}$ and $y_{2,t}$ were equal to 2000 and 100 respectively. (The data for the OAQC were from [4].)

(9)) controlled by the DHOBE-MR controller (designed using the reduced model of equations (10) and (11)) are shown in Figure 5 (a). The two dashed lines in the plot are the outputs of the uncontrolled process. The solid lines in the plot with symbols (i.e., circles or squares) depict the controlled outputs. One can see that the controlled process outputs (the removal rate and the non-uniformity) are closer to the targets than the uncontrolled outputs during the entire simulation run.

2) Approximate the Underlying Process – A Linear Model (Scenario 2)

In this scenario, following [4], we used a linear model to fit the underlying process at the beginning:

$$\hat{y}_{1,t} = 1600 + 150u_{1,t} - 40u_{2,t} + 180u_{3,t} + 25u_{4,t} - 0.9t \quad (12)$$

$$\hat{y}_{2,t} = 250 + 30u_{1,t} + 100u_{2,t} + 30u_{3,t} + 35u_{4,t} + 0.05t \quad (13)$$

As the underlying process was approximated well by the linear model (based on the results of [4] and our own simulations), the DHOBE-MR controller based on this model also performed well (Figure 5 (b)).

3) A Quadratic Model with Step Disturbances (Scenario 3)

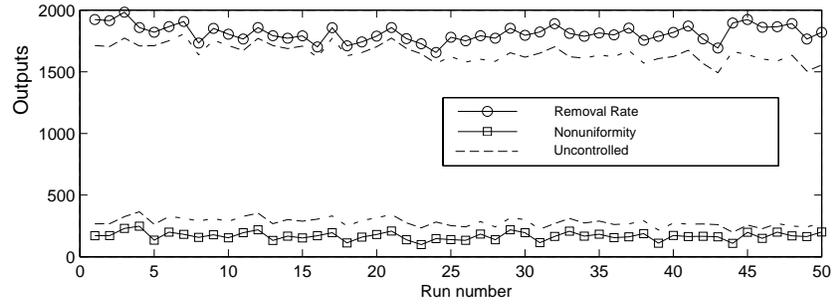
In this simulation, following [4], two shift disturbances (step disturbances) were fed into the underlying process. The quadratic initial model was used and the constraints were the same as before. The shift for the first response $y_{1,t}$ happened at $t = 20$ with magnitude -100. At $t = 30$, another shift occurred with magnitude 50 for $y_{2,t}$ ². Our simulation results are shown in Figure 5 (c). The DHOBE-MR controller performed well in this case also.

The final results with regard to the statistical variance analysis are listed in Table 3. The data on the OAQC performance provided here follow precisely the results in [4]. The following data are also listed in Table 3 for the convenience of comparison:

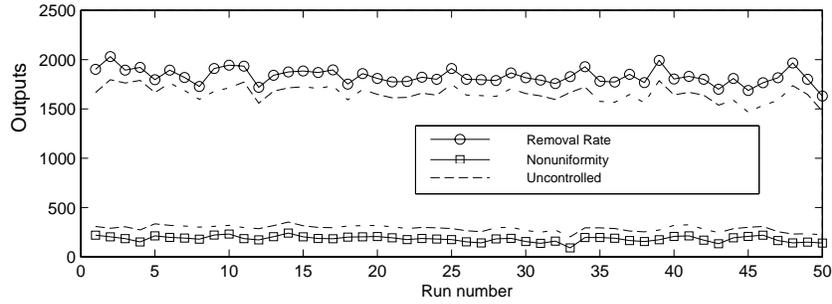
- \bar{y}_i : the mean of the i th output of the process.
- S_{y_i} : the standard deviation of the i th output of the process. The smaller its value, the better.

Table 3 shows that the mean values of the process responses (outputs) controlled by the DHOBE-MR controller are closer to the target values than those of the OAQC. The $RMSD$ values of the outputs controlled by the DHOBE-MR controller are smaller than those of the OAQC. Only the standard deviations of response $y_{1,t}$ controlled by the DHOBE-MR controller are larger than those of the OAQC. But standard deviation is not the performance metric of interest. Therefore, the DHOBE-MR controller performed slightly better than the OAQC in all scenarios. For more comparisons of the DHOBE-algorithm-based RtR controllers with the OAQC, we refer readers to [5].

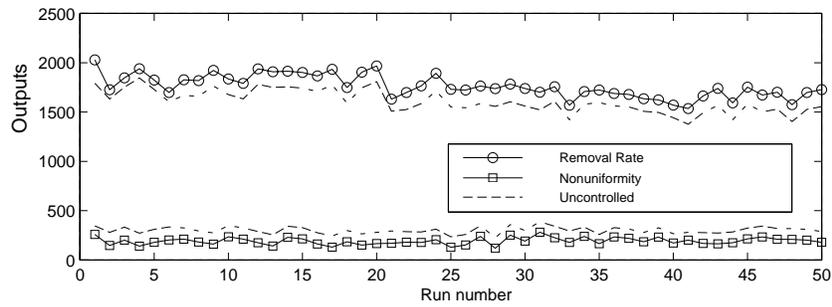
²The magnitudes of these shifts were too small to be discerned with the other noises in [4]. However, to make the comparison fair, we used the same values as in [4].



(a)



(b)



(c)

Figure 5: A CMP process controlled by the DHOBE-MR Controller. The outputs are the removal rate and the non-uniformity respectively. (a) Scenario 1: A quadratic model was used to approximate the underlying process. (b) Scenario 2: A linear model was used to approximate the underlying process. (c) Scenario 3: A step disturbance with magnitude -100 happened to response $y_{1,t}$ at run 20; another step disturbance with magnitude 50 happened to response $y_{2,t}$ at run 30.

4 Conclusions

RtR control methods are classified and compared in this paper. Depending on the property of a semiconductor process, we can select a proper RtR control method. For a process that can be approximated well by a linear model, an EWMA controller usually works well and it is unnecessary to apply more complex control methods. Many semiconductor processes are subjected to small drift disturbances or other small-sized noises. These perturbations can be compensated by using the EWMA method or some other linear-model-based methods. The linear-model-based RtR controllers are usually simple to implement and very efficient to deal with such small disturbances. When there exists a large deviation, a rapid mode can be added to the EWMA controller [14]. The rapid mode may be used to return the process outputs quickly back to target values. However, to the authors' best knowledge, there are no explicit formulae for applying the rapid mode.

Many plasma processes have been shown to exhibit small to large nonlinearities in behavior. The photoresist process requires a dynamic process model too. In this case, it may be necessary to deploy nonlinear model based RtR controllers. The LSE method based RtR controllers can more accurately describe a nonlinear process than the EWMA controller in general. They are also more complex to design than the EWMA controller. The set-valued RtR controllers are even more complex than the LSE method based RtR controllers. They can use a high-order polynomial model to fit a process. Hence, the set-valued RtR controllers may provide a better fit for a process than the LSE method based RtR controllers.

In the application of the set-valued RtR controllers, the SVR-MOVE controller and the DHOBE-MR controller are recommended. They had good performance under various conditions in our simulations. The SVR-MOVE controller and the DHOBE-MR controller performed especially well when there existed large step disturbances and model errors. The DHOBE-SV controller is more conservative and may cause large variability. This controller is usually not recommended in application.

The other nonlinear-model-based RtR controllers such as the ANN-based RtR controllers may provide an even better fit for a severe nonlinear process than the set-valued RtR controllers. But in application, the other nonlinear-model-based RtR controllers have not shown superb performance. In a comparative simulation in [18], the SVR-MOVE controller had slightly better performance than an ANN-based RtR controller. Therefore, more work needs to be done to improve the other nonlinear-model-based RtR control methods.

References

- [1] J. S. Baras, N. S. Patel, "Designing response surface model-based run-by-run controllers: A worst case approach", *IEEE Trans. Components, Packaging and Manufacturing Technology*, vol. 19, no. 2, pp. 98-104, 1996.
- [2] D. Boning, W. Moyne, T. Smith, etc., "Run by run control of chemical-mechanical polishing," 1995 *IEEE/CPMT Int'l Electronics Manufacturing Technology Symposium*, pp. 81-87, 1995.
- [3] S. W. Butler and J. A. Stefani, "Supervisory run-to-run control of polysilicon gate etch using in situ ellipsometry," *IEEE Trans. Semiconductor Manufacturing*, vol. 7, pp. 193-201, 1994.
- [4] E. D. Castillo and J. Y. Yeh, "An adaptive run-to-run optimizing controller for linear and nonlinear semiconductor processes", *IEEE Trans. on Semiconductor Manufacturing*, Vol. 11, No. 2, pp. 285-295, 1998.
- [5] H. Deng, *Run to Run Controller for Semiconductor Manufacturing*. Master Thesis, University of Maryland, College Park, 1999.

- [6] S. Dasgupta and Y.F.Huang, "Asymptotically convergent modified recursive least-squares with data-dependent updating and forgetting factor for systems with bounded noise", *IEEE Trans. Information Theory*, vol IT-33, No. 3, p383-392, 1987.
- [7] C. D. Himmel and G. S. May, "Advantages of plasma etch modeling using neural networks over statistical techniques", *IEEE Trans. Semiconductor Manufacturing*, vol. 6, no. 2, 1993.
- [8] Y. L. Huang, *et al*, "Constructing a reliable neural network model for a plasma etching process using limited experimental data", *IEEE Trans. Semiconductor Manufacturing*, vol. 7. no. 3, 1994.
- [9] A. Ingolfsson and E. Sachs, "Stability and sensitivity of an EWMA controller", *Journal of Quality Technology*, vol. 25, pp. 271-287, 1993.
- [10] E. Palmer, W. Ren, C. J. Spanos, "Control of photoresist properties: A Kalman filter based approach," *IEEE Trans. Semiconductor Manufacturing*, vol. 9, no. 2, pp. 208-214,1996.
- [11] J. A. Mullins, *et al*, "An evaluation of model predictive control in run to run processing in semiconductor manufacturing", *SPIE*, vol. 3213, pp. 182-189, 1997.
- [12] C. D. Natale, *et al*, "Modeling of APCVD-doped silicon dioxide deposition process by a modular neural network", *IEEE Trans. Semiconductor Manufacturing*, vol. 12, no. 1, 1999.
- [13] Z. Ning, *et al*, "A comparative analysis of run-to-run control algorithms in the semiconductor manufacturing industry", 1996 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pp. 375-381, 1996.
- [14] E. Sachs, A. Hu, and A. Ingolfsson, "Run by run process control: combining SPC and feedback control", *IEEE Trans. Semiconductor Manufacturing*, vol. 8, pp. 26-43, 1995.
- [15] T. H. Smith and D. S. Boning, "A self-tuning EWMA controller utilizing artificial neural network function approximation techniques", 1996 *IEEE/CPMT Int'l Electronics Manufacturing Technology Symposium*, pp. 355-363, 1996.
- [16] X. A. Wang and R. L. Mahajan, "Artificial neural network model-based run-to-run process controller", *IEEE Trans. Components, Packaging and Manufacturing Technology, Part C*, vol. 19, no. 1, 1996.
- [17] C. Zhang, H. Deng and J. S. Baras, "The set-valued run-to-run controller with ellipsoid approximation," *Proceedings of the AEC/APC 2000, Lake Tahoe, NV*, vol. 1, pp. 167-178, 2000.
- [18] C. Zhang, H. Deng and J. S. Baras, "Comparison of run-to-run control methods in semiconductor manufacturing processes," *Proceedings of the AEC/APC 2000, Lake Tahoe, NV*, vol. 3, pp. 889-896, 2000.

"The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government."