TECHNICAL RESEARCH REPORT

Comparison of Run-to-Run Control Methods in Semiconductor Manufacturing Processes

by Chang Zhang, Hao Deng, John S. Baras

T.R. 2000-33

Comparison of Run-to-Run Control Methods in Semiconductor Manufacturing Processes

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

Department of Electrical and Computer Engineering
University of Maryland, College Park,
College Park, MD, 20742

August 22, 2000

Abstract

Run-to Run (RtR) control plays an important role in semiconductor manufacturing. In this paper, RtR control methods are generalized. The set-valued RtR controllers with ellipsoid approximation are compared with other RtR controllers by simulation according to the following criteria: A good RtR controller should be able to compensate for various disturbances, such as process drifts, process shifts (step disturbance) and model errors; moreover, it should be able to deal with limitations, bounds, cost requirement, multiple targets and time delays that are often encountered in real processes. Preliminary results show the good performance of the set-valued RtR controller. Furthermore, this paper shows that it is insufficient to use linear models to approximate nonlinear processes and it is necessary to develop nonlinear model based RtR controllers.

1 Introduction

Run-to Run (RtR) control plays an important role in semiconductor manufacturing. The RtR controller is a model-based process control system that combines the advantage of both

*Phone: 301-405-6578, Fax: 301-314-9281, Email: zchang@isr.umd.edu

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

‡This work was supported by the Center for Satellite and Hybrid Communication Networks and SEIL lab, under NASA cooperative agreement NCC3-528

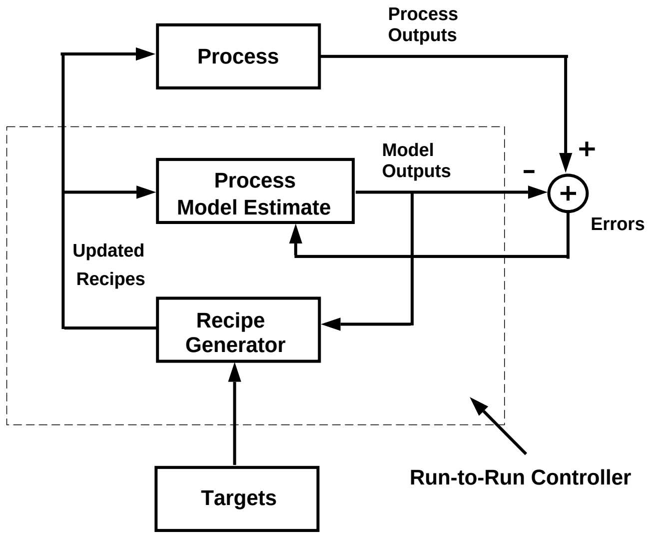

Figure 1: Structure of a RtR Controller.

the statistical process control and the feedback control. In general, the goal of the controller is to reduce the variability of the process outputs, as measured by the mean square errors between the process outputs and the targets. It can be achieved by adjusting process inputs (recipes) at the beginning of each run. A good RtR controller should be able to compensate for various disturbances, such as process drifts, process shifts (step disturbances) due to maintenance or other factors, model or sensor errors, etc. Moreover, it should be able to deal with limitations, bounds, cost requirement, multiple targets and time delays that are often encountered in real processes.

A typical block diagram of a RtR controller is shown in Figure 1. The controller provides recipes (inputs) based on post-process measurements at the beginning of each run; then it updates the process model according to the measurements at the end of the run; finally, it provides new recipes for the next run of the process. The RtR controller does not modify recipes during a run because of the following reasons: 1) Cost. It is usually very expensive to obtain real-time information in a semiconductor process; 2) Stability. Frequent changes of inputs to the process may increase the variability of the process's outputs and even make the process unstable. The initial process model is derived from former experiments such as the response surface model (RSM) methods. When the controller is online, the model within the controller is updated according to the new measurements from run to run. Different model updating methods leads to different kinds of controllers.

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 EWMA controller, the ANN-EWMA controller and the OAQC controller are introduced in section 3.1-3.3; in section 3.4, two different set-valued RtR controllers are compared; finally, conclusions and summary are given in the last section.

2 Classification of RtR Control Methods

In general, RtR control methods fall into the following six categories (Because some RtR controllers are commercial products, they may not be listed here).

1. The Exponentially Weighted Moving Average (EWMA) method. It is one of the most popular RtR control methods. It uses history data to linearly update process models by giving less weights to the old data [19],[2]. The EWMA method is applicable to processes that can be approximated by linear models. Sometimes, multiple linear models are used and a supervisory module is used to switch the models. For a detailed theoretical analysis of the EWMA method, please refer to [13]. There are lots of modifications to the EWMA method. For example, the double exponential forecasting filter method [3] uses a Predictor Corrector Controller (PCC) to eliminate the impact of machine and process drift. The Artificial Neural Network (ANN) EWMA [20] method makes the EWMA method applicable to some higher order linear-in-parameter processes. J. A. Mullins uses a discrete process model for a linear process that has the ability of model-predictive control [16].

2. The machine learning algorithm. A typical example is the Knowledge Based Interactive Controller (KIRC) [18]. It is a machine learning algorithm for RtR control that 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 [18] shows that the KIRC is only applicable to linear processes.

3. Least Square Recursive (LSR) method. It recursively approximates the process model by minimizing the least square error between the model outputs and targets. Typical examples are the Optimized Adaptive Quality Control (OAQC) [4] method and the Kalman filter approach [15]. In its optimizing mode, the OAQC method updates the model at every run; in the controller mode, it uses a quadratic cost function to maintain the response of the process at the desired target with regards to the variation of the tunable parameters. It integrates the multivariate control chart as a dead-band to the controller in order to erase outliers. The Kalman filter approach is used to recursively adjust the coefficients of a simple static process model. The effectiveness of the scheme is demonstrated by experiments. Currently, the LSR method is limited to polynomial processes that can be approximated by a second-order equation.

4. The probabilistic approach [10]. It uses the probability theory to analyze the process. Analytic formulas for the probability of stability are given in the particular case of an EWMA controller. However, there is a key question of reliability of this methodology. Furthermore, it assumes that the noises are Gaussian to derive the formulas, which limits the practical meaning of this method. At present, it is limited to first order processes, though it has the potential to be used in higher order processes.

5. The Artificial Neural Network (ANN) method. It was shown that the ANN has great potential in modeling severe nonlinear semiconductor processes [11], [17], [12]. However, a drawback of the ANN method is that it does not supply an explicit model for the process. Thus, it causes difficulties when one tries to apply optimal control to adjust recipes. Wang[21] uses a Taylor expansion to find a linear model to describe the ANN model. But it finally becomes a linear RtR control method. Its performance is only comparable to that of an ordinary EWMA controller. T. H. Smith [20] uses the ANN EWMA method to control a second order process. It is successful for small disturbances or parameter variations in a limited number of runs (only 40). For a large model error, the process controlled by it becomes unstable. The reason is that the controller uses an EWMA module to feed into the ANN model, which limits the ability of the ANN to approximate a nonlinear process. D. Dong and Zafiriou use an ANN approach to control batch-to-batch processes in chemical engineering that can be modeled by a first principle model [8]. The processes are different from semiconductor processes, which are usually much more complex. Therefore, to the best of the authors' knowledge, there has not been a successful scheme to apply the ANN approach to the RtR control of severe nonlinear semiconductor processes.

6. The set-valued approach. The set-valued RtR controller is much more robust than other regular RtR controllers. It seeks a safe estimate of the process model in the feasible parameter set in each run. The identified model is insensitive to various noises [23]. It was first proposed by Baras [1] to be applied in RtR control. The main difficulty of the set-valued based RtR controller is the excessive computational time required to calculate the feasible sets and solving the optimization problem within this set. The problem can be simplified by using ellipsoids to approximate the feasible sets. There are mainly two ellipsoid algorithms available at present: the Modified Optimal Volume Ellipsoid (MOVE) algorithm and the Dasgupta Huang Optimal Bounding Ellipsoid (DHOBE) algorithm. For details of these two algorithms, please refer to [22] and [23]. The corresponding controllers will be called the SVR-MOVE controller and the SVR-DHOBE controller respectively.

Next, we are going to compare the set-valued RtR controllers with several typical RtR controllers.

3 Comparison of RtR Control Methods by Simulation

3.1 Comparison of the SVR-MOVE Controller with the EWMA Controller

In this section, we are going to compare the SVR-MOVE controller with the EWMA controller. The model (Equation (1) and (2)) comes from a Low Pressure Chemical Vapor Deposition (LPCVD) furnace process. In this process, two objects R_1 and R_2 are controlled. They are the deposition rates in $\overset{o}{A}\ /min$ on the first and last wafer respectively. The targets are fixed at 169.75 $\overset{o}{A}$ $/min$ and 141.7 $\overset{o}{A}$ $/min$ respectively. We want to maintain the outputs of the process as close to the targets as possible.

$$R_1 = exp(c_1 + c_2 lnP + c_3 T^{-1} + c_4 Q^{-1}) \tag{1}$$

$$R_2 = R_1 \frac{1 - S' C_{gs} R_1 Q^{-1}}{1 + S' C_{gs} R_1 Q^{-1}} \tag{2}$$

where T stands for the temperature in K, P the pressure in mtorr and Q the silane flow rate in sccm. They are the inputs (recipes) to the process. We adjust them to maintain the process outputs on targets. The process parameters are $c_1 = 20.65$, $c_2 = 0.29$, $c_3 = -15189.21$, $c_4 = -47.97$, $S' = 4777.8$ and $C_{gs} = 1.85 \times 10^{-5}$. Here the units are omitted for convenience.

The SVR-MOVE controller is compared with the EWMA controller in two cases: 1) The noises are white noises and drifts; The drifts in process R_1 and R_2 are equal to -0.3 in each run. 2) the noises are white noises and shifts. The shifts in the process occur at run 4. The simulation results are shown in Figure 2 and Figure 3 respectively. The target, the 3σ upper bound and lower bound are shown in these figures by three horizontal straight lines. The weight parameter for the EWMA controller is 0.35. It is obtained by selecting the weight that has the optimal performance among multiple weight parameters. Though the EWMA method can control multiple objects, it is used to control only the single process R_1. Therefore, the performance of the EWMA controller can not be better if it is used to control two processes. The SVR-MOVE controller is applied to control two targets R_1 and R_2. From Figure 2, it can be seen that both controllers control R_1 well. The controlled process stays in the 3σ region satisfactorily. Figure 3 shows that the SVR-MOVE controller returns the output to the target immediately after detecting the step disturbance; The EWMA controller needs more steps to return the output within the 3σ bound.

The comparison shows that under a drift disturbance, The performance of the SVR-MOVE controller is comparable to that of the EWMA controller. Under a shift disturbance, its performance is better than that of the EWMA controller.

3.2 Comparison of the SVR-MOVE Controller with the ANN EWMA Controller and the EWMA Controller

Many semiconductor processes can be subjected to small shifts or drift changes. These perturbations can be compensated by using the EWMA method or some other linear model based methods. Unfortunately, this is not always the case. For example, many plasma processes have been shown to exhibit small to large nonlinearities in behavior. Furthermore, the photoresist process and the Chemical Mechanical Planarization (CMP) process require dynamic process models too. Therefore, it is necessary to develop nonlinear algorithms to

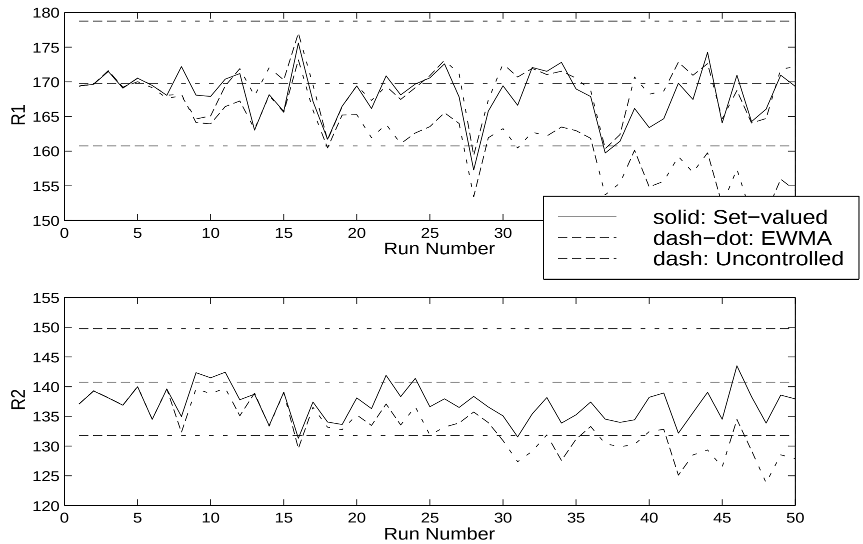

Figure 2: An LPCVD furnace process controlled by the SVR-MOVE controller and the EWMA controller. The disturbances include white noises and drifts.

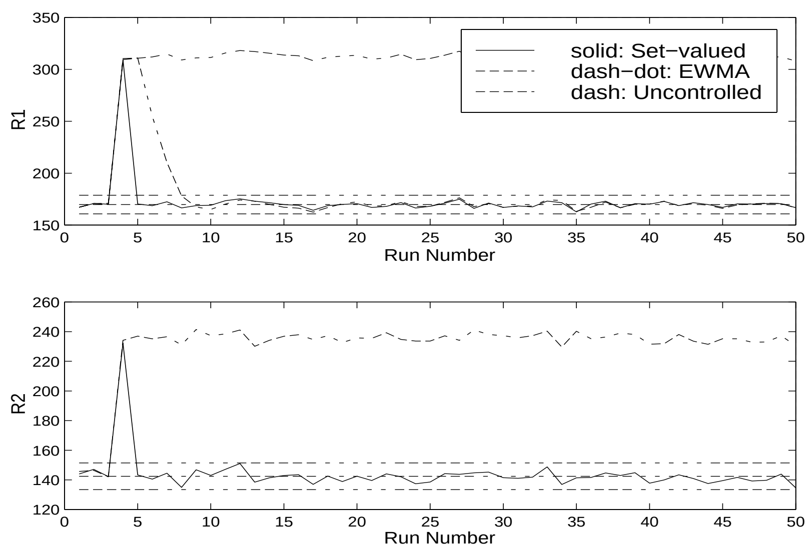

Figure 3: An LPCVD furnace process controlled by the SVR-MOVE controller and the EWMA controller. The disturbances include white noises and shifts.

solve this type of problems. In this comparison, we will see that the EWMA controller and the ANN-EWMA controller can not work well for nonlinear processes.

The comparison is based on the model in [20]. The ANN EWMA approach was proposed by T. H. Smith to monitor nonlinear processes. As in the EWMA method, it assumes that the underlying process model is relatively static, and to adapt model offsets is enough to control the process. In comparison, the process output $y_p[k]$ is of the form:

$$y_p[k] = b_p + f_p(x[k]) + v[k] + \delta \cdot k \tag{3}$$

where b_p is the actual process offset, $x[k]$ is the inputs, $v[k]$ is normally distributed white noise with zero mean and covariance matrix Λ, δ is a drift term, k is the run number, and $f_p(x[k])$ is a full second-order polynomial function of the form:

$$f_p(x[k])(i) = \sum_{j=0}^{q} \beta_p(i,j)x(i)x(j), i = 1, ..., p \tag{4}$$

where $\beta_p(i,j)$ are the perturbed versions of the nominal coefficients $\beta(i,j)$. They have the relation:

$$\beta_p(i,j) = (1 + \varphi_{ij})\beta(i,j) \tag{5}$$

where φ_{ij} is uniformly distributed zero-mean random variables with variance σ. For a small model error, $\sigma = 0.1$; for a large model error, $\sigma = 0.3$. For more information about the variances, please refer to [20].

It is found that the EWMA controller is often unstable in both cases for even the most conservative weights; the ANN-EWMA controller is stable under a small model error ($\sigma = 0.1$); it becomes unstable when there is a large model error (σ=0.3) [20]. The simulation results for the SVR-MOVE controller under small and large model errors are shown in Figure 4 and 5 respectively. It can be seen that the SVR-MOVE controller controlled processes are stable in both cases. This comparison shows that the set-valued RtR controller has stronger ability to deal with model errors and noises than the EWMA controller and the ANN-EWMA controller. The SVR-MOVE controller is much more robust than them.

3.3 Comparison of the SVR-DHOBE Controllers with the OAQC Method

According to the method of choosing the estimate in the ellipsoid, there are two DHOBE algorithm based RtR controllers available: If the center of the ellipsoid is chosen as the estimate, then we call the controller DHOBE-MR controller; if the point in the ellipsoid which minimizes the worst-case cost is chosen as the estimate, then we call it the DHOBE-SV controller. Detailed introductions about the OAQC method can be found in [4]. Detailed

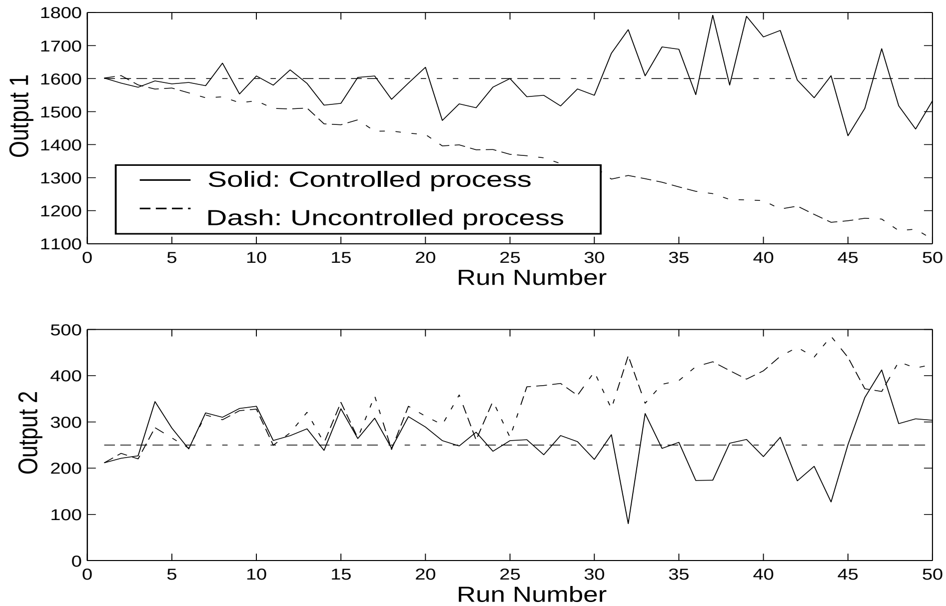

Figure 4: A process controlled by the SVR-MOVE controller under a small model error.

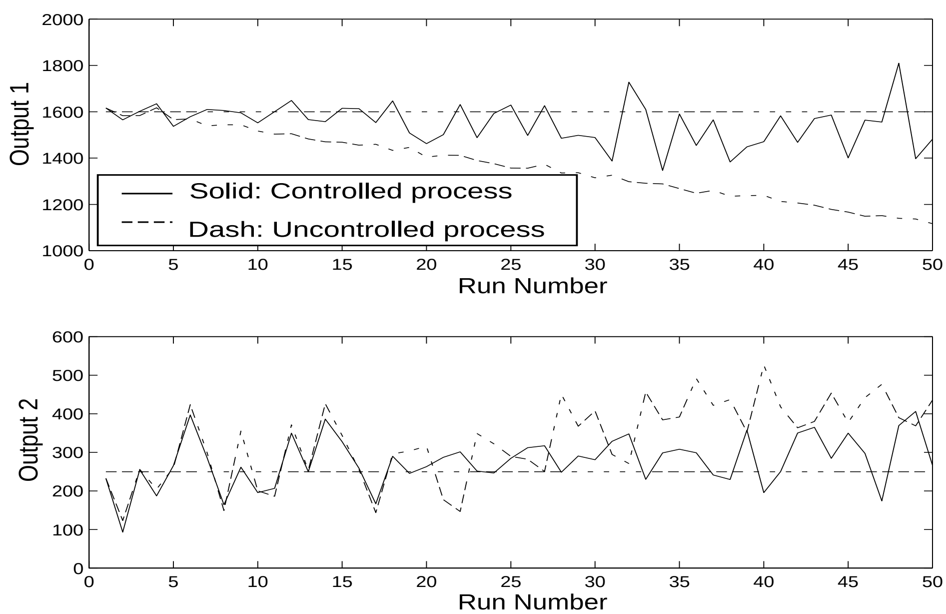

Figure 5: A process controlled by the SVR-MOVE controller under a large model error.

Scenario	Method	$\bar{y}_1$	$\bar{y}_2$	S_{y1}	S_{y2}	MSD_1	MSD_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
1	DHOBE-SV	1787.7	168.1	82.8	34.7	228.2	76.9
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
2	DHOBE-SV	1807.4	177.5	85.9	36.1	211.9	86.1
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
3	DHOBE-SV	1747.0	190.8	109.2	37.5	275.9	98.3

Table 1: Comparison of the SVR-DHOBE controllers with the OAQC method for CMP 4x2 models

comparison can be found in [7]. The SVR-DHOBE controllers are simulated under exactly the same circumstances as the OAQC method. Two responses y_1 and y_2 are controlled. For response 1, a large value is preferred; for response 2, a small value is desired. The final results with regards to the statistical variance analysis are listed in Table 1 and Table 2.

In the tables, the symbols have the meaning:

- $\bar{y}_i$ stands for the sampling values from the real process's ith output.
- S_{y_i} is the standard deviation of the process's ith output.
- MSD_i is the square root of mean square deviation of the process's ith output from its target value.

Table 1 shows the result for a CMP 4x2 model. For response 1, the means of the DHOBE algorithm based RtR controllers are better than the OAQC controller and their standard deviations are comparable to the OAQC controller. For the second response, the means of the DHOBE algorithm based RtR controllers are comparable to the OAQC controller and the standard deviations are better than the OAQC controller.

Table 2 shows the simulation result for a CMP 3x2 model. In scenario 1, we use a second order model to approximate the process model; In scenario 2, we use a linear model to approximate the process model. It can be seen from the table that the mean square errors for scenario 2 are about two times larger than those for scenario 1. It shows that it is insufficient to use linear models to approximate nonlinear processes.

Scenario	Method	$\bar{y}_1$	$\bar{y}_2$	S_{y1}	S_{y2}	MSD_1	MSD_2
1	OAQC	2069.9	478.8	143.8	53.5	193.5	95.0
1	DHOBE-MR	2005.5	490.5	139.7	41.2	235.9	98.6
1	DHOBE-SV	2002.8	490.4	141.4	42.7	238.9	98.9
2	OAQC	1950.4	595.0	430.7	99.6	543.9	220.4
2	DHOBE-MR	1921.5	663.9	457.0	99.9	568.4	271.9
2	DHOBE-SV	1921.8	659.6	381.6	71.6	499.0	256.8

Table 2: Comparison of the SVR-DHOBE controllers with the OAQC method for CMP 3x2 models

3.4 Comparison of the SVR-MOVE Controller with the SVR-DHOBE Controller

Both the SVR-MOVE controller and the SVR-DHOBE controller use ellipsoids to approximate the feasible parameter sets and they both update the process models only when it is necessary. The difference between them lies in: The derivation of the MOVE algorithm is based on a geometric point of view; the DHOBE algorithm uses a Recursive Least Square (RLS) type scheme to update the ellipsoid.

The comparison was made by simulation for two photoresist processes.

3.4.1 An Almost Linear Photoresist Process I

The following is the model used in the photoresist process I [14].

$$\begin{aligned} T = & - 13814 + \frac{2.54 \cdot 10^6}{\sqrt{SPS}} + \frac{1.95 \cdot 10^7}{BTE\sqrt{SPS}} \\ & - 3.78BTI - 0.28SPT - \frac{6.16 \cdot 10^7}{SPS} \end{aligned} \tag{6}$$

where T is the resist thickness in Angstroms and the target is fixed at 12373.621 Angstroms. We want the output T to be as close to the target as possible. SPS is the spin speed in RPM, SPT the spin time in seconds, BTI the baking time in seconds and BTE the baking temperature in degrees Celsius. They are the inputs (recipes) to the process, which are confined to:

$$4500 < SPS < 4700$$

$$
\begin{aligned}
15 < SPT < 90 \\
105 < BTE < 135 \\
20 < BTI < 100
\end{aligned}
$$

After changing process variables, it can be simplified to an almost linear process. The simplified model is shown in the following equation:

$$
\begin{aligned}
T = & - 13814 + 2.54 \cdot 10^6 u_1 + 1.95 \cdot 10^7 u_1 u_2 \\
& - 3.78 u_3 - 0.28 u_4 - 6.16 \cdot 10^7 {u_1}^2
\end{aligned} \tag{7}
$$

where:

$$
\begin{aligned}
u_1 &= \frac{1}{\sqrt{SPS}} \\
u_2 &= \frac{1}{BTE} \\
u_3 &= BTI \\
u_4 &= SPT
\end{aligned}
$$

The output of the process in each run is:

$$
y_k = T + d_1 \cdot k + v_1 \tag{8}
$$

where $d_1 = -0.3$ and v_1 is Gaussian with zero mean and variance 9.

The simulation results for the SVR-MOVE controller and the SVR-DHOBE controller are shown in Figure 6 and Figure 7 respectively. We can see that both controllers can control the almost linear process well under the disturbance of drift. However, the SVR-DHOBE controller controlled process has some overshoots, which affect the control quality. In the following, white noise in the process is removed, and only the drift exists as the disturbance. From Figure 8 and 9, it can be seen clearly that the SVR-MOVE controller works well, but the process controlled by the SVR-DHOBE controller has obvious over-shoots at run 64 and 65.

3.4.2 A Full Second-order Nonlinear Photoresist Process II

The model used here is a full second-order nonlinear process II [14].

$$
R = \quad 134.4 - 0.046 SPS + 0.32 SPT - 0.17 BTE
$$

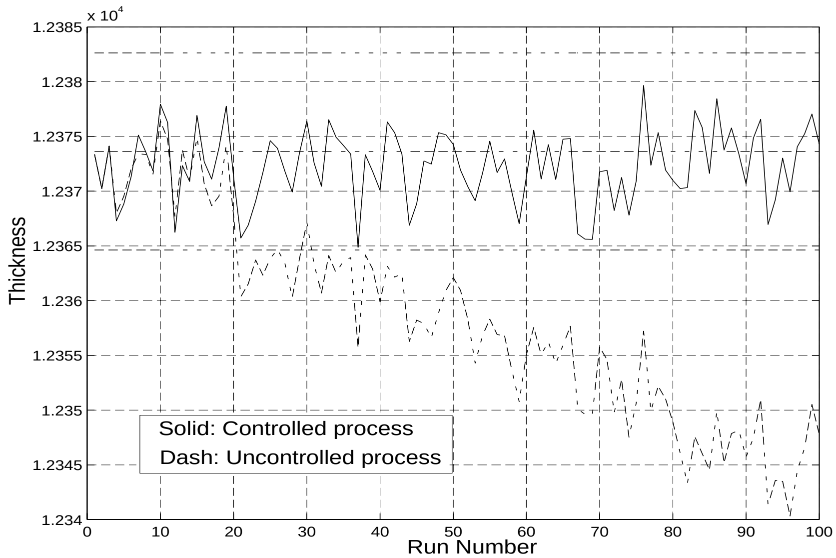

Figure 6: Photoresist process I controlled by the SVR-MOVE controller under drift.

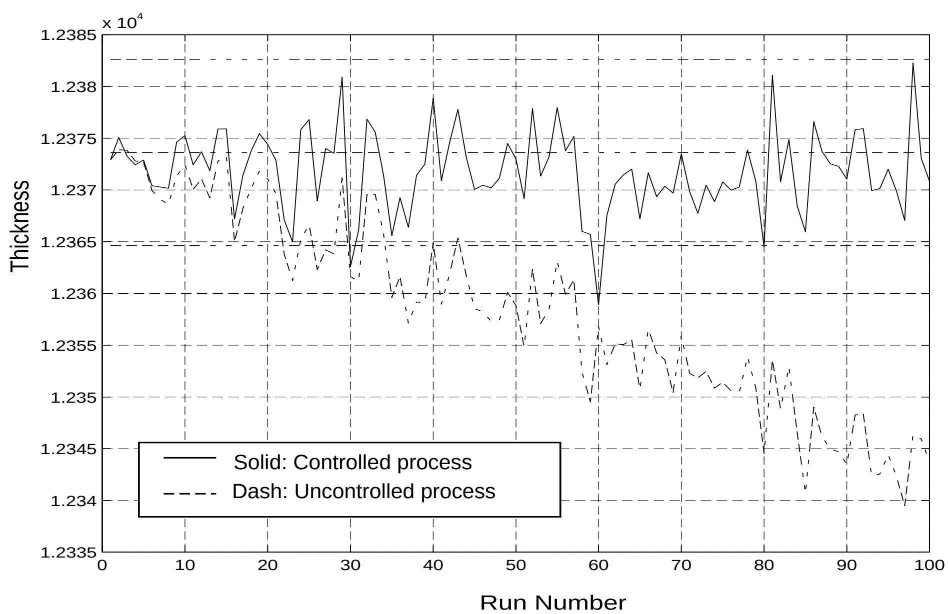

Figure 7: Photoresist process I controlled by the SVR-DHOBE controller under drift.

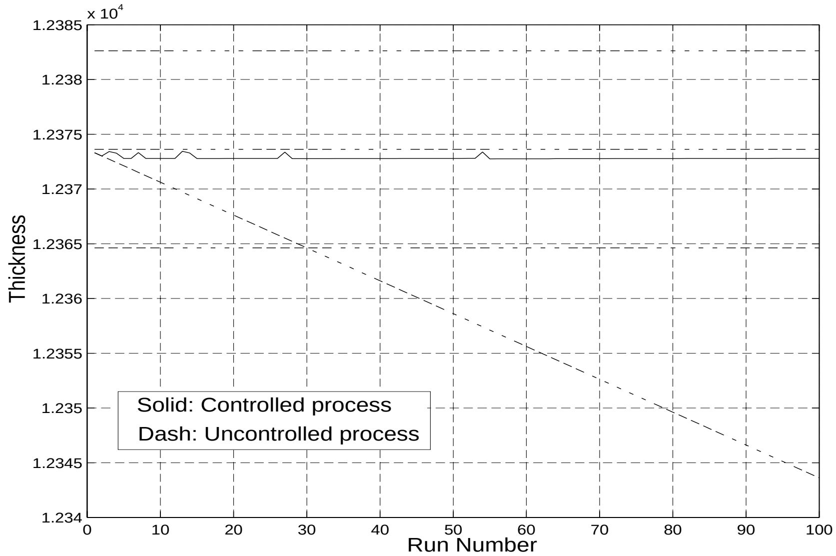

Figure 8: Photoresist process I controlled by the SVR-MOVE controller. At this time, white noise is removed and only the drift exists. The uncontrolled process diverges as a straight line.

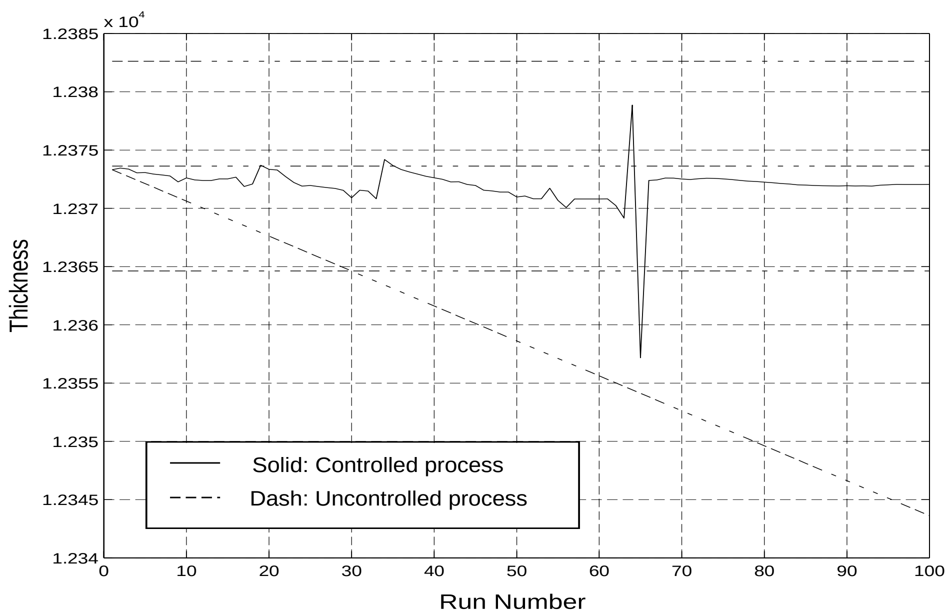

Figure 9: Photoresist process I controlled by the SVR-DHOBE controller. At this time, white noise is removed and only the drift exists. The uncontrolled process diverges as a straight line.

$$
\begin{aligned}
&+ \quad 0.023BTI - 4.34 \cdot 10^{-5} \cdot SPS \cdot SPT \\
&+ \quad 5.19 \cdot 10^{-5} \cdot SPS \cdot BTE - 1.07 \cdot 10^{-3} \\
&\times \quad SPT \cdot BTE + 5.15 \cdot 10^{-6} \cdot (SPS)^2 \\
&- \quad 4.11 \cdot 10^{-4} \cdot SPT \cdot BTI
\end{aligned} \tag{9}
$$

Where R is the reflectance in % and the other variables are defined as in previous section. The target is fixed at 39.4967%.

After variable substitution, equation (9) is changed into:

$$
\begin{aligned}
R = & \quad 134.4 - 0.046u_1 + 0.32u_2 - 0.17u_3 + 0.023u_4 \\
- & \quad 4.34 \cdot 10^{-5}u_1u_2 \\
+ & \quad 5.19 \cdot 10^{-5}u_1u_3 - 1.07 \cdot 10^{-3}u_2u_3 \\
+ & \quad 5.15 \cdot 10^{-6}u_1^2 - 4.11 \cdot 10^{-4}u_2u_4
\end{aligned} \tag{10}
$$

where

$$
\begin{aligned}
u_1 &= SPS \\
u_2 &= SPT \\
u_3 &= BTE \\
u_4 &= BTI
\end{aligned}
$$

The output of the process in each run is:

$$
y_k = R + d_1 \cdot k + v_1 \tag{11}
$$

where $d_1 = -0.3$ and v_1 is Gaussian with zero mean and variance 9.

From Figure 10 and Figure 11, it can be seen that in the second-order case, the SVR-MOVE controller performs better than the SVR-DHOBE controller. The process controlled by the SVR-DHOBE controller has some overshoots at run 52, 90 and 95.

4 Summary

RtR control methods are generalized and compared in this paper. Based on previous discussions and results, the application scope and complexity of the RtR control methods can be generalized in Table 3, where "Y" denotes "Applied", "N" denotes "Not applied", "L" denotes "Low", "H" denotes "High", and "M" means "Medium".

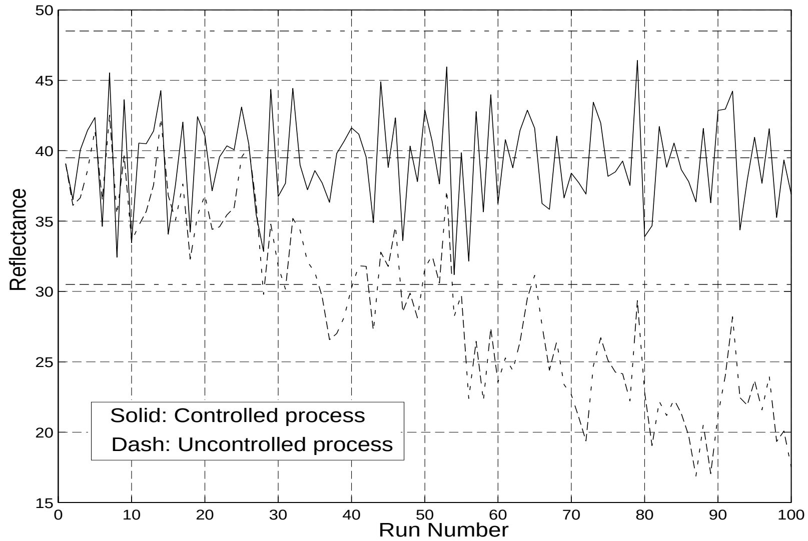

Figure 10: Photoresist process II controlled by the SVR-MOVE controller under drift.

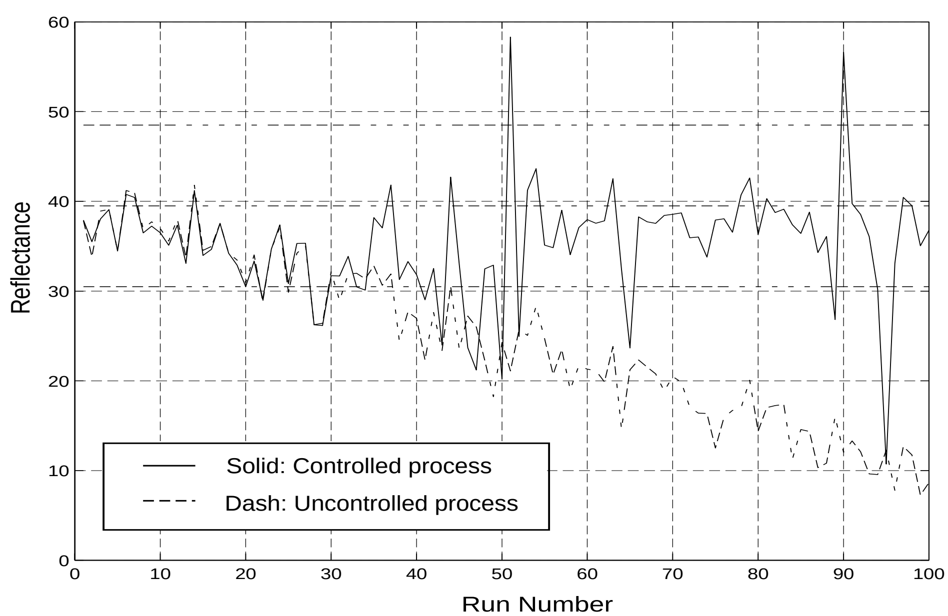

Figure 11: Photoresist process II controlled by the SVR-DHOBE controller under drift.

RtR Control Methods	Linear Process	Light Nonlinear Process	Severe Nonlinear Process	Complexity
EWMA Method	Y	Y	N	L
Machine Learning Method	Y	N	N	H
LSR Method	Y	Y	Y	M
Probability Method	Y	N	N	M
Neural Network Method	Y	Y	N	H
Set-valued Method	Y	Y	Y	M

Table 3: Generalization of RtR control methods

Preliminary simulation results show that the set-valued RtR controller with ellipsoid approximation has better or comparable performance over some other control methods. In some cases, the SVR-MOVE controller performs better than the SVR-DHOBE controller. It also shows that it is insufficient to use linear models to approximate severe nonlinear processes and it is necessary to develop nonlinear RtR controllers for semiconductor processes. More simulations will be conducted and we expect to apply the set-valued RtR controller with ellipsoid approximation to some real semiconductor processes in the near future.

References

[1] J. S. Baras, N. S. Patel, "Designing response surface model-based run-by-run controllers: A worst case approach", IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 19, 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. Semiconduct. Manufact., 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 Transactions on Semiconductor Manufacturing, Vol. 11, No. 2,pp. 285-295, 1998.

[5] M. F. Cheung, S. Yurkovich and K. M. Passino, "An optimal volume ellipsoid algorithm for parameter set estimation", Proceedings of the 30th Conference on Decision and Control, pp. 969-974, 1991.

[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 Transactions on Information Theory, vol IT-33, No. 3, p383-392, 1987.

[7] H. Deng, C. Zhang, J. S. Baras, “The set-valued run-to-run controller based on the DHOBE algorithm”, submitted.

[8] D. Dong, T. J. McAvoy, E. Zafiriou, “Batch-to-batch optimization using neural network models”, Int. Eng. Chem. Res. Vol. 35, pp. 2269-2276, 1996.

[9] E. Fogel and Y. F. Huang, “On the value of information in system identification-bounded noise case”,Automatica, vol. 18, no. 2, pp. 229-238, 1982.

[10] E. S. Hamby, P. T. Kabamba, and P. Khargonekar, “A probabilistic approach to run-to-run control,” IEEE Trans. Semiconduct. Manufact., vol. 11, no. 4, pp. 654-669, 1998.

[11] C. D. Himmel and G. S. May, “Advantages of plasma etch modeling using neural networks over statistical techniques”, IEEE Trans. Semiconduct. Manufact., vol. 6, no. 2, 1993.

[12] Y. L. Huang, etc, “Constructing a reliable neural network model for a plasma etching process using limited experimental data”, IEEE Trans. Semiconduct. Manufact., vol. 7. no. 3, 1994.

[13] A. Ingolfsson and E. Sachs, “Stability and sensitivity of an EWMA controller”, Journal of Quality Technology, vol. 25, pp. 271-287, 1993.

[14] S. Leang and C. J. Spanos, “Statistically based feedback control of photoresist application”, IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp.185-190, 1991

[15] E. Palmer, W. Ren, C. J. Spanos, ”Control of photoresist properties: A Kalman filter based approach,” IEEE Trans. Semiconduct. Manufact., vol. 9, no. 2, pp. 208-214,1996.

[16] J. A. Mullins, etc, “An evaluation of model predictive control in run to run processing in semiconductor manufacturing”, SPIE, vol. 3213, pp. 182-189, 1997.

[17] C. D. Natale, etc, “Modeling of APCVD-doped silicon dioxide deposition process by a modular neural network”, IEEE Trans. Semiconduct. Manufact., vol. 12, no. 1, 1999.

[18] Z. Ning, etc, “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.

[19] E. Sachs, A. Hu, and A. Ingolfsson, “Run by run process control: combining SPC and feedback control”, IEEE Trans. Semiconduct. Manufact., vol. 8, pp. 26-43, 1995.

[20] T. H. Smith and D. S. Boning, “Artificial neural network exponentially weighted moving average controller for semiconductor processes,” J. Vacuum Science Tech. A, vol. 15, no. 3, pp. 236-239, 1997.

[21] X. A. Wang and R. L. Mahajan, “Artificial neural network model-based run-to-run process controller”, IEEE Trans. Components, Packaging and Manu. Tech., Part C, vol. 19, no. 1, 1996.

[22] C. Zhang, H. Deng and J. S. Baras, “The set-valued run-to-run controller in semiconductor manufacturing processes”, technical report, UMCP, 1999.

[23] C. Zhang, H. Deng and J. S. Baras, “The set-valued run-to-run controller with ellipsoid approximation”, technical report, UMCP, 2000.