TECHNICAL RESEARCH REPORT

The Set-Valued Run-to-Run Controller with Ellipsoid Approximation

by Chang Zhang, John S. Baras

T.R. 2000-32

The Set-Valued Run-to-Run Controller with Ellipsoid Approximation

Chang Zhang* and John S. Baras†
Department of Electrical and Computer Engineering
University of Maryland, College Park
College Park, MD, 20742

January 8, 2000

Abstract

In order to successfully apply Run-to-Run (RtR) control or real time control in a semiconductor process, it is very important to estimate the process model. Traditional semiconductor process control methods neglect the importance of robustness due to the estimation methods they use. A new approach, namely the set-valued RtR controller with ellipsoid approximation, is proposed to estimate the process model from a completely different point of view. Because the set-valued RtR controller identifies the process model in the feasible parameter set which is insensitive to noises, the controller is robust to the environment noises. Ellipsoid approximation can significantly reduce the computation load for the set-valued method. In this paper, the Modified Optimal Volume Ellipsoid (MOVE) algorithm is used to estimate the process model in each run. Design of the corresponding controller and parameter selection of the controller are introduced. Simulation results showed that the controller is robust to environment noises and model errors.

---

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

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

1 Introduction

In semiconductor manufacturing, run-to-run (RtR) control is paid more and more attention. The RtR control method is applicable to many semiconductor processes such as the photolithography, the Chemical Mechanical Planarization (CMP), the Low Pressure Chemical Vapor Deposition (LPCVD) furnace, the sputter deposition, the ion implantation , the photoresist processes, etc.

In order to successfully apply RtR control or real time control in semiconductor process, it is very important to estimate the process model. Traditional RtR control methods neglect the important concept of robustness due to the estimation methods they use [16], [17], [11], [3], [15], [19], [14], etc. A new approach, namely the set-valued RtR controller with ellipsoid approximation, is proposed from a completely different point of view. Because of the existence of noises, we can not accurately estimate the process model. The possible estimates for the process model in the next run is a set. We call this set the feasible model parameter set. Because the set-valued RtR controller identifies the process model in the feasible parameter set which is insensitive to noises, the controller is robust to the environment noises.

The spectrum of problems that have been approached by the set-valued approach are control, image processing, speech processing, system identification, spectral estimation, etc [7]. The main difficulty in the application of the set-valued based RtR controller lies in the excessive computational time required to calculate the feasible sets and solving the optimization problem within these sets. It is hard to describe these sets with explicit formulas, because they can be very irregular.

Generally, the feasible parameter set of a process can be estimated by the set-valued approach in the following ways [18]: 1. The ellipsoidal approach. It is natural to use ellipsoids to approximate the region of indeterminacy. Because it has the following advantages: An ellipsoid is simply characterized by a vector center and a matrix; for convex regions, ellipsoids can be used to obtain a satisfactory approximation; linear transformations map ellipsoids into ellipsoids. 2. The orthotopic bounding. The feasible parameter set S is bounded with an orthotope aligned with the co-ordinate axes. S is defined by a set of 2N (N is the output parameter vector dimension) linear inequalities. Each bound can be obtained by solving a linear programming problem (e.g., the simplex method). 3. The exact bounding. Some approaches are applicable to obtain the exact description of the set S in some special situations [4], [13]. For approaches 2 and 3, they are usually very complex, and the estimate of the process model within these sets is extremely difficult to obtain. Therefore, we choose the ellipsoidal approach to approximate the feasible parameter set.

In application of the ellipsoid algorithms, the minimum volume ellipsoid that bounds the parameter set is desired. According to the difference of the search for the minimum bounding ellipsoid, there are mainly two algorithms: the Optimal Volume Ellipsoid (OVE) algorithm [6] and the Optimal Bounding Ellipsoid (OBE) algorithm [8]. The OVE algorithm was developed by M. F. Cheung, etc. It is based on Khachiyan's ellipsoid algorithm[1] developed for solving the linear programming problem. However, the OVE algorithm can not track fast changing processes. In this paper, it is modified not only to track such processes, but also to deal with various disturbances. The new algorithm is called the Modified OVE (MOVE) algorithm. The OBE algorithm was developed by Fogel and Huang as a set-membership parameter estimation algorithm [10]. An important OBE algorithm is the Dasgupta and Huang OBE (DHOBE) algorithm [8]. It differs from the previous OBE method by introducing a forgetting factor which tries to shrink the ellipsoid each time the model is updated. For application of the DHOBE algorithm in RtR control, please refer to [9].

Both the MOVE algorithm and the OBE algorithm use ellipsoids to approximate the feasible parameter sets, and 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 geometrical point of view, but the OBE algorithm uses a Recursive Least Square (RLS) type scheme to update the center of the ellipsoid. For a detailed comparison of the two ellipsoid algorithms in RtR control, please refer to [20].

The rest of the paper is organized as follows: The introduction of the structure of the set-valued RtR controller with ellipsoid approximation is given in section 2.1; the MOVE algorithm is described in section 2.2; selection of important parameters for the controller and the procedure of the controller are given in section 2.3 and section 2.4; the application of the controller is simulated on several photoresist processes in section 3; finally, conclusions and future work are given in the last section.

2 Design of the Set-valued RtR Controller with Ellipsoid Approximation

The idea of the controller is to use the ellipsoid algorithm to estimate the process model in each run. If the disturbance exceeds certain threshold, then the process model will be updated.

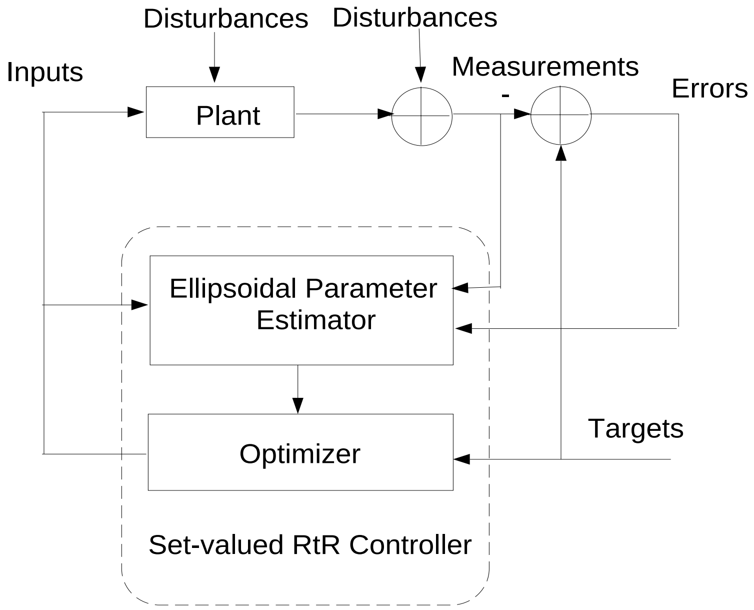

Figure 1: Structure of the set-valued RtR controller with ellipsoid approximation

2.1 Structure of the Set-valued RtR Controller with Ellipsoid Approximation

In general, the function of a RtR controller is to maintain the outputs of a process as close to targets as possible. The structure of the set-valued RtR controller with ellipsoid approximation is shown in Figure 1. In this controller, the ellipsoidal parameter estimator approximates the feasible sets with ellipsoids and estimates the process model for the next run. The optimizer then optimizes the model supplied from the ellipsoidal parameter estimator and adjusts inputs to the plant. The plant model can be linear or non-linear. There are different methods to select the estimate of the process model within the ellipsoid. For example, the estimate may be based on the worst case approach [2], where the authors seek a process model within the ellipsoid to minimize the worst-case cost. In this paper, the center of the ellipsoid is used as the estimate, since it has the following advantages:

1) Simplicity. The center of the ellipsoid is available right after each recursion of the MOVE algorithm. The worst-case approach needs to take into account of the cost function, and solve the mini-max problem in the estimation, which makes it very complex.

2) Optimality. The center of the ellipsoid is an optimal estimate in a geometric sense. The worst-case approach may be conservative compared to this approach. Comparative

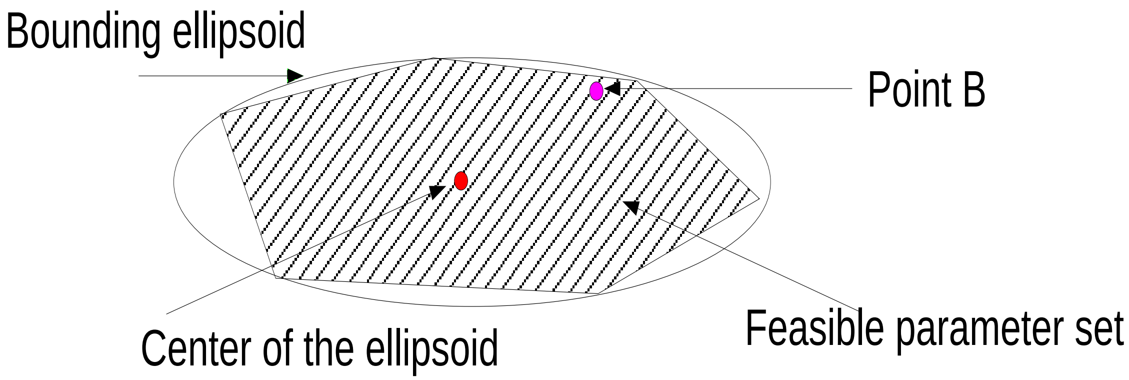

Figure 2: The center of an ellipsoid is a safe estimate of the process model

experiments show that the controller using the center as the estimate of the process model has better or comparable performance over the controller taking the worst-case approach [9].

3) Safety (Robustness). The center of the ellipsoid is a safe estimate of the process model. Especially for convex sets, the center is one of the safest points that can not easily fall out of the feasible parameter set under unknown disturbances, as shown in Figure 2. In the figure, point B is not a safe estimate, since it can go easily out of the feasible set. Since most semiconductor processes can be described by convex functions, the center of the ellipsoid is safe in general.

The ellipsoidal parameter estimator uses the MOVE algorithm to calculate the ellipsoid which bounds the feasible parameter set. It is introduced in the following section.

2.2 The MOVE Algorithm

For a linear-in-parameter system, it can be rewritten as the following form:

$$y_k = X_k^T \theta_k + \eta_k \tag{1}$$

where k is the run number, y_k is the output, X_k is the vector of inputs, θ_k is the vector of process parameters to be estimated, and η_k is the noise. For example, a process with model:

$$\begin{aligned} y_k &= c_{k,1} + c_{k,2} \cdot u_{k,1} + c_{k,3} \cdot u_{k,2} + c_{k,4} \cdot u_{k,3} \\ &\quad + c_{k,5} \cdot u_{k,1} \cdot u_{k,2} + c_6 \cdot u_{k,3}^2 + \eta_k \end{aligned} \tag{2}$$

where $u_{k,1}$, $u_{k,2}$ and $u_{k,3}$ are inputs, and $c_{k,1}$, ..., $c_{k,6}$ are the model parameters to be estimated. It can be rewritten in the form of equation (1), with $X_k = [1, u_{k,1}, u_{k,2}, u_{k,3}, u_{k,1}u_{k,2}, u_{k,3}^2]^T$, and $\theta_k = [c_{k,1}, c_{k,2}, c_{k,3}, c_{k,4}, c_{k,5}, c_{k,6}]^T$.

Suppose that the noise bound is γ, then the feasible parameter set is:

$$F_k = \{\theta_k : |y_k - X_k^T \theta_k| \leq \gamma\} \tag{3}$$

In general, this set is difficult to calculate. Then we want to find the minimum volume ellipsoid E_k, such that:

$$E_k = \min\{vol(\text{Ellipsoid } E)\} \tag{4}$$

with

$$\text{Ellipsoid } E \supset F_k \tag{5}$$

The ellipsoid algorithm then produces, at each step k, a set of estimates bounded by the ellipsoid:

$$E_k = \{\theta_k : (\theta_k - \hat{\theta}_k)^T P_k^{-1} (\theta_k - \hat{\theta}_k) \leq 1\} \tag{6}$$

Where P_k determines the volume of the ellipsoid, and $\hat{\theta}_k$ is the center of the ellipsoid at run k. Therefore, with the change of the process, there is a series of moving ellipsoids that bound the changing process parameter sets.

The MOVE algorithm can be generalized as the following steps:

Step 1. Calculate the following parameters:

$$\alpha = \frac{y_k + \gamma - X_k^T \hat{\theta}_{k-1}}{\sqrt{X_k^T P_{k-1} X_k}} \tag{7}$$

$$\beta = \frac{\gamma}{\sqrt{X_k^T P_{k-1} X_k}} \tag{8}$$

If $\alpha > 1$, then reset β to $\beta - \frac{\alpha-1}{2}$ and $\alpha = 1$. If $2\beta - \alpha > 1$, then reset β to $\frac{1+\alpha}{2}$.

Step 2. Calculate 3 intermediate variables τ, δ and σ.

(i) If $\alpha \neq \beta$, then find the real solution τ of

$$\begin{aligned}&(n+1)\tau^2 + \{\frac{(1+\alpha)(\alpha - 2\beta + 1)}{\beta - \alpha}\\&+2[n(\beta - \alpha) + 1]\}\tau + n\alpha(\alpha - 2\beta) + 1 = 0\end{aligned} \tag{9}$$

such that $\alpha - 2\beta < \tau < \alpha$. In the above equation, n is the dimension of the estimated vector $\hat{\theta}$.

$$\delta = \frac{(\tau+1)^2(\beta-\alpha) - \tau(1+\alpha)(2\beta-\alpha-1)}{\tau+\beta-\alpha} \tag{10}$$

$$\sigma = \frac{-\tau}{\beta-\alpha} \tag{11}$$

(ii) If $\alpha = \beta$, then $\tau = 0$, and

$$\delta = \frac{n}{n-1}(1-\beta^2) \tag{12}$$

$$\sigma = \frac{1-n\beta^2}{1-\beta^2} \tag{13}$$

Step 3. If $\sigma > 0$, then update the ellipsoid:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + \frac{\tau P_{k-1} X_k}{\sqrt{X_k^T P_{k-1} X_k}} \tag{14}$$

$$P_k = \delta(P_{k-1} - \sigma \frac{P_{k-1} X_k X_k^T P_{k-1}}{X_k^T P_{k-1} X_k}) \tag{15}$$

If $\sigma \leq 0$, then the ellipsoid is not updated and go to next step directly.

Step 4. Expand the ellipsoid. This is the modified part. Due to the existence of drift noise, the real size of the bounding ellipsoid will be larger than the one obtained by the OVE algorithm. Therefore, the ellipsoid should be expanded a little bit in each step. The expansion is closely related to the size of the drift noise.

$$P_k = P_k + F \tag{16}$$

Where F is usually set as follows:

$$\begin{bmatrix} F(1,1) & 0 & 0 & \cdots & 0 \\ 0 & F(2,2) & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & F(n,n) \end{bmatrix} \tag{17}$$

Without the expanding matrix, the OVE algorithm will fail in the case of drift disturbance. The value selection of the expanding matrix will be discussed in the following part.

The algorithm can be initiated with a sufficiently large E_0 containing the feasible parameter set. For example, we can let $P_0 = 10^6 \cdot I$.

The convergence of the OVE algorithm is given in [6] by using a Lyapunov value function. The MOVE algorithm does not change the center of the ellipsoid. Instead, it insures the expanded ellipsoid can cover the area that the parameter reside in the next step. Therefore, the algorithm will not return a null value and reset with proper selection of parameters.

The corresponding set-valued RtR controller using the MOVE algorithm is called the SVR-MOVE controller. Next, we introduce the parameter selection of the SVR-MOVE controller.

2.3 The Parameter Selection of the SVR-MOVE Controller

There are three important parameters in the SVR-MOVE controller:

1) The threshold for judging drifts and shifts (step disturbances). To discern the existence of a step disturbance in a process, a threshold parameter ζ which is usually equal to the 3σ bound of the process is added. Once ζ is exceeded, it means that a step disturbance occurs and the ellipsoid matrix is reset to a large value, such as $P = 10^5 \cdot I$.

2) The noise bound γ. It should be given a small value, which ensures that the volume of the feasible parameter set is small. However, it can not be too small, otherwise the bounding ellipsoid will be small, which causes the reset of the ellipsoid too often. In practice, it is found that the range [0.01,0.1] is good enough.

3) The expanding matrix F. The most important parameter of F is F(1,1), since it is directly related to the drift noise. As to the other parameter F(i,i)s, i=2, ..., n in the expanding matrix F, they should be set smaller than F(1,1), since they are related to higher order model parameters and have very strong effect on the process. A large value of F(i,i), i=2,...,n may cause large variation and even instability. There is a trade-off in choosing the value of $F(1,1)$. The larger the value of $F(1,1)$, the stronger the tracking ability; On the other hand, F(1,1) increases the size of the bounding ellipsoid, which is contradictory to the idea of finding the minimum volume ellipsoid. Therefore, an extremely large F(1,1) may cause large variation and even instability. The tradeoff is illustrated here by simulation

on the following Low Pressure Chemical Vapor Deposition (LPCVD) furnace process. The process model is:

$$\begin{aligned} R_1 = \quad & exp(20.65 + 0.29 lnP - 15189.21 T^{-1} \\ & -47.97 Q^{-1}) \end{aligned} \tag{18}$$

$$R_2 = R_1 \frac{1 - 0.0884 R_1 Q^{-1}}{1 + 0.0884 R_1 Q^{-1}} \tag{19}$$

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. R_1 and R_2 are the deposition rates in $\overset{o}{A}\ /min$ on the first and last wafer respectively. The target rates are fixed at 169.75 $\overset{o}{A}\ /min$ and 141.7 $\overset{o}{A}\ /min$ respectively.

Equation (18) can be simplified to a linear process by taking "logarithm" operation and variables substitution.

$$ln(R_1) = 20.65 + 0.29 u_1 - 15189.21 u_2 - 47.97 u_3 \tag{20}$$

where

$$\begin{aligned} u_1 &= lnP \\ u_2 &= T^{-1} \\ u_3 &= = Q^{-1} \end{aligned}$$

But Equation (19) is still nonlinear after the operation.

$$\begin{aligned} ln(R_2) &= ln(R_1) + ln(1 - 0.0884 R_1 u_3) \\ & \quad -ln(1 + 0.0884 R_1 u_3) \\ &= 20.65 + 0.29 u_1 - 15189.21 u_2 - 47.97 u_3 \\ & \quad +ln(1 - 0.0884 R_1 u_3) \\ & \quad -ln(1 + 0.0884 R_1 u_3) \end{aligned} \tag{21}$$

The outputs of the process in each run are:

$$y_k^1 = R_1 + d_1 \cdot k + v_1 \tag{22}$$

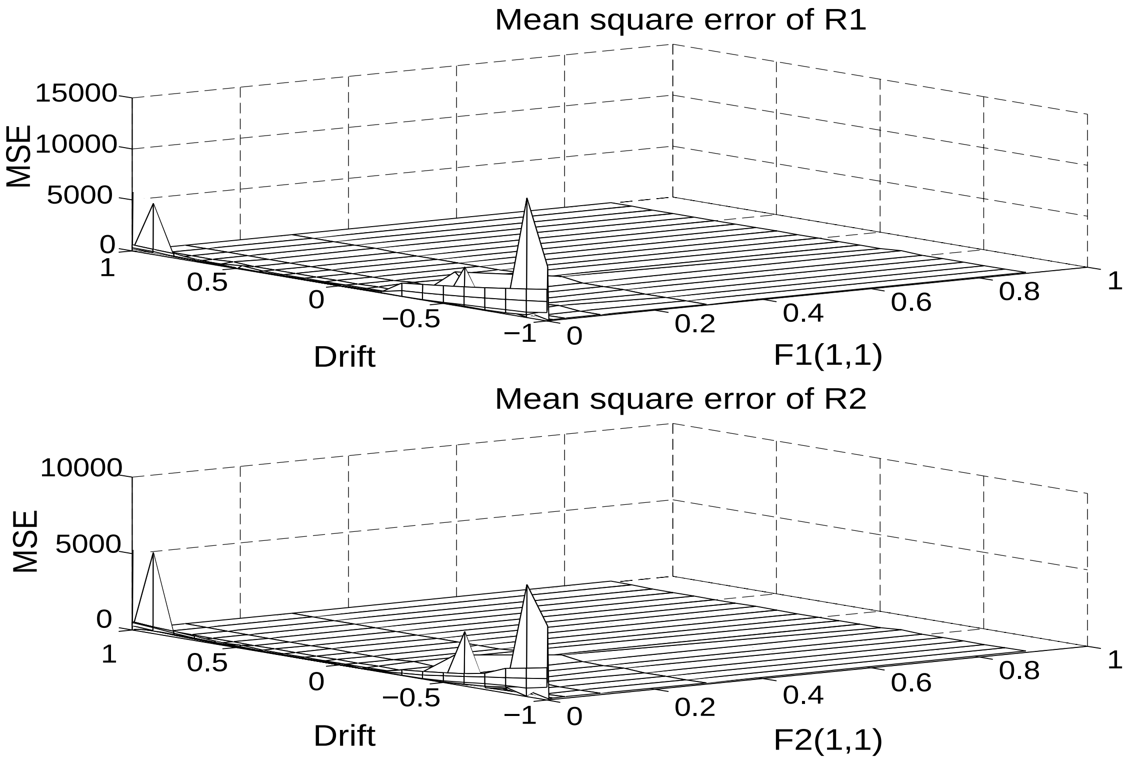

Figure 3: MSEs of the process with respect to F(1,1)s

$$y_k^2 = R_2 + d_2 \cdot k + v_2 \tag{23}$$

where y_k^1 and y_k^2 are outputs of the process at run k, d_1 and d_2 are the drifts, and v_1 and v_2 are white noises. Here $d_1 = d_2 = -0.3$.

First, let us look at the relationship between a drift and F(1,1) without other disturbances, i.e., $v_1 = v_2 = 0$. We fix $F(i,i) = 10^{-12}$, i=2, 3, 4 in this section. The simulation result is shown in Figure 3. It can be seen that when F1(1,1) and F2(1,1) are small, the Mean Square Errors (MSEs) are large; when F1(1,1) and F2(1,1) are larger than 10^{-2}, the MSEs are small and do not change much when F(1,1)s are increased. Therefore, without other noises except the drifts, a fairly large F(1,1) is preferred.

In real life, there exist various noises. Next, white noises v_1 and v_2 are added to the processes. The simulation results are shown in Figure 4 and Figure 5, where the MSEs are obtained by setting F1(1,1)=F2(1,1)=5×10^{-6} and F1(1,1)=F2(1,1)=0.05 separately. In both figures, MSEs are equal to zero when the drifts and noise variances are zero. But in Figure 4, only MSEs are equal to zero at the point where the drifts and noise variances are

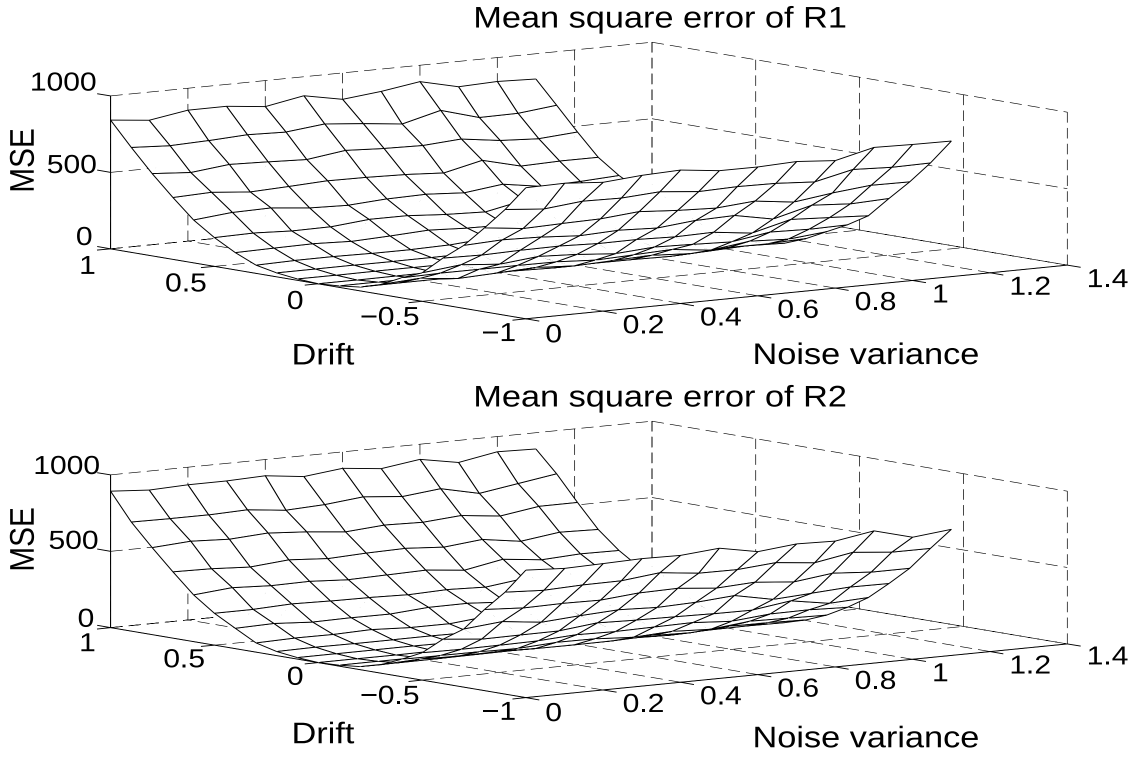

Figure 4: MSEs with respect to white noise variances and drifts when F1(1,1)=F2(1,1)=5×10^{-6}

all equal to zero, and they increase rapidly when the drifts increase. In Figure 5, when the noise variances are zero, the MSEs are approximately equal to zero all the time. It also can be seen in Figure 5, the MSEs increase rapidly with the increase of noise variances.

Therefore, there is a trade-off in the selection of F(1,1)s in the MOVE algorithm. Usually when the drifts are small, small F(1,1)s are preferred; when the drifts are large, large F(1,1)s should be chosen. Furthermore, F(i,i)s, i=2, 3, ..., n should be given smaller values than F(1,1) in general. When there exist very strong and irregular noises, F(i,i)s can be increased correspondingly to increase the control ability of the controller.

2.4 Procedure of the SVR-MOVE Controller

The working procedure of the SVR-MOVE controller can be generalized as:

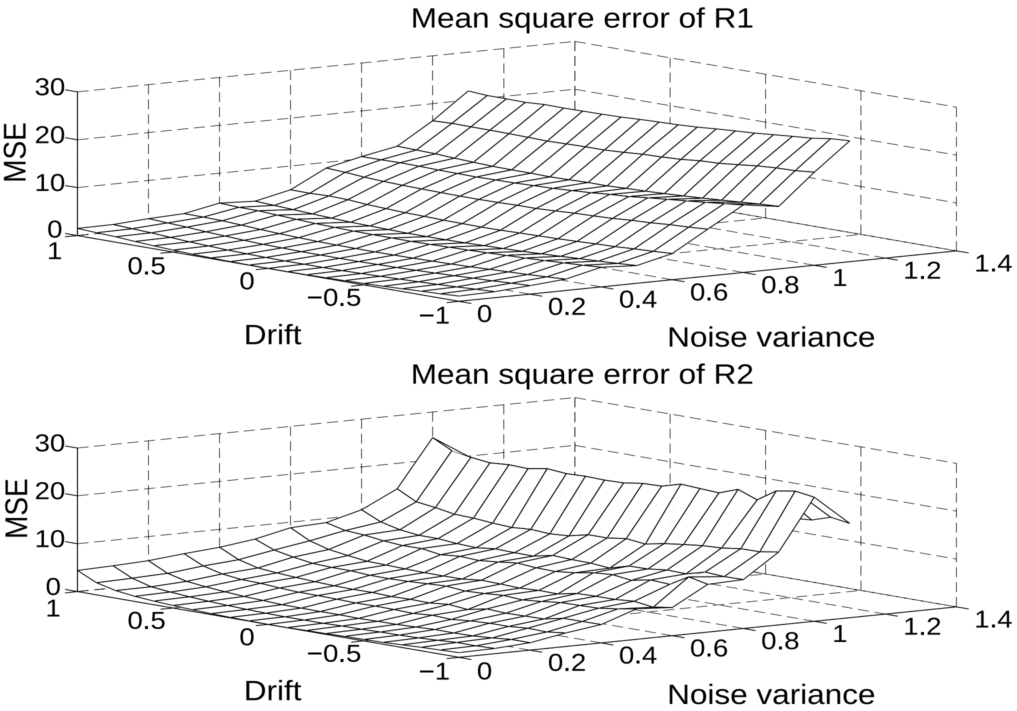

Figure 5: MSEs with respect to white noise variances and drifts when F1(1,1)=F2(1,1)=0.05

Step 1: Initialize process model, cost function and recipes. The model can be obtained from experiments.

Step 2: Setting targets and constraints for the inputs or outputs.

Step 3: Setting the controller parameters.

Step 4: Generating recipes by the process model to minimize the cost function.

Step 5: Measure outputs, update the process model if necessary. The process model is updated only if noises exceed certain threshold, which reduces the variation of the process.

Step 6: Go to step 4.

3 Application of the SVR-MOVE Controller

3.1 An Almost Linear Photoresist Process I

The following is the model used in the photoresist process I [12].

$$\begin{aligned} T = & - 13814 + \frac{2.54 \cdot 10^6}{\sqrt{SPS}} + \frac{1.95 \cdot 10^7}{BTE\sqrt{SPS}} \\ & - 3.78BTI - 0.28SPT - \frac{6.16 \cdot 10^7}{SPS} \end{aligned} \tag{24}$$

Where T is the resist thickness in Angstroms, and the target is fixed at 12373.621 Angstroms. 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:

$$\begin{aligned} 4500 < SPS < 4700 \\ 15 < SPT < 90 \end{aligned}$$

$$
\begin{aligned}
105 &< BTE < 135 \\
20 &< BTI < 100
\end{aligned}
$$

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

$$
\begin{aligned}
T = & - 13814 + 2.54 \cdot 10^6 u_1 + 1.95 \cdot 10^7 u_1 u_2 \\
& - 3.78 u_3 - 0.28 u_4 - 6.16 \cdot 10^7 {u_1}^2
\end{aligned}
\tag{25}
$$

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:

$$
y_k = T + d_1 \cdot k + v_1 \tag{26}
$$

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

The parameters for the SVR-MOVE controller are $n = 6$, $\gamma = 0.05$, $P_0 = 10^5 \cdot I$, $\hat{\theta}_0 = [-13814,$
$2.54 \cdot 10^6, 1.95 \cdot 10^7, -3.78, -0.28, -6.16 \cdot 10^7]^T$ and

$$
F = \begin{bmatrix}
20 & 0 & 0 & 0 & 0 & 0 \\
0 & 10^{-8} & 0 & 0 & 0 & 0 \\
0 & 0 & 10^{-8} & 0 & 0 & 0 \\
0 & 0 & 0 & 10^{-8} & 0 & 0 \\
0 & 0 & 0 & 0 & 10^{-8} & 0 \\
0 & 0 & 0 & 0 & 0 & 10^{-8}
\end{bmatrix}
\tag{27}
$$

The simulation result is shown in Figure 6. In this figure, the solid line is the controlled process output, which stays in the 3σ region satisfactorily. The dashed line, which diverges

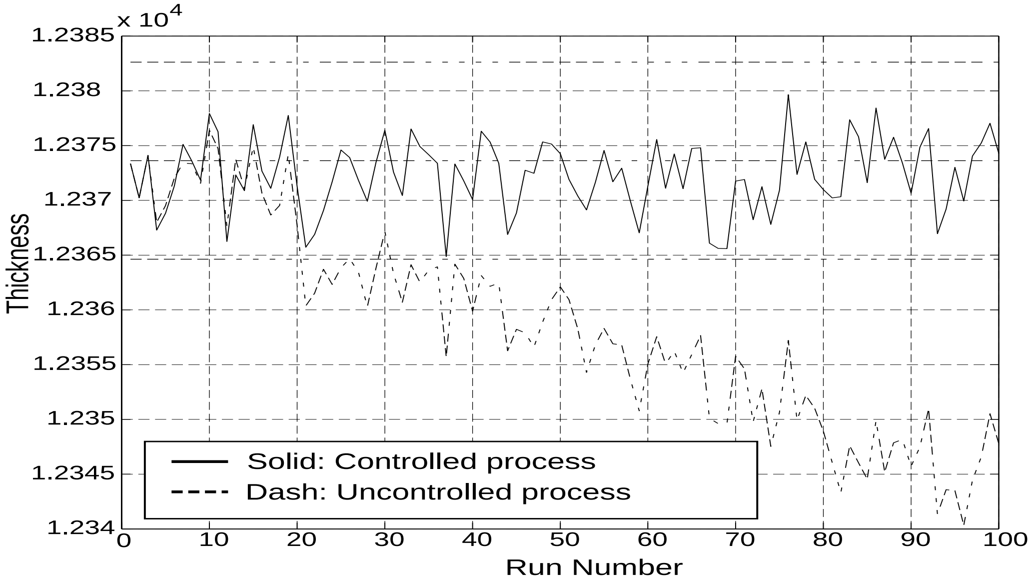

Figure 6: Photoresist process I controlled by the SVR-MOVE controller under drift

because of the drift disturbance, is the uncontrolled process. The three straight dashed lines in the figure are the $+3\sigma$, target and -3σ lines respectively.

In the next, white noise in the process is removed and only the drift exists as the disturbance. From Figure 7, it can be seen that the controlled process stays very close to the target, and the uncontrolled process diverges as a straight line. This shows the validity of the SVR-MOVE controller to deal with drifts.

3.2 A Full Second-order Nonlinear Photoresist Process II

Now the process is a full second-order nonlinear process [12].

$$\begin{aligned} R = & \quad 134.4 - 0.046SPS + 0.32SPT - 0.17BTE \\ + & \quad 0.023BTI - 4.34 \cdot 10^{-5} \cdot SPS \cdot SPT \\ + & \quad 5.19 \cdot 10^{-5} \cdot SPS \cdot BTE - 1.07 \cdot 10^{-3} \\ \times & \quad SPT \cdot BTE + 5.15 \cdot 10^{-6} \cdot (SPS)^2 \\ - & \quad 4.11 \cdot 10^{-4} \cdot SPT \cdot BTI \end{aligned} \tag{28}$$

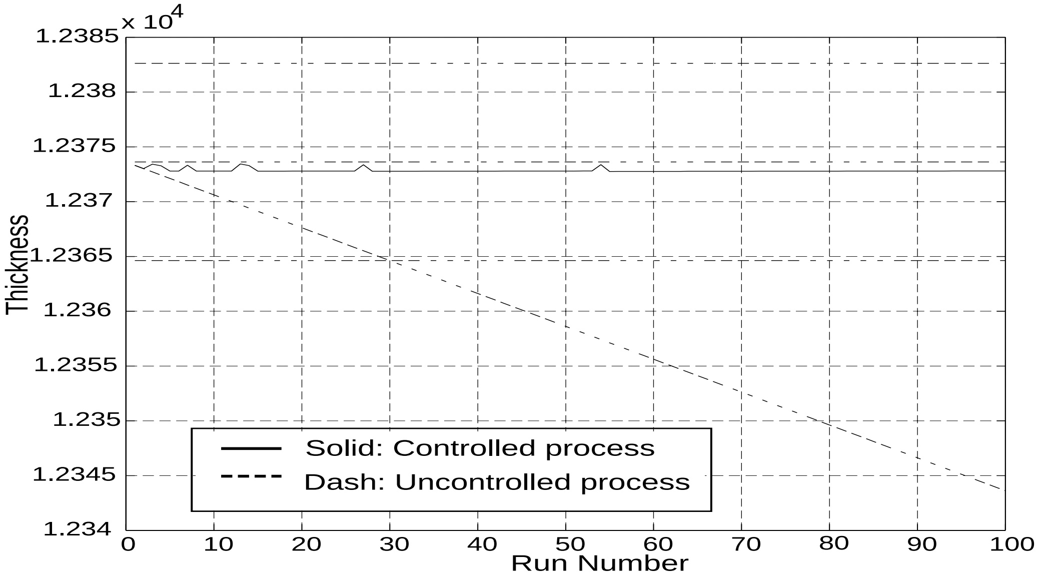

Figure 7: Photoresist process I controlled by the SVR-MOVE controller under drift without white noise

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

After variables substitution, the model is changed into:

$$\begin{aligned} R = & \quad 134.4 - 0.046u_1 + 0.32u_2 - 0.17u_3 + 0.023u_4 \\ - & \quad 4.34 \cdot 10^{-5}u_1u_2 \\ + & \quad 5.19 \cdot 10^{-5}u_1u_3 - 1.07 \cdot 10^{-3}u_2u_3 \\ + & \quad 5.15 \cdot 10^{-6}u_1^2 - 4.11 \cdot 10^{-4}u_2u_4 \end{aligned} \tag{29}$$

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:

$$y_k = R + d_1 \cdot k + v_1 \tag{30}$$

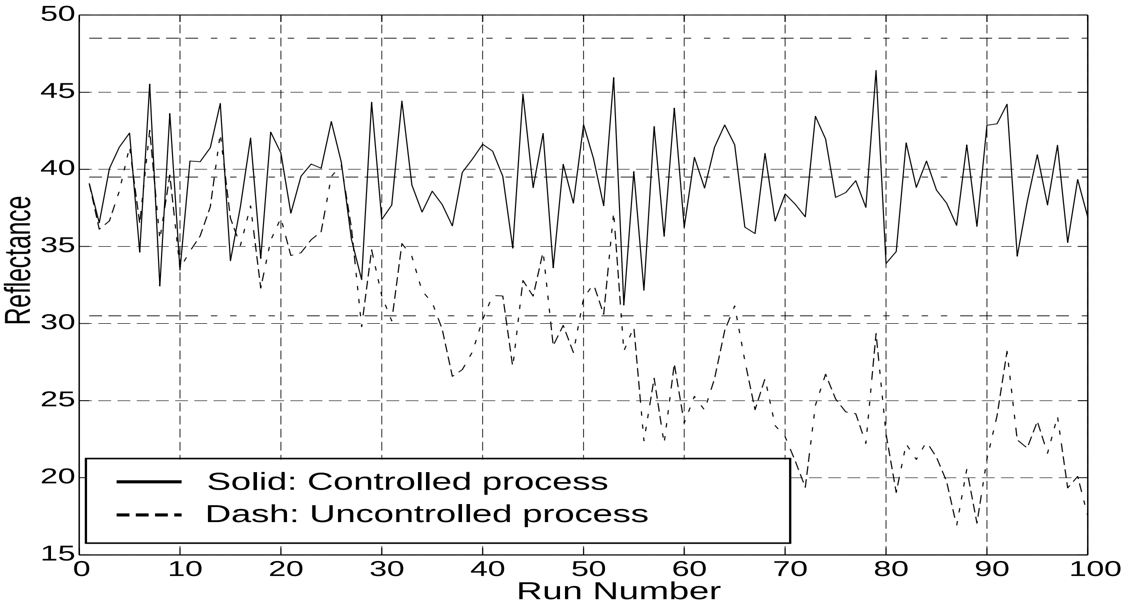

Figure 8: Photoresist process II controlled by the SVR-MOVE controller under drift

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

The parameters for the SVR-MOVE controller are $n = 10$, $\gamma = 0.05$, $P_0 = 10^5 \cdot I$, $\hat{\theta}_0 = [134,$
$-0.046, 0.32, -0.17, 0.023, -4, 34 \cdot 10^{-5}, 5.19 \cdot 10^{-5}, -1.07 \cdot 10^{-3}, 5.15 \cdot 10^{-6}, -4.11 \cdot 10^{-4}]^T$ and

$$F = \begin{bmatrix} 0.05 & 0 & 0 & \cdots & 0 \\ 0 & 10^{-6} & 0 & \cdots & 0 \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 10^{-6} \end{bmatrix}_{10} \tag{31}$$

From Figure 8, it can be seen that the controller still controls the nonlinear process well, and the uncontrolled process diverges.

3.3 Photoresist Process I with a Large Model Error and Multiple Noises

We still use the photoresist process I in section 3.1. Now assume that we do not know the exact underlying process model. It means that there is a large model error at the beginning of the process. Several different kinds of noises are added too. Now $\hat{\theta_0} = [-13600, 2.5 \cdot 10^6, 2.01 \cdot 10^7, -4.02, -0.31, -5.99 \cdot 10^7]^T$.

The output of the process in each run is:

$$y_k = T + d_1 \cdot k + v_1 + v_2 + v_3 + v_3 \cdot v_4 \tag{32}$$

where T is defined the same as in equation (25). The noises are: $d_1 = -0.3$, v_1 is Gaussian with zero mean and variance 9, v_2 is the product of two independent Gaussian variables with zero means and variance 1s, v_3 is a random variable with uniform distribution in [-1,1], and v_4 is again Gaussian with zero mean and variances 1. A large step disturbance will occur at run 30.

The parameters for the controller are: $n = 6$, $\gamma = 0.05$, $P_0 = 10^{-5} \cdot I$ and

$$F = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.05 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.05 \end{bmatrix} \tag{33}$$

The simulation result is shown in Figure 9. It can be seen that the process returns to the target right after the process begins; when the large step disturbance occurs at run 30, the process has a large disturbance too, but it returns to the target again immediately. This shows the capability of the scheme to deal with large model errors, large disturbance and multiple noises.

4 Summary

The set-valued RtR controller with ellipsoid approximation gives a good and safe estimate of the process model in a minimum volume ellipsoid. It is easily applicable to various

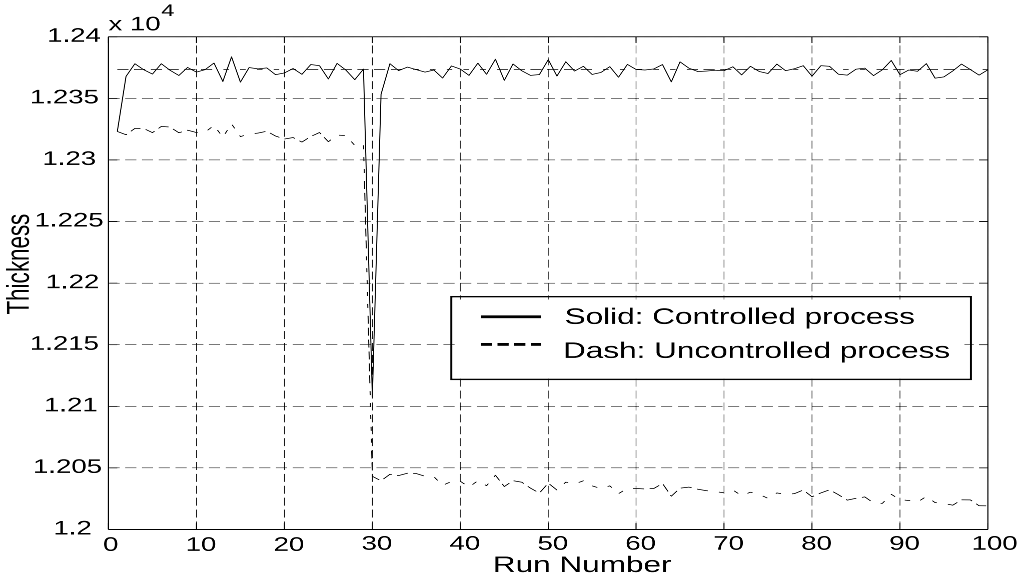

Figure 9: Photoresist process I with large model error and various disturbances controlled by the SVR-MOVE controller

semiconductor manufacturing processes. The MOVE algorithm can track fast changing processes and deals with various disturbances. The SVR-MOVE controller is much more robust to disturbances and model errors than regular RtR controllers. For comparisons of the SVR-MOVE controller with some other typical RtR controllers, please refer to [20].

In the selection of parameters for the SVR-MOVE controller, further theoretical analysis is still needed. For those processes with dynamic nature that can not be expressed in a polynomial form, a much more general scheme is still not available. Our ultimate goal is to develop a general set-valued controller for a larger set of industrial processes.

References

[1] B. Aspvall and R. E. Stone, "Khachiyan's linear programming algorithm", Journal of Algorithms, vol. 1, pp. 1-13, 1980.

[2] J. S. Baras, N. S. Patel, "Designing response surface mdoel-based run-by-run Controllers: A worst case approach", IEEE Transactions on Components, Packaging and Manufacturing Technology, vol. 19, 1996.

[3] 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.

[4] V. Broman and M. J. Shensa, “Polytopes, a novel approach to tracking”, Proceedings of the 25th IEEE Conference on Decision and Control, Athens, pp. 1749-1752, 1986.

[5] 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.

[6] 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.

[7] P. L. Combettes, “The foundations of set theoretic estimation”, Proceedings of the IEEE, vol. 81, no. 2, pp. 182-208, 1993.

[8] 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.

[9] H. Deng, C. Zhang and J. S. Baras, “The set-valued run-to-run controller based on the DHOBE algorithm”, submitted to Automatica, 2000.

[10] 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.

[11] 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.

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

[13] S. H. Mo and J. P. Norton, “Fast and robust algorithm to compute exact polytope parameter bounds”, Math. Comput. Simulation 32, pp. 481-493, 1990.

[14] 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.

[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] 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.

[17] 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.

[18] E. Walter, etc, “Estimation of parameter bounds from bounded-error data: A survey”, Mathematics and Computers in Simulation 32, pp. 449-468, 1990.

[19] 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.

[20] C. Zhang, H. Deng and J. S. Baras, “Comparison of run-to-run control methods in semiconductor manufacturing”, to appear in the Proceedings of AEC/APC 2000, 2000.