

Copyright
by
Ninad Narendra Patwardhan
2010

**The Thesis Committee for Ninad Narendra Patwardhan
Certifies that this is the approved version of the following thesis:**

**Evaluation and Extension of Threaded Control for High-mix
Semiconductor Manufacturing**

**APPROVED BY
SUPERVISING COMMITTEE:**

Supervisor:

Thomas F. Edgar

Co-Supervisor:

Robert Flake

**Evaluation and Extension of Threaded Control for High-mix
Semiconductor Manufacturing**

by

Ninad Narendra Patwardhan, B.E.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

December 2010

Dedicated to my family and friends.

Acknowledgements

I would like to thank Dr. Edgar for giving me the opportunity to work on this project and for his valuable guidance, encouragement and support during my time in his research group. It was a wonderful experience working with him.

This thesis wouldn't have been complete without the support of Dr. Christopher Bode and Broc Stirton from Global Foundries Inc. I would like to thank them for their valuable time and support. Their expert advice kept me right on track during the project. Their comments were very helpful in understanding the real wafer manufacturing challenges and their severity.

I would also like to thank Dr. Flake from ECE Department for supervising this project and for his encouragement and advice all through my Master's.

I want to thank all my research group members Gill, Ela, Doug, Anh, Jong, Kody, Ivan, Kriti, Ben, Ramiro and Xiao. They have been a great source of knowledge. I learnt a lot of things from them through the control group seminars in our research group. I would also like to thank ex-Edgar group members Amogh Prabhu, Jin Wang and Hyung Lee for helping me with my questions.

I would like to thank my lovely wife, Ela and my mom, Nayana for standing behind me during difficult times in my life. Thanks to my brother Neeraj and my dear friends Shraddha, Aniket, Radhika, KMK, Shivaji and Vinay for keeping me enthusiastic and motivated for being able to complete this work. Thanks to all my friends, Saurabh, Pranav, Rutuparna, Shahid, Shruti, Jidnyasa, Amrut and Ashwini for making my Master's an enjoyable journey.

December 2010

Abstract

Evaluation and Extension of Threaded Control for High Mix Semiconductor Manufacturing

Ninad Narendra Patwardhan, M.S.E.

The University of Texas at Austin, 2010

Supervisor: Thomas F. Edgar

Co-Supervisor: Robert H. Flake

In the recent years threaded run-to-run (RtR) control algorithms have experienced drawbacks under certain circumstances, one such trait is when applied to high-mix of products such as in Application Specific Integrated Circuits (ASIC) foundries. The variations in the process are a function of the product being manufactured as well as the tool being used. The presence of semiconductor layers increases the number of times the lithography process must be repeated. Successive layers having different patterns must be exposed using different reticles/masks in order to maximize tool utilizations.

The objectives of this research are to develop a set of methodologies for evaluation and extension of threaded control applied to overlay. This project defines

methods to quantify the efficacy of threaded controls, finds the drawbacks of threaded control under production of high mix of semiconductors and suggests extensions and alternatives to improve threaded control.

To evaluate the performance of threaded control, extensive simulations were performed in MATLAB. The effects of noise, disturbances, sampling and delays on the control and estimation performance of threaded controller were studied through these simulations. Based on the results obtained, several ideas to extend threaded control by reducing overall number of threads, by improving thread definitions and combinations have been introduced. A unique idea of sampling the measurements dynamically based on the estimation accuracy is also presented. Future work includes implementing the extensions to threaded control suggested in this work in real production data and comparing the results without the use of those methods. Future work also includes building new alternatives to threaded control.

Table of Contents

List of Tables	x
List of Figures	xi
Chapter 1	1
Introduction.....	1
1.1 Semiconductor Manufacturing processes	2
1.2 Critical parameters for lithography control.....	6
1.3 Run-to-Run Control	8
1.4 Exponentially Weighted Moving Average (EWMA) Control.....	9
1.5 Threaded Control	12
1.6 Research Objectives and Overview of Thesis	13
CHAPTER 2	14
Evaluation of Threaded Control.....	14
2.1 Simulation environment.....	15
2.2 Tool Dedication	17
2.3 Moving Window Approach	19
2.4 Sampling and Scheduling	20

2.5	Overview of the simulations to be performed.....	20
2.6	Performance Simulations and Results	21
2.7	Sampling Simulations	28
2.8	Effect of Metrology Lag and Out of Order Metrology	31
2.9	Summary	32
CHAPTER 3		33
	Drawbacks, Extensions and Alternatives to Threaded Control	33
3.1	Drawbacks of Threaded Control.....	33
3.2	Possible Extensions to Threaded Control	35
3.3	Alternatives to threaded control.....	44
3.4	Summary	45
CHAPTER 4		47
	Conclusions and Future Work	47
	References	50
	Vita	54

List of Tables

Table 1.1: Wafer Processing, step by step procedure	5
Table 2.1: Control Threads formed from combinations of Tool and Product	16
Table 3.1: Effect of Tolerance on the total number of threads	38
Table 3.2: Effect of thread combinations on the estimation accuracy at filter weight=0.3	38

List of Figures

Figure 1.1: Backend Processing, Metal connections	6
Figure 1.2: Block diagram for EWMA controller applied to Overlay Lithography Process.	11
Figure 2.1: Product-Tool combinations emulating the high-mix manufacturing environment	16
Figure 2.2: Simulated Data for 5000 runs.....	17
Figure 2.3: Moving Window with a size of 10 runs	19
Figure 2.4: EWMA filtering of Thread 8, effect of small filter weight $\lambda = 0.1$	21
Figure 2.5: EWMA filtering of Thread 8, effect of large filter weight $\lambda = 0.8$	22
Figure 2.6: Effect of increasing filter weight on the MSE's in the estimated states. Filter weight (λ) is varied from 0.1 in the top left plot to 0.9 in the bottom right plot.	23
Figure 2.7: Effect of filter weight on the MSE's in the estimated states.	24
Figure 2.8: Effect of filter weight on the MSE's in the estimated states.	25
Figure 2.9: Effect of change in the step size on the MSE for Thread 3.....	25
Figure 2.10: Effect of several steps of small magnitude on the MSE.....	26
Figure 2.11: Simulation for drifting thread state	27
Figure 2.12: Comparison of MSE vs. filter weights for thread states with and without any drifts	27
Figure 2.14: Effect of Random Sampling on the MSE for Thread1	29
Figure 2.15: Effect of Uniform Sampling on the performance.....	30
Figure 3.1: Actual states for simulated data divided into 9 threads.....	36

Figure 3.2: MSE against the filter weights for individual and combined threads	39
Figure 3.3: Number of Samples vs. the Tolerance.....	41
Figure 3.4: MSE vs. Tolerance	42
Figure 3.5: MSE vs. Number of Samples	43

Chapter 1

Introduction

Semiconductor manufacturing is a rapidly progressing industry both technologically and commercially. The technology used for manufacturing semiconductors is changing every day and the competition in the industry has driven the minimum feature size that can be printed down to 32 nm. The size of the silicon wafer that can be manufactured has gone up to 12 inches in diameter and about 1.17 billion transistors are manufactured on a single chip. Printing several hundreds of chips on a single wafer has enabled mass production in this field and hence the manufacturing cost of the chips has gone down. The number of transistors that can be integrated on a chip inexpensively is said to follow Moore's Law [1], which states that the number of transistors on a computer chip doubles every two years. This trend has been true for about 45 years till now and it is believed that it will continue the same way in the coming 5 years.

The manufacturing technology has made this transition in semiconductor manufacturing possible. Today the semiconductor manufacturing process takes as many as 200 steps with several layers and interconnects. Since there are hundreds of chips on a single wafer, the cost of a processed wafer is huge. Each manufacturing step is important and needs accurate manufacturing technology and measurements. The following section describes how the silicon crystal is grown into ingots, how the wafers are formed and how the wafer is processed to form several hundreds of chip on a single wafer. Section

1.1 describes the steps used in the critical processes in semiconductor manufacturing industry. Table 1 demonstrates the masking process which transfers the pattern from reticle on to the wafer.

1.1 SEMICONDUCTOR MANUFACTURING PROCESSES

The chip manufacturing process can be summarized as follows:

- 1) Chip design-Engineers design circuit needed and how it should work. The whole process of printing this circuit on the chip depends on the design of the chip. Several masks/reticles are designed and manufactured to make the required patterns on the wafer.
- 2) Fabrication: A sequence of multiple lithographic and chemical processes is used to transfer the pattern from the masks to the chip.
- 3) Cutting the wafer into small chips: Hundreds of chips are printed on each wafer. Each wafer is cut to separate each chip from the wafer without damaging the wafer and avoiding cracks and mechanical wear. Several samples are marked for testing and the defective chips are separated.
- 4) Backend Processing: The chip is attached to a body to add to its mechanical strength since silicon is brittle. The connections are made using aluminum or copper.
- 5) Testing: The chips are tested mechanically and electrically. The working of the circuit is compared to what was designed.

The manufacture of a chip is really the fabrication and the backend processing, which are elaborated in more detail in this section.

1.1.1 Si Crystal Growth

Silica is available in abundance in beach sand which represents about 28% of the earth's crust. The silicon is obtained from silica using a special extraction process. First step is to convert the quartzite (SiO_2) into metal grade silicon or MGS. The silicon is purified as well as converted into crystalline form. The MGS is converted to Electronic Grade Silicon or EGS which is a several step process. Czochralski (CZ) or Float-Zone (FZ) method is used to form pure crystalline silicon in the form a boule [2]. The amount of dopant placed in the crucible during the CZ or FZ methods determine the doping concentration in the silicon crystal and the pulling rate affects the size of the ingot that is produced. After the crystal is grown, individual wafers are formed with the help of a sequence of mechanical operations. First the grown crystal is shaped to a uniform diameter. Mechanical lapping removes the saw damage from the wafer surfaces and helps in making the surface smooth. At this stage, the wafer is processed through chemical etching followed by chemical mechanical polishing (CMP). In CMP, wafers are polished under a pressure of 20psi. Wafers are rotated in the polishing machine in slurry of suspended SiO_2 particles in an aqueous solution of NaOH. The SiO_2 particles abrade the oxide away. The polished clean wafer in the form of a shining silver disc is ready for the next processing step which is Lithography.

1.1.2 Lithography

Lithography is the process of transferring the pattern from the mask to the wafer. It is a sequence of photographic and chemical processes. The wafers are first covered with a SiO_2 layer by exposing to extreme heat and light. Next the wafer is covered with a

uniform chemical layer called a photo-resist. Ultraviolet light is shined on the wafer through the mask. Only parts of the wafer where we want the circuit to be integrated are exposed. The resist that reacts with the ultra-violet light becomes soluble. The chip is baked in an oven to harden the resist on the unexposed wafer. Then the wafer is immersed in a solvent. The soft part of the resist exposed to the light gets dissolved and hence the desired pattern is transferred to the wafer. Next the wafers go into the oxidation chamber followed by diffusion chamber where the p and n type wells are formed depending on the design of the chips. The layering and masking steps are repeated following the layout of the next mask.

Alignment of the layers is a key challenge in photolithography. Inaccurate alignment leads to inconsistencies in the thickness of the lines which might result in short circuit and destroy the chip. There are several methods and precautions to get the correct alignment within layers. Once all the layers are transferred to the wafer as per the design, the chips are ready for back end processing. Table 1.1 below shows the summary of wafer processing with single step of lithography.


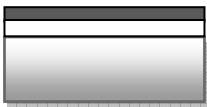
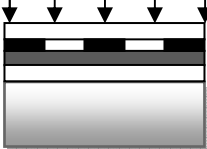
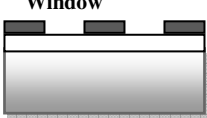

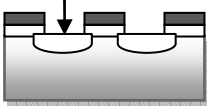

Wafer Processing Procedure (Cross section View)		
1	<p>Oxide Layer</p>  <p>Si</p>	Silicon substrate is covered with a uniform oxide layer.
2	<p>Oxide Layer</p>  <p>Resist layer</p>	Thin uniform layer of photo-resist is formed on the top surface of the chip. Generally spin coating is used to do this. The wafer is pre-baked for exposure.
3	 <p>Mask</p> <p>UV Light</p> <p>Resist</p>	A mask is placed on top of the photo-resist and the chip is exposed to UV light to transfer the pattern from the mask to the wafer.
4	 <p>Window</p> <p>Resist</p>	The exposed part of the resist reacts with UV light and becomes soluble. The wafer is then immersed in a solvent, creating windows as shown in the diagram. The wafers are post baked before the next processing step to harden the resist in the unexposed regions.
5	<p>Oxide Layer</p>  <p>Si</p>	Plasma/wet etching removes the oxide layer from the window area, which exposes the bare Si- substrate.
6	<p>Doped regions (p/n-type wells)</p> 	By diffusion/ion-implantation process dopants are introduced into the silicon substrate to form p-type or n-type wells depending on the design of the circuit being printed and the substrate doping.
7	<p>Oxide Layer</p>  <p>Si</p>	After the diffusion the photo-resist is etched off again using plasma/wet etch.

Table 1.1: Wafer Processing, step by step procedure

1.1.3 Backend processing

The layers printed on the wafer are connected as per the design of the circuit. Aluminum or copper are the popular metals used for the layer interconnects. Chemical vapor deposition (CVD) or physical vapor deposition (PVD) techniques are used for depositing the metal on the wafers.

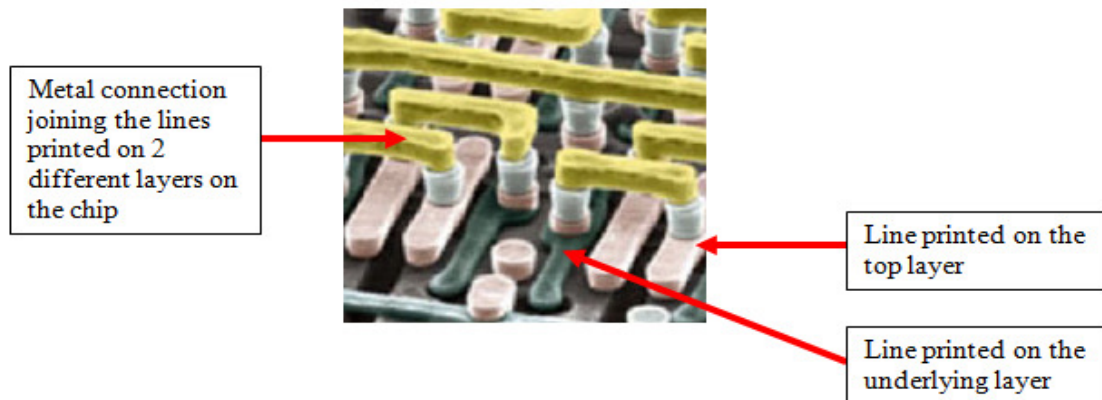


Figure 1.1: Backend Processing, Metal connections (source: www.hardwaresecrets.com)

1.2 CRITICAL PARAMETERS FOR LITHOGRAPHY CONTROL

Two of the most important aspects of lithography sequence are size and position of the photo-resist pattern with respect to the substrate pattern.

1.2.1 Critical Dimension (CD)

Measure of width of a particular feature within a given pattern is referred to as critical dimension or CD. Accuracy in the critical dimension (CD) after lithography is required at a number of steps such as Shallow Trench Isolation (STI), gate etch and

interconnect damascene patterning. This provides tighter control of the electrical properties of the transistors. The CD is known to be a function of the exposure dose and focus. The depth of focus is generally flat in the given CD resolution. Hence, CD can be controlled by manipulating the exposure dose at every step. CD is measured using either scanning electron microscopy and CD controllers are specially developed to control the process and keep the CD variability within the specification limits [3], [4].

1.2.2 Overlay

The position of the resist pattern relative to the underlying layers is known as overlay. Few of the common sources of overlay error are masks errors, lens distortion, magnification, wafer distortion, displacement of wafer alignment in translation and rotation, and overlay metrology error [5]. The various overlay error parameters are either controlled separately or combining them into a linear model. Overlay lithography control is of primary interest for this thesis; thus overlay errors are discussed in more detail.

The metrology of overlay typically employs overlay targets resident within the device pattern to measure the relative position of two adjacent layers. First a set of sites across a single wafer and among several wafers in a single lots are selected for metrology. Least-squares minimization techniques are used to fit an overlay model to these site measurements, the result of which is a set of lot-average overlay error parameters. As described by Bode et al. [5] the overlay errors can be categorized as intra-field and inter-field errors. Intra-field are the errors that vary over the reticle field while inter-field errors refer to the positioning errors that vary across the wafer.

The following example for step-and-repeat overlay models employed in the semiconductor industry was discussed by Bode et al. [5].

$$O_{X,x} = T_X + E_X X + R_X Y + m_X x - r_x y + \rho_{X,x} \quad \dots(1)$$

$$O_{Y,y} = T_Y + E_Y Y + R_Y X + m_y y - r_y x + \rho_{Y,y} \quad \dots(2)$$

The two component overlay errors $O_{X,x}$ and $O_{Y,y}$ in the equations above are a combination of both inter-field and intra-field errors. Inter-field error is composed of the mean wafer translations T_X and T_Y , wafer scaling errors E_X and E_Y and wafer rotation errors R_X and R_Y whereas intra-field error include the magnification parameters m_x and m_y and reticle rotation r_x and r_y . The overlay model is fitted to each set of overlay measurements by minimizing the residual terms of the equations, $\rho_{X,x}$ and $\rho_{Y,y}$. Stepper systems generally include corrections for the set of parameters included in the models shown above.

1.3 RUN-TO-RUN CONTROL

In semiconductor manufacturing, the products must be removed from the chamber and transferred to the measurement tool by a transport system for accurate measurements of the parameters of interest. In-situ measurement is difficult to employ in most of the semiconductor processes. Thus, run-to-run control was employed in semiconductor manufacturing and was observed to be an effective technique. Run-to-Run (RtR) control is a type of algorithm seeking to minimize the variance and deviation from set point by updating the recipe on a lot by lot or run by run basis [6], [7]. Run-to-run control compensates for the error term in the process at each run.

Miller[8] identified the problem of multi-product, multi-process manufacturing and he proposed use of four different RtR control strategies.

Run-to-Run control is particularly useful for compensation of processes with a drifting controlled variable. RtR control can return the process to target even after a step disturbance. As RtR controller became more popular in the semiconductor manufacturing, it became more apparent that some of its unique characteristics needed enhanced algorithm development. One such trait was the high-mix of products manufactured in a single factory, such as application specific integrated circuit (ASIC) fabs/foundries. With the advancement in technology, new products go into production while the old ones are phased out. The mix of the products thus keeps on changing over time, which sometime leads to major difficulties in design and deployment of RtR controller. The cost of processing equipments is really high in semiconductor industry as compared to other manufacturing industries which demands less idle time as possible and hence rules out the chance of tool dedication and tool matching for specific products. Therefore, one lot of specific products can take several processing paths through the fab than the next lot of the same product, which causes variance in the product quality.

1.4 EXPONENTIALLY WEIGHTED MOVING AVERAGE (EWMA) CONTROL

EWMA algorithm is one of the most popular RtR control algorithms and has been used extensively in different semiconductor manufacturing processes [9], [10]. The EWMA controller assumes a drifting process where the variation in the process can be modeled as an integrated moving average process. EWMA filter is recursive by its nature

and it weights the data in exponentially reducing magnitudes. It has a single tuning factor, which is called as filter weight (denoted as λ).

Similar models have been assumed for simulations of run-to-run control by Firth [11] and Prabhu [12]. The equations are referenced from these articles unless otherwise specified.

Assuming the process model,

$$y_k = bu_k + \hat{e}_k \quad \dots (3)$$

Where,

$k = 0, 1, 2, 3 \dots$ runs

y_k = model output for the k^{th} run

b = process gain

u_k = input at k^{th} run

\hat{e}_k = disturbance at the k^{th} run

The observer updates the disturbance using the EWMA equation,

$$\hat{e}_{k+1} = \lambda e_k + (1 - \lambda) \hat{e}_k \quad \dots (4)$$

Tuning parameter (filter weight) $0 \leq \lambda \leq 1$

\hat{e}_{k+1} = disturbance estimated at the $(k+1)^{\text{th}}$ run (New state being estimated)

e_k = measured disturbance at k^{th} run (old observation)

\hat{e}_k = estimated disturbance at the k^{th} run (old observation)

The weighting factor λ is used to tune the filter to obtain optimum result by assigning more or less weighting to the new value as compared to the old value. The simulation results for optimal tuning of threaded EWMA filter will be explained in detail in the following chapters.

The control law used to determine the manipulated variable input is inversion of the process model,

$$\hat{u}_{k+1} = \frac{SP - \hat{e}_{k+1}}{b} \quad \dots (5)$$

Where,

\hat{u}_{k+1} = estimate of the input for the next run

SP = set point for the output

\hat{y}_{k+1} = estimate of the output at the next run obtained from the model.

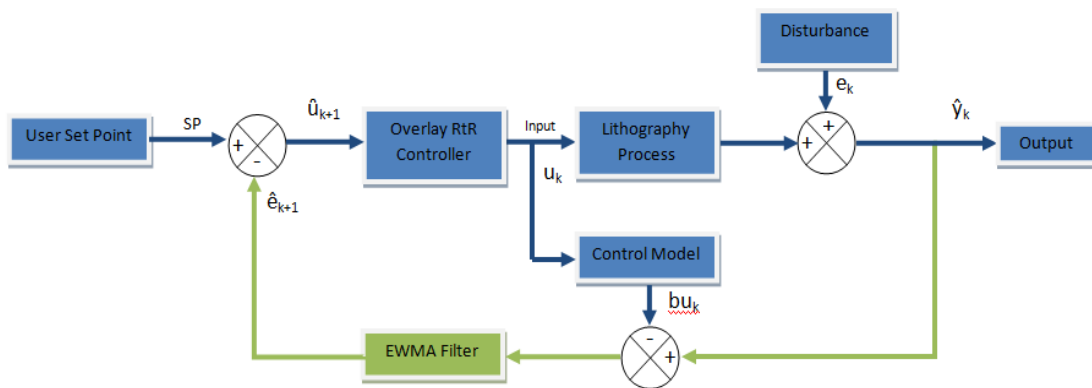


Figure 1.2: Block diagram for EWMA controller applied to Overlay Lithography Process.

Figure 1.2 demonstrates a typical EWMA RtR controller used for lithography overlay control. Equations (3), (4) and (5) describe the functioning of the above block diagram. It

can be seen that the system follows the IMC (internal model control) structure and is a purely integral controller in nature. From the block diagram, the transfer function for the overlay RtR controller can be derived for process and model as shown in [14], [15], [13].

1.5 THREADED CONTROL

In threaded control, lots having roughly the same incoming process state are grouped together and are called control threads or streamlines (Bode et al. [16]). Each of the groups is segregated from the rest of the groups based upon the criteria determining the incoming state. The threads are generally defined based on certain contexts like the product being made, the tool used for the processing and the reticle used can affect the state of the tool or the incoming state of a particular lot. Thus threaded control lumps the effects of several contexts into a disturbance and hence eliminates the need for estimating the effect of disturbance due to each of the contexts individually.

A simple example can be considered here to explain the threaded control approach more clearly. Suppose there are two tools on which two different products are being manufactured. The contexts of tools and products are identified as the factors affecting the variability in the process based on comparative study. Thus, based on the contexts there will be four combinations of tool and product hence four threads.

Although threaded control is a popular and extensively used, threaded control has certain disadvantages when applied to a high mix of semiconductor products. The disadvantages of threaded control are discussed in detail in Chapter 3.

1.6 RESEARCH OBJECTIVES AND OVERVIEW OF THESIS

The Objectives of this research are to develop a set of methodologies for evaluation and extension of threaded control applied to overlay control. This project defines methods to quantify the efficacy of threaded controls, finds the drawbacks of threaded control under production of high mix of semiconductors and suggests extensions / alternatives to improve threaded control.

The sections discussed earlier 1.1-1.5 gave a brief background of the project. Chapter 2 discusses the evaluation of threaded control. The evaluation consists of knowing the control performance which is an indicator of the accuracy of state estimates, automation performance which indicates how often the states can be estimated robustly as well as the visibility of the estimates. Simulation experiments were carried out for proving the theoretical concept and the background for running the simulations will be presented in the next chapter. Chapter 2 discusses the results obtained from the performance simulations.

Chapter 3 discusses and summarizes the drawbacks of threaded control in detail and shows the need for extension of the threaded control applied to high mix environment. Chapter 3 also presents the possible extensions to threaded control by overcoming few of the drawbacks discussed in the previous chapter. It also discusses novel non-threaded control strategies developed in the recent years and demonstrates how they are better or worse than the threaded RtR control. This chapter also shows a few simulation results for the extensions/alternatives developed. Chapter 4 summarizes the results obtained and the key contributions of the research done and suggests a direction for the future work.

CHAPTER 2

Evaluation of Threaded Control

RtR and threaded control techniques introduced in Chapter 1 have been developed and employed for several years now. In all these years threaded run to run control has improved significantly in many aspects but the fundamental technique of employing threaded control is unchanged. Bode [3] , Miller et al.[8], Firth [17] and Braun et al.[18] have discussed several drawbacks of threaded control which will be discussed later. Firth et al.[11] developed a non-threaded algorithm to address the issues raised with high-mix control. The algorithm was based on certain assumptions and simulations were performed using the data generated from simulations for comparison of performance with the threaded control technique. Just in time adaptive estimation (JADE) was compared with EWMA technique when applied to step and ramp disturbances, noise and delays. JADE performed better in the state estimation with increasing magnitudes of step and ramp disturbances on the simulated data. Prabhu et al.[12] developed a new method for state estimation based on random walk model. This model combined with a moving window approach and least squares solution provided better estimates with high-mix environment and with low runner threads. The performance of random walk model was compared with JADE and EWMA control methods using simulated data. The results from the simulations show that the method shows lowest estimation error for simulated processes as compared to JADE and threading. Wang et al.[19] developed a framework based on the best linear unbiased estimate (BLUE) to study the similarities and differences in the

different non threaded estimation techniques based on recursive least squares, JADE and Kalman filter. It was observed that the RLS is the same as Kalman filter assuming output variance as 1. Performance of RLS was almost same to that with Kalman filter. It was observed that because JADE resets the estimated covariance in every run, it loses the statistical properties of the process contained in the historical data, this result in the loss of estimation performance. However, it was also observed that the performance of JADE was improved by applying higher weightings on the previously estimated states for a stationary process. Thus, the need to evaluate the performance of threaded control technique with noise, disturbances and delays was observed. This chapter discusses about evaluation of threaded control in terms of control performance and automation performance and several other concerns.

2.1 SIMULATION ENVIRONMENT

To quantify the performance of the EWMA controller simulation environment was setup to emulate the high mix manufacturing. The example consisted of three products being manufactured on three tools, which forms nine combinations of tool and product or nine control threads/streamlines.

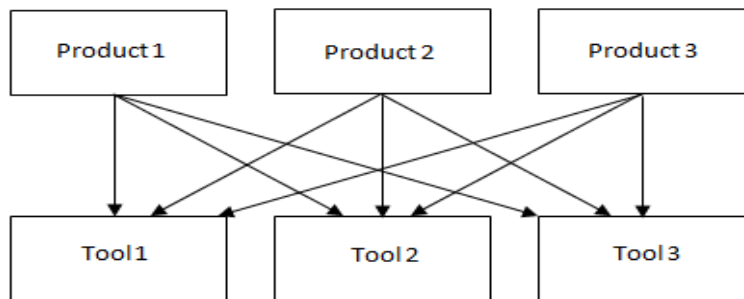


Figure 2.1: Product-Tool combinations emulating the high-mix manufacturing environment

Thread #	Product Number	Tool Number
1	1	1
2	1	2
3	1	3
4	2	1
5	2	2
6	2	3
7	3	1
8	3	2
9	3	3

Table 2.1: Control Threads formed from combinations of Tool and Product

Overlay state data was generated using simulations for 5000 runs by adding common disturbance signals as well as. The noise signal was made adjustable by specifying a multiplier. EWMA RtR filter was applied to this data to observe the performance with the simulated data. Figure 2.2 shows the data generating using MATLAB simulation for 5000 run data. The data is divided into nine different threads described before. Each thread shown in a different color is a combination of respective states of the tool and product used for the particular run number.

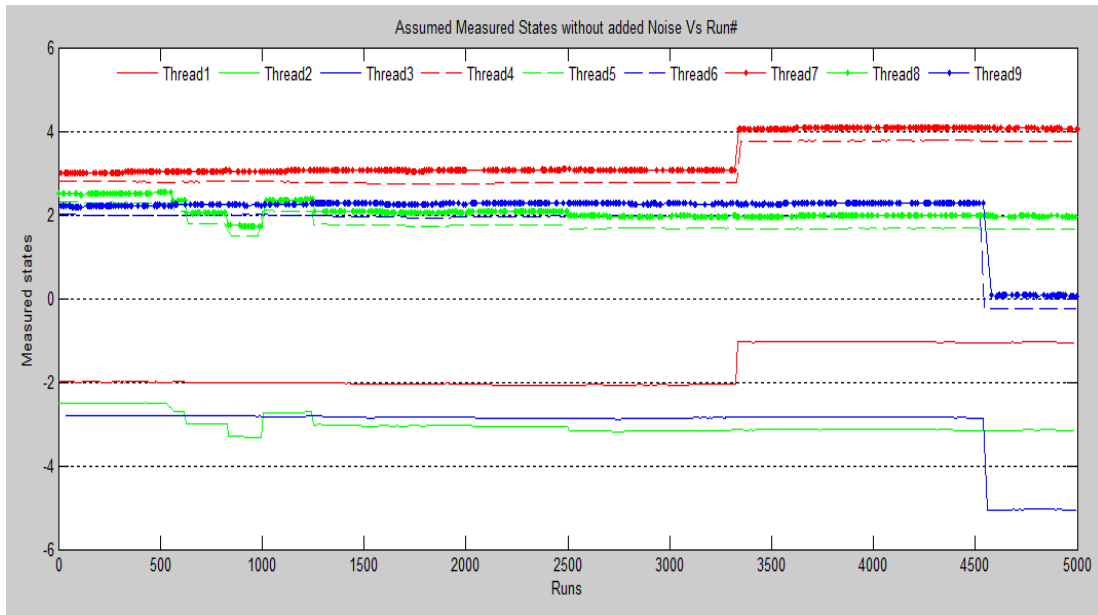


Figure 2.2: Simulated Data for 5000 runs.

2.2 TOOL DEDICATION

In practice, if the same lithography tool is used for patterning successive layers of the product, minor imperfections in the pattern are repeated from layer to layer and hence effectively cancel them from the overlay error. Hence the ideal solution to reduce the overlay error would be to use the same tool to process all the product layers. However in a fab facility, there are several products being manufactured at the same time and the number of tools is small due to the manufacturing costs of the equipment, hence it is almost impossible to dedicate a tool to a particular product. Also, dedication would decrease the total number of products being manufactured. The next product would have to wait for the current product being manufactured completely before its manufacture would begin. Thus, normally the product layers are processed on whichever tool that is

available to minimize the process downtime. This leads to mix and match of tools and product layers causing larger registration errors. Thus tool dedication and matching is performed to an extent that is possible.

Every product has a choice of tools to get manufactured out of which one of the tools gives the best results. However, that particular tool might not be always free to accept a new product or layer. This results in a compromise on the accuracy against the production and a different available tool gets selected. Under the event of maintenance, tool failures or replacement the distribution of the products on the available tools again changes.

To simulate tool dedication, following algorithm was used.

Every tool has three kinds of status condition:

1. Free- No product is being manufactured on the tool and it can accept a next product/layer
2. Busy- The tool is busy processing another product/layer and cannot accept the next product/layer.
3. Down- The tool is not working/ needs maintenance. In this kind of status, a backup tool replaces primary or secondary tool.

The tool status for each tool was selected randomly to create all different combinations of tools and products. The probability distributions between the tool selections can be optimized to get the best results. Thus, dedication of tools is a problem of constrained optimization.

2.3 MOVING WINDOW APPROACH

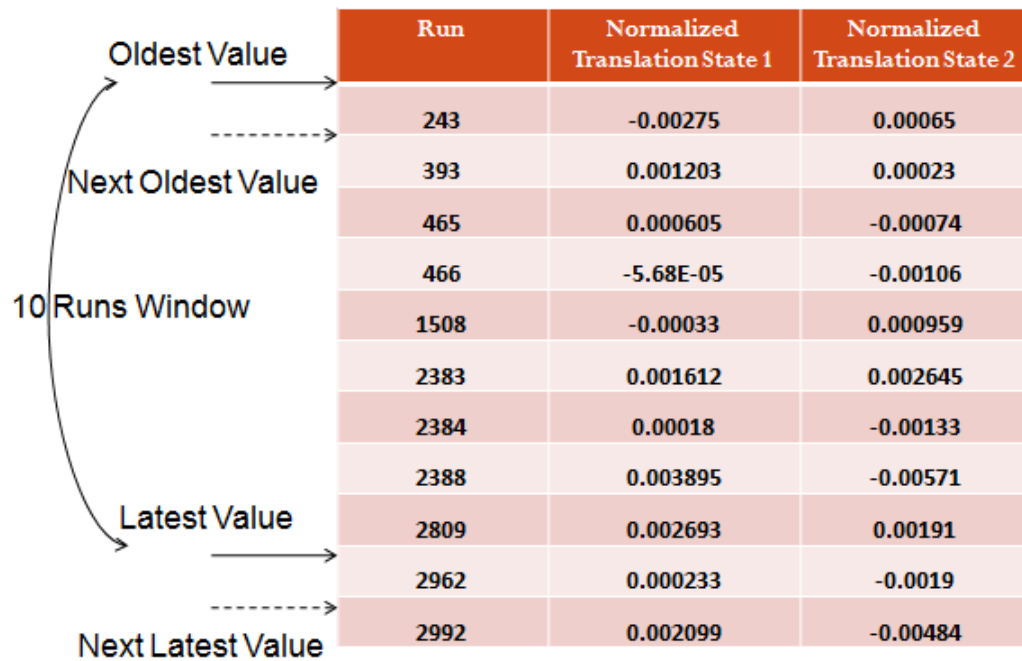


Figure 2.3: Moving Window with a size of 10 runs

One of the factors that can affect the performance of the EWMA controller is the amount of historical data considered. Hence, a moving window of data is used. The historical data of past n runs is considered for estimating the $(n+1)^{\text{th}}$ run, where n is the size of the window. This ensures that the estimate is based on the past n runs which are current and no obsolete data is considered. The choice of window size is a trade-off between maximizing the use of available data and minimizing the computation time. The general principle for deciding the window size is that the window size should be big enough to get a good estimate of the model. Figure 2.3 shows the moving window for 10 runs used in the simulations. The first column shows the run number, while the other two columns show the normalized translation state measured at the respective run. The window of the data moves every run as shown in Figure 2.3. The window of runs 1-10 is

used for EWMA estimation for the eleventh run. Then, second run replaces the first run which was the oldest run while the eleventh run replaces tenth run which was the latest run which forms the new window of data. This new window is used for estimating the twelfth run and so on. The window size is fixed to 10 runs depending on the history of the process.

2.4 SAMPLING AND SCHEDULING

Metrology tools used in semiconductor manufacturing are really expensive. The processed wafers are moved to the metrology tool for every measurement which makes metrology of each wafer infeasible. Sampling algorithms are used to select the best wafers for measurement. The number of samples is kept as minimum as possible while keeping tighter control and the process stable. The samples to be measured are scheduled for a measurement on the metrology tools using scheduling algorithms. Sampling will be discussed in a little more detail in Chapter 3.

2.5 OVERVIEW OF THE SIMULATIONS TO BE PERFORMED

Following is the list of simulations that were performed to study the effects on the performance of threaded control:

- a. Effect of change in EWMA filter weight
- b. Effect of large step
- c. Effect of several steps
- d. Effect of noise
- e. Effect of drift

- f. Effect of Metrology Lag and out of order metrology
- g. Random and Uniform Sampling

2.6 PERFORMANCE SIMULATIONS AND RESULTS

The previous section gave the background of performance simulations that were run for evaluating the threaded controller. This section will summarize the results obtained by running the simulations for the example data. The same sets of simulations were performed on the sample data set from real fab to confirm the correctness of the results. The data set consisted of the X and Y translation errors, magnification and rotation errors, etc for different lots. The data was separated using the factor 'layer' as the context. The following sections will discuss the results obtained from the simulations.

Effect of change in EWMA filter Weight

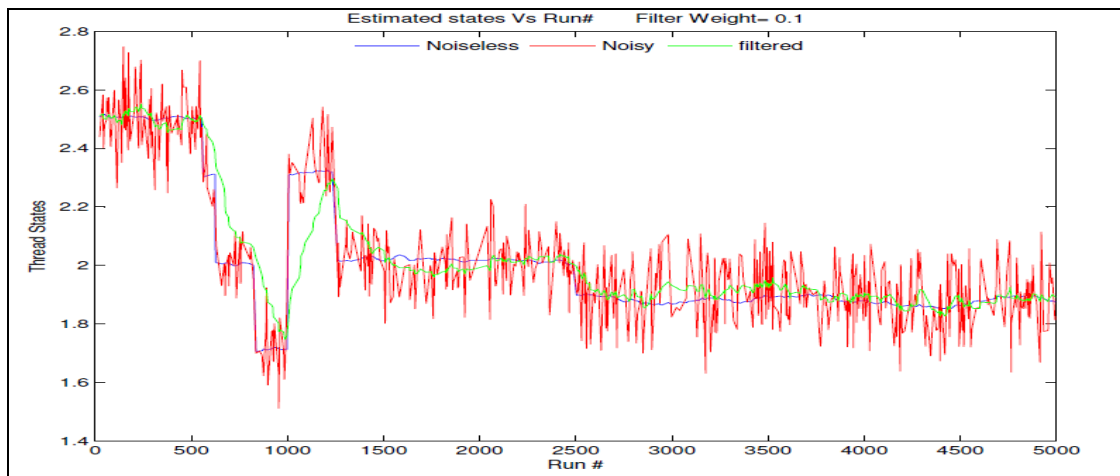


Figure 2.4: EWMA filtering of Thread 8, effect of small filter weight $\lambda = 0.1$

Figure 2.4 shows the effect of EWMA filtering of noisy measured signal. The measured state for 5000 runs was simulated using MATLAB code. The measured signal was

superimposed with normalized random noise of zero mean and a standard deviation of 0.001. EWMA filter was tuned to a filter weight of 0.1 for thread 8. The filter tries to nullify the effect of noise and the disturbance. The signal in blue represents the pure measured state; the signal in red represents the state after superimposing the noise. In the above plot, due to the small filter weight, the EWMA filter responds sluggishly. The signal in green in the diagram above is the EWMA filtered signal.

It indicates that the filter applies more weight to the old estimate than the new value. Hence the filtering of the data is aggressive. The variance in the state estimate is much lower as a result of effective filtering of the noise. As a result the filtered signal is very close to the noiseless signal.

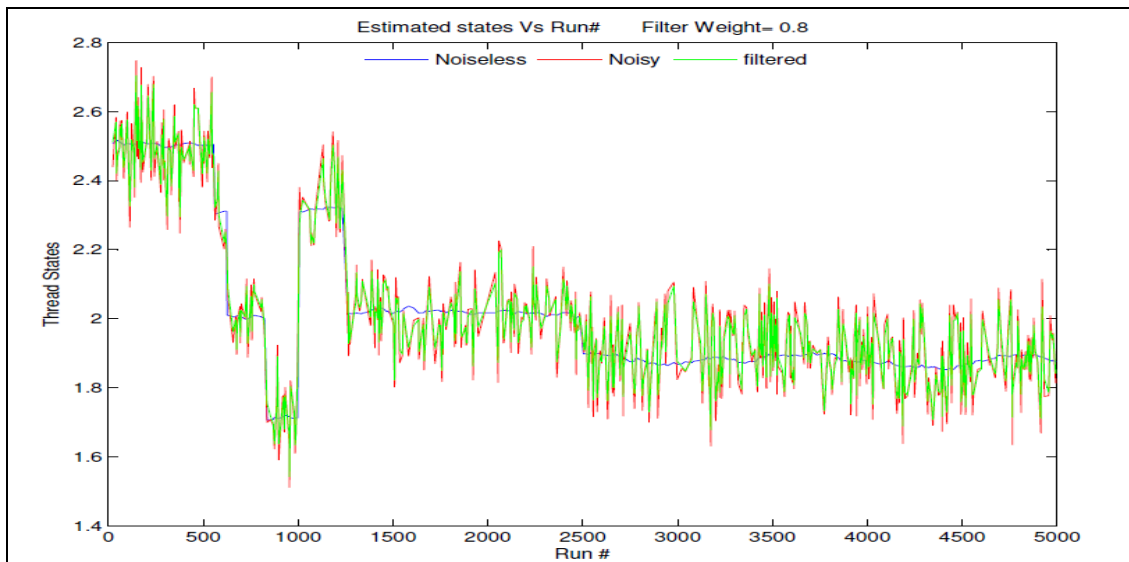


Figure 2.5: EWMA filtering of Thread 8, effect of large filter weight $\lambda = 0.8$

Figure 2.5 shows the behavior of the EWMA filter at a larger filter weight ($\lambda = 0.8$). As can be seen from the graph above, the filter responds faster than it was observed for a smaller filtering weight of 0.1. At the same time, the variance in the filtered thread states

is larger, which means that the data was less aggressively filtered. Thus, the choice of filter weight is a tradeoff between the response time and the aggressive filtering of data.

2.6.1 Optimal Tuning of EWMA filter

The 5000 run data was separated into nine threads depending on the tool and product for each run. To find out the optimal value of tunable parameter λ , simulations were run by varying the filter weight and the effect on the mean square errors in the estimates was observed. The simulations were run on all the nine threads separately to observe the effect of tuning. Figure 2.6 shows the effect of change in filter weights on the threaded state of thread1 data when the filter weight is increased from 0.1 (top left plot) to 0.9 (bottom right plot). Figure 2.7 shows the effect of change in the filter weight on the mean squared error in the state estimations for Thread 1. It was observed that the MSE was the lowest at filter weights between 0.2-0.4. These filter weights are typically used in semiconductor manufacturing applications. Similar results were observed by Bode [3] and Wang et al.[20] .

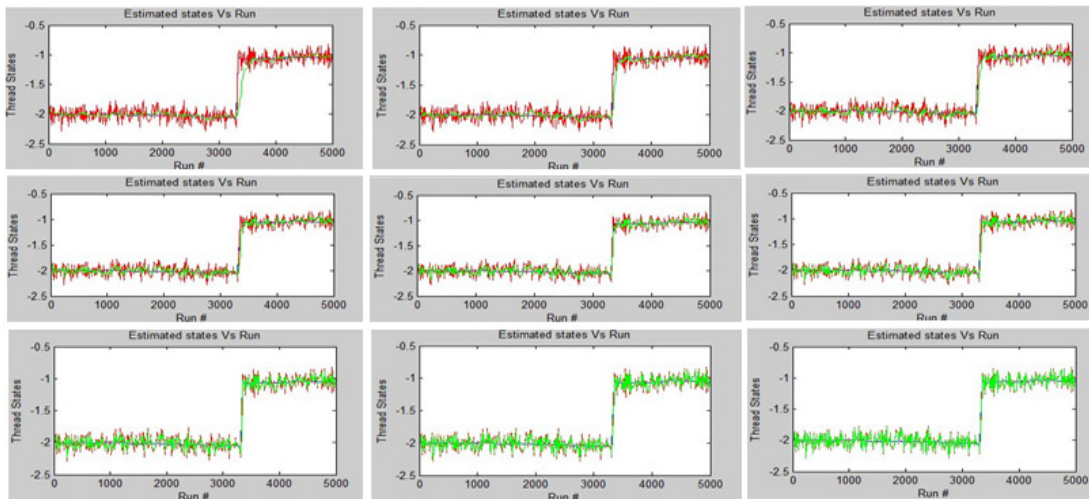


Figure 2.6: Effect of increasing filter weight on the MSE's in the estimated states. Filter weight (λ) is varied from 0.1 in the top left plot to 0.9 in the bottom right plot.

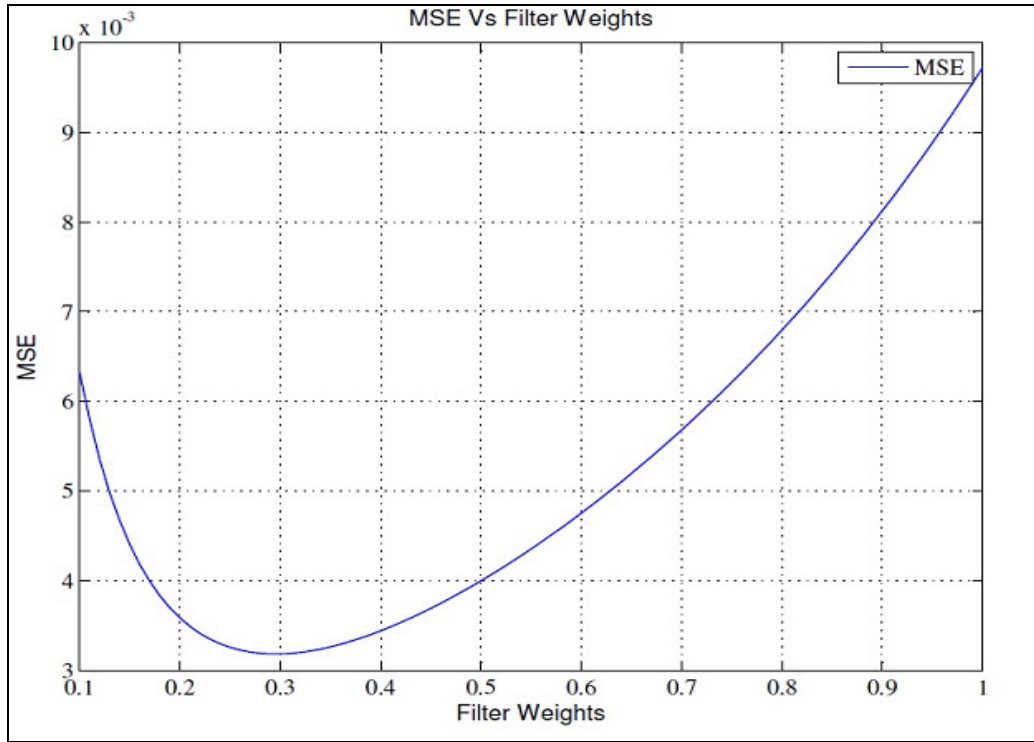


Figure 2.7: Effect of filter weight on the MSE's in the estimated states.

2.6.2 Effect of a large step

A step disturbance is a commonly observed disturbance in the lithography process. To study the effect of a step disturbance on the control performance of the EWMA filter, simulations were run. As discussed before the optimal filter weight for semiconductor manufacturing processes is in the range of 0.2-0.4. The EWMA filter was run with an assumed filter weight of 0.3 for all threads. The results are shown in Figure 2.8. A large step disturbance was introduced in tool 3 to emulate a disturbance due to a maintenance event. This step disturbance affects threads 3, 6 and 9 because all of them make use of the same tool 3.

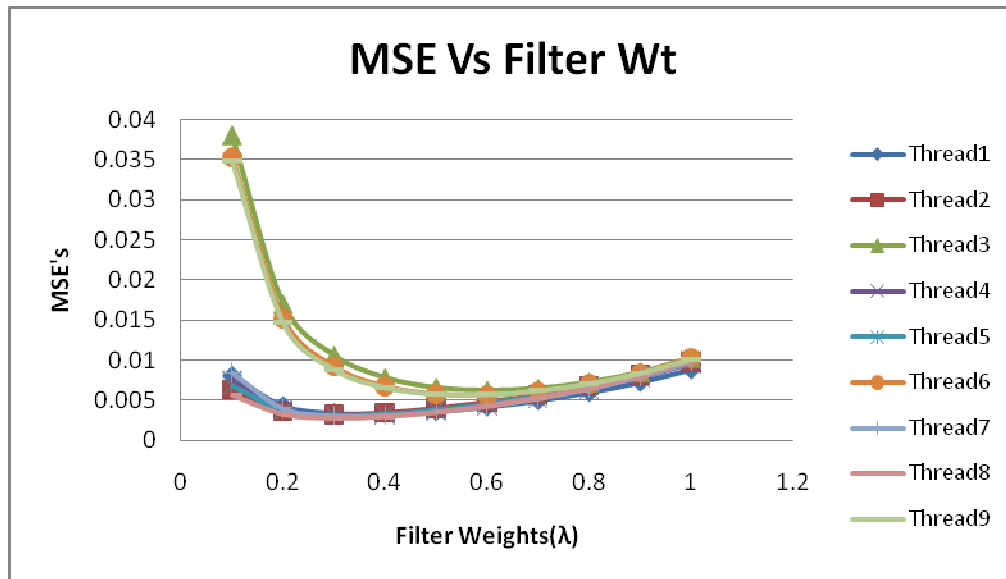


Figure 2.8: Effect of filter weight on the MSE's in the estimated states.

To see how the threads 3, 6 and 9 behave when the step size for tool 3 is reduced, simulations were re-run. Figure 2.9 shows that the MSE decreases with percentage reduction in the step size.

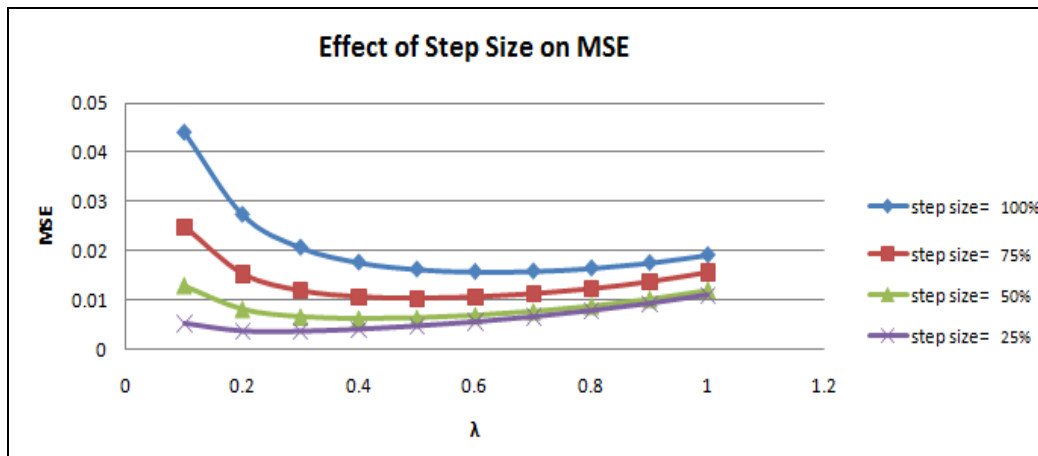


Figure 2.9: Effect of change in the step size on the MSE for Thread 3

2.6.3 Effect of several steps of small magnitude

If there are several steps in a thread it would severely degrade the performance of the EWMA control. To confirm this, simulations were performed. Steps of small magnitude but repeating at regular small interval were introduced in Tool 2. Thus the performance of all three threads using this tool would be significantly different than the other threads. Figure 2.10 shows the result for this simulation. Threads 2, 5 and 8 have MSE's significantly different than the other threads at all values of the filter weight.

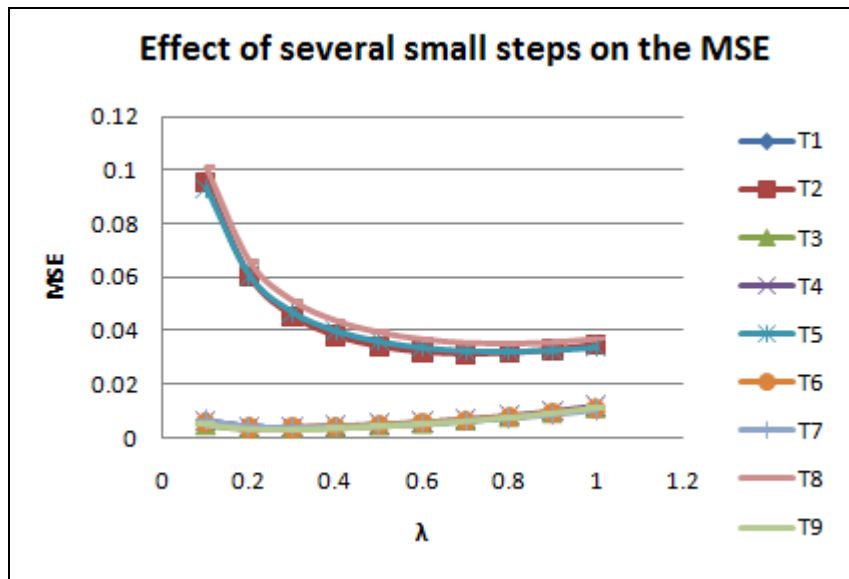


Figure 2.10: Effect of several steps of small magnitude on the MSE

Simulations were run to compare the performance of threads having a large step and the threads having the effect due to steps of small magnitude. It was observed that small repeating steps in the context are worst and severely degrade the control performance.

2.6.4 Effect of drift in the context states

To study the effect of drifting signal, simulations were performed using combination of original step signal (left plot in Figure 2.11) and a ramp increasing at 0.0001 every run. The resultant disturbance is shown in plot on the right.

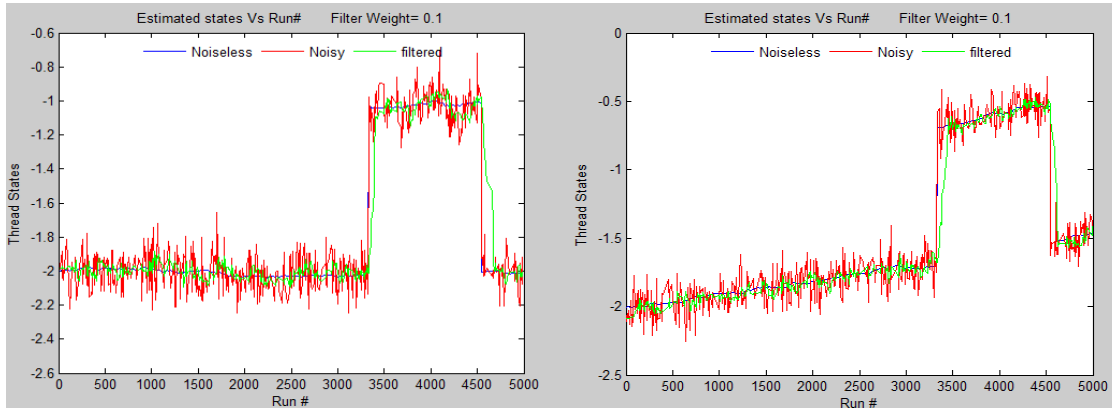


Figure 2.11: Simulation for drifting thread state

Figure 2.12 explains the effect of drift on the MSE's. The drift in the states does add to the variation but it does not show a significant increase in the mean square errors at smaller filter weights. With a filter weight of 0.3 (which is the optimal tuning value), there is almost no effect of drift on the MSE.

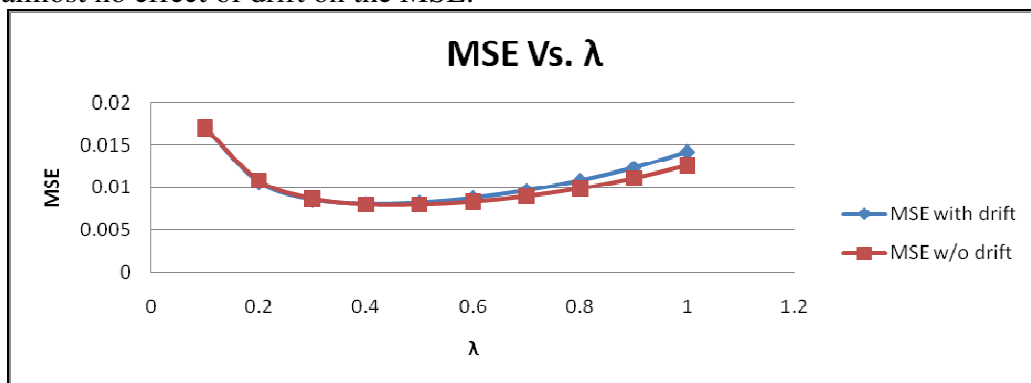


Figure 2.12: Comparison of MSE vs. filter weights for thread states with and without any drifts

2.6.5 Effect of Noise

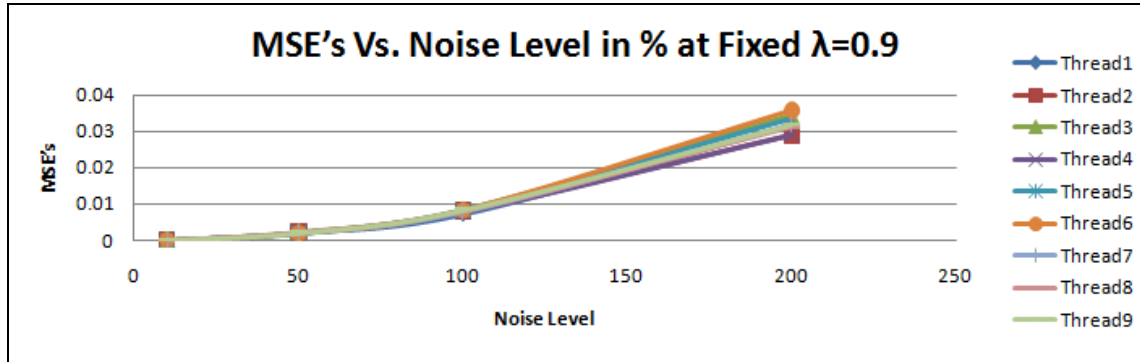


Figure 2.13: Effect of Noise on the MSE

Figure 2.13 explains the effect of noise on the MSE's. It can be observed from the figure that higher noise level demands lower filter weights since the filtering needs to be more aggressive. At higher filter weight as the noise level is increased the MSE shows an increasing trend because the filtering shows fast response but less aggressive filtering of the noisy signal.

2.7 SAMPLING SIMULATIONS

To study how the threads of data behave when the data is sampled, simulations were performed with two sampling techniques, random and uniform sampling. In random sampling, the wafers to be measured are selected randomly from the runs; whereas in uniform sampling method, wafers are measured every N runs.

2.7.1 Random Sampling

To see how the random sampling technique would affect the performance of the EWMA control, the sampling frequency was varied from 10-100%. The simulation was run for 100 or more times and averaged at each sampling frequency to get stable results.

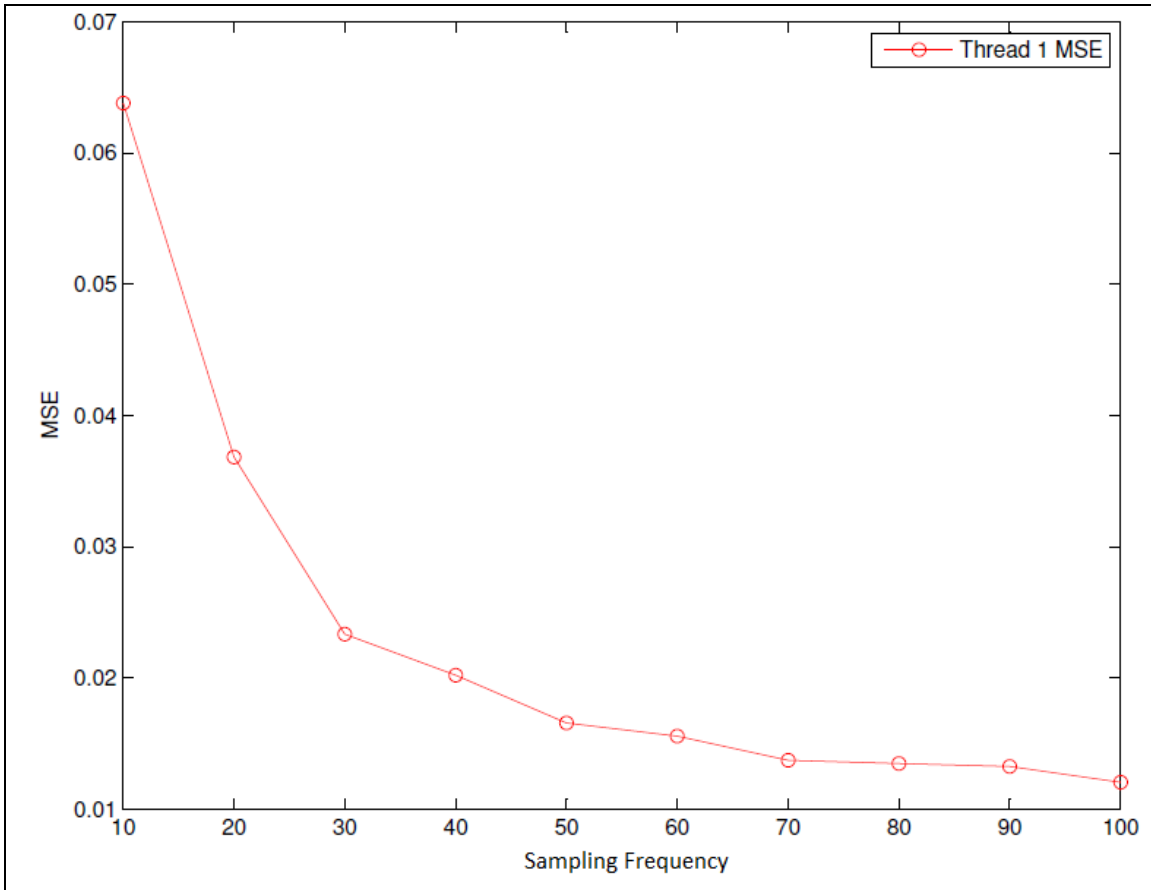


Figure 2.14: Effect of Random Sampling on the MSE for Thread 1

As can be observed from Figure 2.14, the MSE tends to reduce with increase in the sampling frequency. Each point in the plot represents mean of results obtained after running 100 simulations at each sampling frequency. Lesser the number of sampled values within the given thread, lesser is the accuracy of the estimate. Thus, the MSE increases at lower sampling frequency. The MSE is lowest at a 100% sampling frequency. 100% sampling frequency would mean that each run within the thread has a measurement. Such would be an ideal case, but it is infeasible and expensive to measure each and every run/lot in the process. Random sampling can yield worse results if the

data selected randomly are bunched at particular sequence of the results (e.g., few runs at the start or the runs at the end).

2.7.2 Uniform Sampling

The sampling frequency was decreased uniformly from 100%-0% in steps. Measurement at every run creates 100% sampling, measuring every other run creates 50% sampling and so on. The uniform sampling rate cannot be smaller than measuring every 10 runs because it has been studied that there can be chance of significant degradation in the performance if there is no measurement in the last 10 runs.

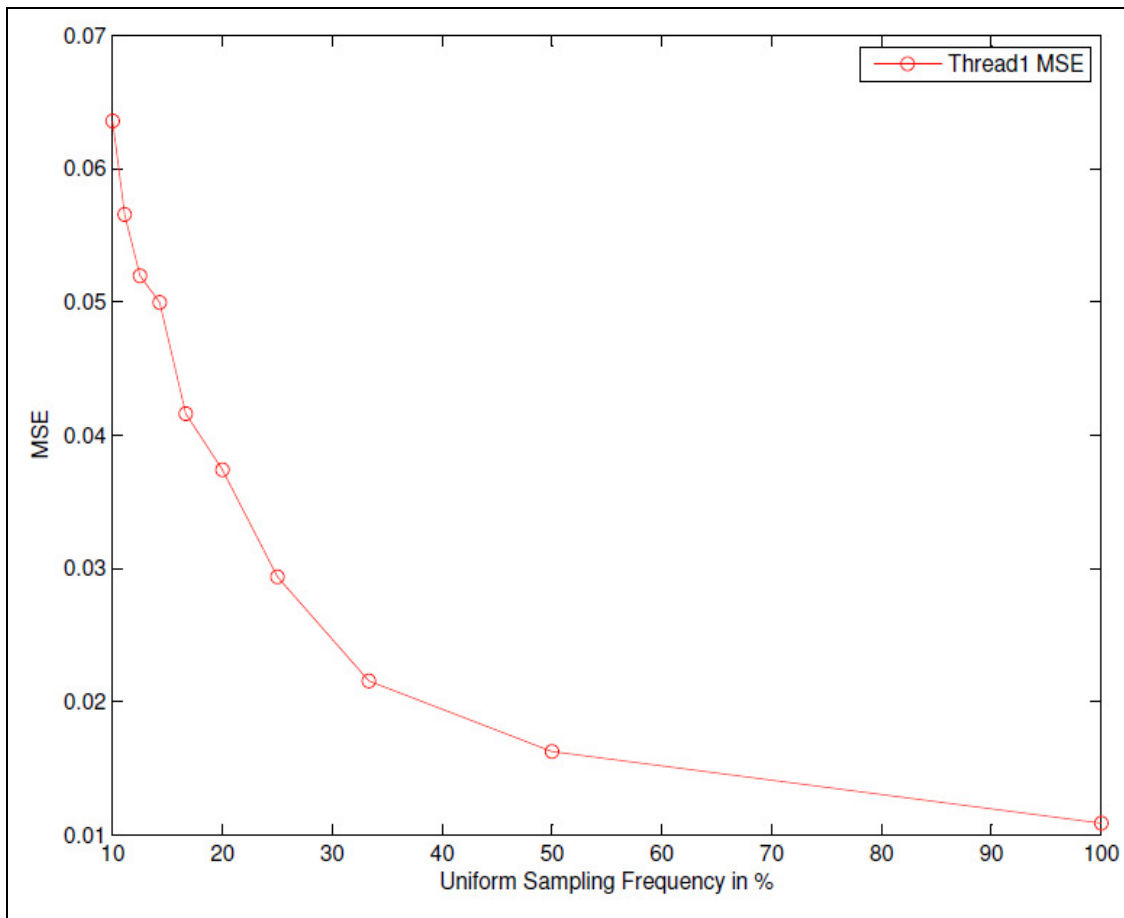


Figure 2.15: Effect of Uniform Sampling on the performance

Figure 2.15 shows the effect of uniform sampling frequency on the MSE. The MSE decreases with increase in the percentage uniform sampling frequency. This result is similar to what was observed in random sampling. More sampling results in more data within the given thread and hence results in better control. The least MSE was observed at the uniform sampling rate of 100% which means that every run is measured which is unfeasible and expensive as discussed before.

Both random and uniform sampling techniques are very old and better, more complex techniques have been developed in recent years, like Mixed Integer Linear Programming (MILP) technique developed by Good et al.[21] but simulating these techniques is beyond the scope. The main goal of sampling can be summarized as to minimize the measurements while keeping tighter control and better stability in the process.

2.8 EFFECT OF METROLOGY LAG AND OUT OF ORDER METROLOGY

The presence of delays hinders the measurements from reaching the controller at the right time and thus affects the control performance. Such delays are always present in any semiconductor manufacturing process and need to be accounted for. Two such types of delay are process delay and metrology delay. Process delay is inherent in the process whereas metrology delay results due to constraints in the metrology or measurement of wafers.

In a fab there are multiple processes running simultaneously. The metrology equipment is very expensive and hence should be used optimally. The wafer to be measured has to be removed from the production line and moved to the metrology tool for measurement. Measuring each and every wafer after the processing is almost impossible and hence different sampling schemes are used to minimize the number of

wafers/lots to be measured while maintaining the process stability. Due to the high volume of manufacturing in a running fab there is always a backlog of wafers waiting for the measurement tool to be free for a measurement. Thus it causes a delay in the process which is called as Metrology Lag. The lots of wafers to be measured are usually arranged as per the priority. The priority of measurement lots depends on a number of things, e.g., the delivery date for the order. In such cases the urgent lots are moved ahead of the lots that were processed before them. This leads to Out of Order Metrology, which means the lots are not measured in the sequence that they were processed. This demands a technique to keep track of the processing and metrology order of the wafers.

Harrison et al.[22] compared threaded EWMA with context-based EWMA and found that the context-based EWMA gives better results in a high-mix of products but threaded EWMA is less sensitive to metrology delays.

2.9 SUMMARY

In this chapter the evaluation of threaded control technique with the use of performance simulations was explained and the results obtained were discussed in detail. The effects of filter weight, step disturbance, noise, sampling and delays were studied. The conclusion from these results for the evaluation of threaded control in high-mix environment will be discussed later in the last chapter. The next chapter will describe the significant drawbacks of threaded control in the high-mix environment in detail.

CHAPTER 3

Drawbacks, Extensions and Alternatives to Threaded Control

3.1 DRAWBACKS OF THREADED CONTROL

It has been a decade since threaded control was first used as an effective control algorithm. In the past years, in many fabs a few products were manufactured and thus the threaded control did not experience significant drawbacks. ASIC fabs and foundries manufacture different products in the same manufacturing facility and thus have a high-mix of products, sometimes well over 100. In recent years it became clear that the threading approach has disadvantages when applied to a high-mix environment under certain circumstances, which are studied in the coming section.

- Large number of estimation variables
- Data poverty
- Low runners
- Lack of information sharing between the threads

3.1.1 Large number of Threads

As we have seen before, threaded state estimation identifies groups of lots having roughly the same incoming process state. Each group is segregated from the rest of the groups based upon the context criterion that determines the incoming state. These groups are referred to as control threads, contexts, or streamlines in the semiconductor industry. The threaded control methodology lumps each of the states into a single, unique disturbance for the model. The thread definition contexts are decided based on heuristics and historical trends. More contexts lead to more threads. Thus, threaded control involves

danger of ‘thread explosion’ by generating a large number of variables to be estimated in the case of high-mix manufacturing.

3.1.2 Data Poverty

Each criterion in the thread definition divides the data set into smaller data sets for each thread. Thus there are fewer data points within individual threads for the estimation of the control state, which would degrade control performance. This scenario is called ‘Data Poverty’ [16], [3].

3.1.3 Low running Threads

A fab operation has a high-mix of products; some of the products have many lots and many products of which only a few lots are run. The lots that do not run frequently are called low-runner products. These low-runner products pose specific challenges to the control system. Some of the feedback loops in the fabs may operate with long time periods between data points in the feedback loop. This long delay may result in a loss of information about the process tool contribution to the variance in those products. The state of the process tool may experience drifts or shifts during the time period in between low-runner product feedback loop data points. These changes to the process tool state cannot be inferred by the controller state until the next lot with the same context is run. At that time, the controller sees the process tool state change as a disturbance to the particular feedback loop that must be rejected. Each feedback loop must comprehend and reject this disturbance separately, because there is no information sharing between the threads. Zheng et al.[23] did a study with more than 70% lots having less than 10 runs.

3.1.4 Lack of information sharing between the threads

Suppose a tool is to be scheduled for maintenance because of degradation in its performance, thus we know the time when it would be taken off production. At that point

of time, all the threads associated with that tool experience a step. Since there is no sharing of information within the threads, each of the threads using that tool needs to reject this disturbance separately. The controller must be able to quantify the disturbances and to determine whether the disturbance is associated with a specific tool/wafer.

Miller [8] proposed five different strategies for RtR control in a high mix environment: threads, grouping, similar controllers, single global controllers and information sharing controller. Some of the new non-threaded techniques like JADE [11], [17], [24] and Random walk [12], [13], discussed in a later section use the concept of an information sharing controller which has shown some advantages over the traditional threaded controllers. Wan et al. [25] have demonstrated control of lithography overlay data using data sharing between a machine controller and process induced error controller.

3.2 POSSIBLE EXTENSIONS TO THREADED CONTROL

3.2.1 Thread definitions with larger number of threads

Threads are defined based on various contexts like tools, reticles and product information. The contexts deciding the thread definitions are selected based on heuristics and the actual production data. ANOVA test/regression analysis could be done before defining the threads. Identification of the factors that have significant effect on the variability could help in thread definitions.

ANOVA can be used to find out the significant factors as well as the significant interaction effects before deciding whether to form threads with the factors separately or in a combination. Similar analysis can be done using regression. Ma et al. [26] used

ANOVA based model for RtR control in high mix process. Vanli et al.[27] have proposed a rigorous statistical method to identify the contexts for thread definitions.

3.2.2 Dynamic thread definitions/combination using threaded EWMA

To develop methodology for thread combination, a tolerance band was defined for the variance in a thread depending on the historical results. Threads whose variance falls within the same tolerance band would be eligible for recombination into a single thread; otherwise they would be kept separate. This would possibly reduce the total number of threads without a drop in the performance of the estimation.

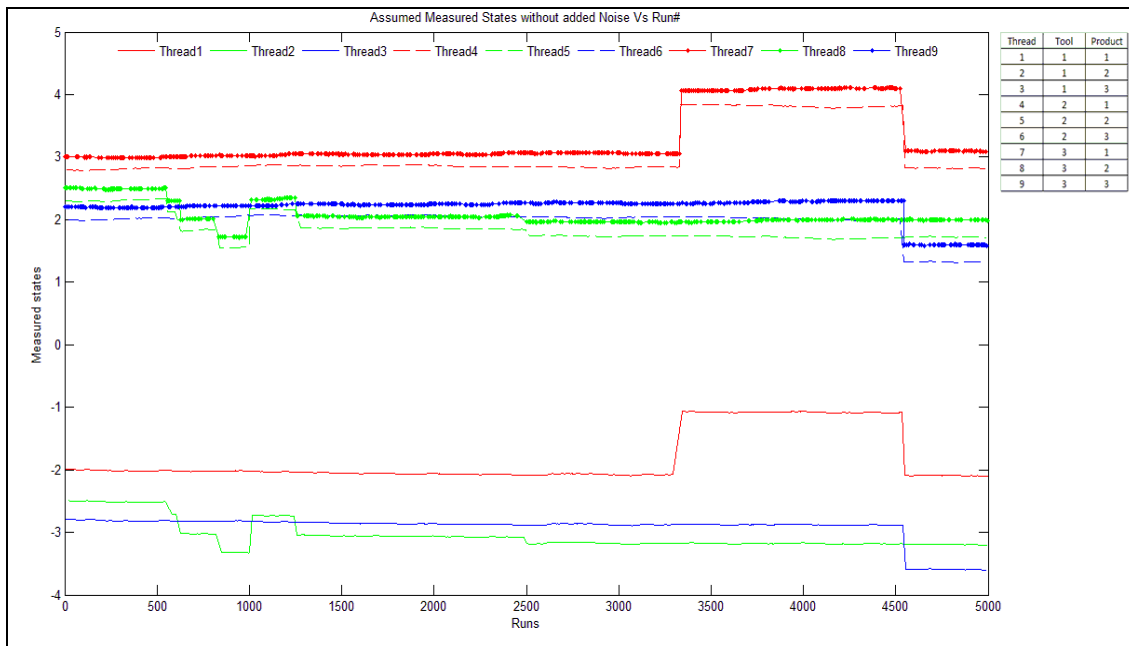


Figure 3.1: Actual states for simulated data divided into 9 threads.

Figure 3.1 above shows the plot of individual states against the runs. It can be visually observed from the figure 1 that threads (4, 7); (6, 9); (5, 8); (2, 3) are close to each other. But we need a criterion to check whether these threads are close enough to

combine them into a single thread. Thus we compare the means of the two threads to define the proximity factor.

$$\text{Proximity factor} = \text{Abs} \{ \text{Mean} (\text{Thread4}) - \text{Mean} (\text{Thread7}) \}$$

The proximity factor is then compared to the tolerance. If the proximity factor is less than the tolerance the two threads would be combined. Where, Thread4 and Thread7 are vectors containing simulated states with the added noise for all the runs within the particular thread.

The above logic defines how the threads to be combined are identified. This logic is performed on all combinations of threads before deciding which threads should be combined based on the tolerance value. Finding out the best value of tolerance is important. The tolerance value should be as tight as possible. For a larger tolerance, the number of threads combined is greater, thus reducing the total number of threads. But a higher tolerance causes loss in performance. The tolerance value should be tuned to reduce the total number of threads without loss of performance.

The three tool and three product simulation studied earlier was used for testing. Logic was developed to determine those threads that should be considered for combination. The results for three different levels of tolerance defined for the simulation data are as shown in Table 3.1.

Tolerance	Threads to be recombined	Details	Total # Threads
0.0005	None	Tolerance is too tight, no threads qualify for recombination.	9
0.005	(Thread5, Thread8) and (Thread6, Thread9)	Thread 5 and 8 form thread 10 and thread 6 and 9 form thread 11.	7
0.01	Thread6, Thread9 and Thread8,Thread9	Threads 6, 8 and 9 all fall into the tolerance limit; hence three of them would be combined into single thread.	8

Table 3.1: Effect of Tolerance on the total number of threads

Next, the tolerance was held constant at a small value of 0.005 and simulations were run to find out the effect of combining threads on the estimation accuracy. The results of this simulation are as shown in Table 3.2 and Figure 3.1.

Threads combined	MSE of Separate Threads		MSE of combined thread
Threads 5,8	T5 0.002886119	T8 0.003033331	T10 0.003555518
Threads 6,9	T6 0.002189333	T9 0.002547776	T11 0.002417298
Thread 4,7	T4 0.007849508	T7 0.008571462	T12 0.00444002

Table 3.2: Effect of thread combinations on the estimation accuracy at filter weight=0.3

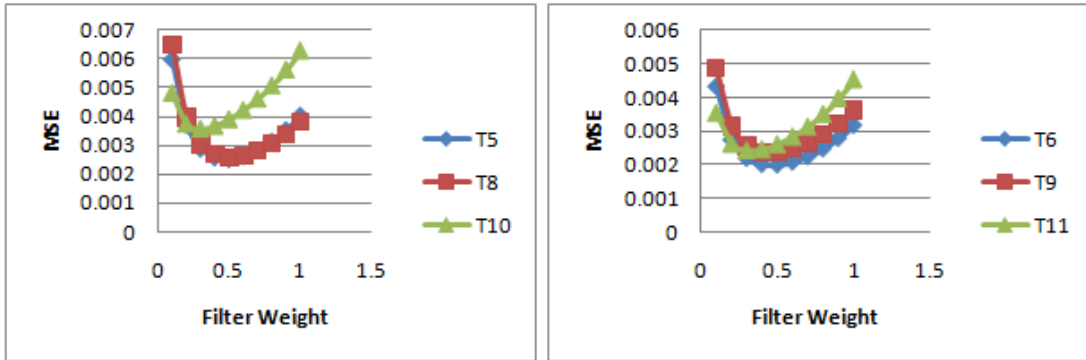


Figure 3.2: MSE against the filter weights for individual and combined threads

The MSE increases if there are disturbances in the threads that were combined. For a smaller proximity factor, the MSE will be lower for the combined thread. In the 1st case, Threads 5 and 8 have the difference in their means within the tolerance but they have disturbances which cause the MSE of combined threads to go up a little. Combination of threads 6 and 9 has an MSE in the range of individual MSE's of the 2 threads. Combination of threads 4 and 7 has a lower MSE than individual MSE's.

Performance of the combined thread largely depends on the nature of the individual threads. The tolerance could be decided as a tradeoff between the number of threads and performance. Optimization/statistics could be used to find the best value of tolerance.

3.2.3 Dynamic Sampling

The control performance/accuracy of the estimates can be used for make a decision about whether to sample for the next run or not.

$$\hat{e}_{k+1} = \lambda e_k + (1 - \lambda) \hat{e}_k \quad \dots (1)$$

While estimating the disturbance ('e' in the above equation) at the $(k+1)^{\text{th}}$ run, we know the measured value as well as the estimate for the k^{th} run. We can compare these two to estimate the accuracy and define some tolerance based on experience. If the estimated accuracy falls within the tolerance, then we can skip the measurement for the next run falling within same thread and instead measure the next run within the thread. A similar idea has been used by Lee [28], where the base uniform sampling frequency is selected based on maximizing the net profit and the rate is decreased or increased based on performance of the process. The sampling algorithm samples more if the process tends to go out of control and thus gives better performance while minimizing the number of measurements.

Simulations were developed based on the dynamic sampling algorithm discussed before. The simulations were run on the real production data for a single thread having 960 data points. By varying the tolerance and the effect on the total number of samples and the MSE was observed. The following few plots explain the results.

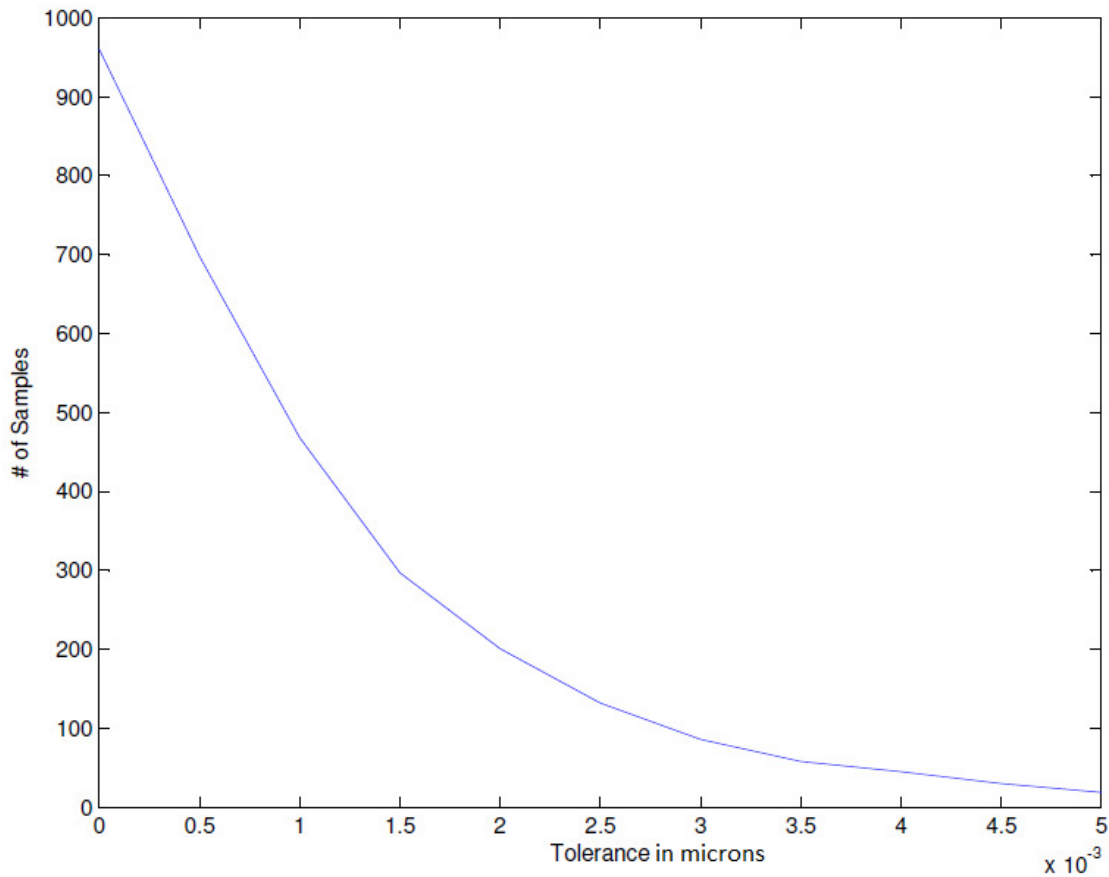


Figure 3.3: Number of Samples vs. the Tolerance

Figure 3.3 shows that the algorithm samples data more when the tolerance is tighter and the error term falls outside the tolerance band occasionally. When the tolerance is zero, to be able to skip the measurements the error term should also be zero. This is practically impossible and thus in this case the sampling would be 100% (all wafers). If the tolerance is set to be 5×10^{-3} , the tolerance is fairly loose and hence almost all of the readings fall within the tolerance and there is only 1 measurement (1st run).

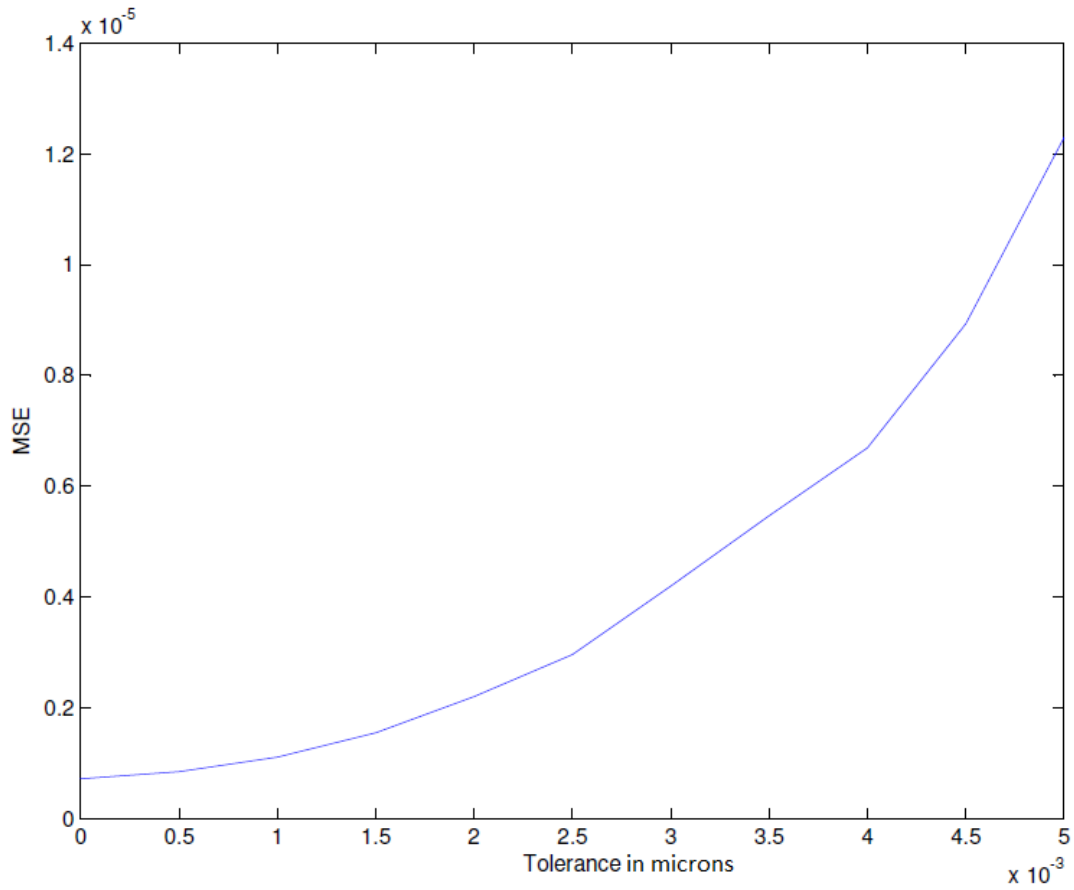


Figure 3.4: MSE vs. Tolerance

Figure 3.4 shows that the MSE increases when the tolerance is increased, which is expected. A larger tolerance band leads to data poverty for the threads as shown above and hence the estimation error/ MSE is more.

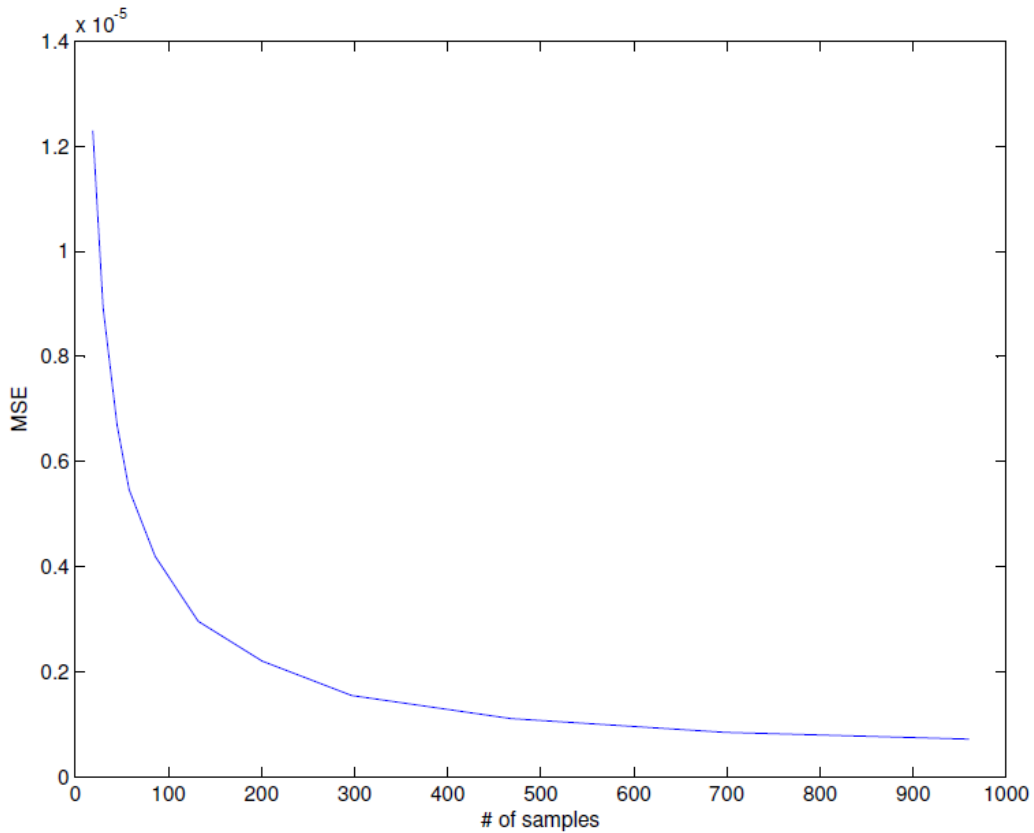


Figure 3.5: MSE vs. Number of Samples

Figure 3.5 shows that the MSE decreases as the number of samples increase. From the graph it can be observed that the MSE does not decrease much when the numbers of samples go past 300. Thus, accuracy in the estimates could be compromised to reduce the total number of samples. This trade-off between accuracy and number of samples can be achieved by setting the tolerance level to the correct value between 0 and $5 \cdot 10^{-3}$. Finding the appropriate value of tolerance depends on several business rules, economical concerns, etc. An optimization problem could be built, where the objective function would be to minimize the number of threads based on the constraints. This problem is left for the future work.

3.3 ALTERNATIVES TO THREADED CONTROL

In the last few years, non-threaded state estimation methods have drawn considerable interest. These methods share information among different contexts. Assuming that the interaction among different individual states is linear, different algorithms such as linear regression and the Kalman filter can be applied to identify the contributions from different variation sources. One of the chief difficulties in these methods is the loss of observability in the context matrix which needs to be inverted at every step. Each method utilizes a different approach to handling this problem and making the system observable.

3.3.1 Just in time adaptive estimation

Just in time adaptive estimation (JADE) algorithm uses recursive least squares parameter estimation to identify the contributions to the variation that are dependent upon manufacturing context. JADE was first developed by Firth et al.[11] and had several advantages over threaded control. JADE had ability to estimate the separate context-based states at the same time. Along with the improvement in state estimation, JADE is able to indicate exactly which context item has undergone a disturbance. JADE shows less degradation in performance with delayed processes as compared to threaded control. Several such advantages have been studied and proved through simulations [11].

JADE has a couple of limitations listed in Firth et al.[11] and Wang et al.[19]. JADE shows improved control vs. EWMA but it depends on the assumption of a correctly specified disturbance model. JADE models the disturbance as a linear combination of context based states from each of the contributing context items. If all important context items contributing to likely disturbances to the process are not included in the JADE disturbance model, the algorithm has difficulty rejecting the unknown disturbance. If the disturbance model is nonlinear and a suitable linear form cannot be

used within the operating region, then the performance degrades. Also, JADE needs qualification runs to obtain a unique solution. Wang et al.[19] demonstrated degradation in performance of JADE due to resetting the estimate covariance at each run. It also suggests increasing the weighting on the previous estimates to improve the performance of JADE when applied to stationary processes.

3.3.2 Recursive least squares estimation

Recursive least squares estimation (RLS) method was recently developed by Wang et al.[19] as an alternative to threaded control. In this method recursive least squares are used as estimator in the RtR control framework. EWMA-type and RLS-type estimates are compared under measurement delay, measurement noise and deterministic drift.

3.3.3 Random walk model

Random walk model (RWM) [12], [13] combined with moving window and least squares solution provides better estimates for processes with high-mix of products and tools with many low runners as compared to alternative methods.

All these control algorithms use recursive least squares solution and have demonstrated advantages over the standard threaded EWMA. They have information sharing; and avoid data poverty and thus may be better state estimators than threaded EWMA.

3.4 SUMMARY

Although non-threaded control algorithms have show better results than the threaded control method under certain assumptions and simulations, they lose the advantage of threaded control [16], [3]. Threaded control lumps the parameters into a

single disturbance and hence removes the need for estimating each disturbance individually. Another advantage of threaded control is that the interaction among different individual states can be nonlinear, provided that the process operating point does not change dramatically. The random walk model for disturbance estimation in high-mix environment developed by Prabhu [12] demonstrated better results than JADE when applied to simulated data and processes. Although the model is relatively new, it eliminates the need for qualification runs and also augments the context matrix with the identity matrix for making it invertible. The random walk model takes into account the context matrix and hence includes information sharing within threads. The limitations/drawbacks for the random walk model are yet to be studied and further research and experimentation needs to be done with it.

CHAPTER 4

Conclusions and Future Work

In the recent years threaded run-to-run (RtR) control algorithms have experienced several drawbacks when applied to high-mix products such as in Application Specific Integrated Circuits (ASIC) foundries. The variations in the process are a function of the product being manufactured as well as the tool being used. The presence of semiconductor layers increases the number of times the lithography process must be repeated. Successive layers having different patterns must be exposed using different reticles/masks in order to maximize tool utilization.

The objectives of this research were to develop a set of methodologies for evaluation and extension of threaded control applied to overlay. This project defines methods to quantify the efficacy of threaded controls, finds the drawbacks of threaded control under production of high mix of semiconductors and suggests extensions and alternatives to improve threaded control.

To evaluate the performance of threaded control, extensive simulations were performed in MATLAB. The effects of noise, disturbances, sampling and delays on the control and estimation performance of threaded controller were studied through these simulations. From the simulations optimal tuning factor for EWMA run-to-run control was found to be in between the range of 0.2-0.4. The results match with the actual value of filter weight used in the semiconductor manufacturing business. The performance of the control system is greatly affected by noise, disturbances, delays and sampling. The

simulations showed that the performance degrades due to step disturbance and varies with the step size. A large step size worsens the control performance. Subsequent steps of large magnitude tend to be more dangerous than a single large disturbance. Noise causes significant rise in the MSE and hence proves to be an important factor determining the accuracy of the control. Process delays and metrology delays also hamper the performance due to the delays in the feedback loop.

The idea of ANOVA (Analysis of Variance) for deciding the contexts in the thread definitions was introduced. ANOVA needs experimental data for extensive analysis. ANOVA needs to be carried on a data set which has all the combinations of found variable and corresponding responses for the dependent variable. It was studied that MANOVA (Multivariate ANOVA) gives more stable results when there are more than one response variable. These problems are recommended for future work.

The effect of sampling was studied using simulations for random and uniform sampling. It was observed MSE decreases with increase in the random sampling frequency and the MSE was lowest at 100% sampling. Although this was the expected result, the motive of sampling is to not sample every run. Random sampling can yield really bad results if the data selected randomly are bunched at the early runs, in the middle or the runs at the end (random but not evenly spaced). It was observed that random sampling shows consistent results only when the simulation is run multiple times at a particular sampling frequency and averaged, since every time the sample space was different. Although uniform sampling easy for scheduling, choosing the appropriate sampling strategy is a key concern since the metrology cost depends the number of samples measured. With higher uniform sampling frequency, the number of total samples

decreases but it also degrades the performance. It was observed that a dynamic sampling algorithm based on these sampling strategies would give the best performance while keeping the number of measurements low. The idea of dynamic sampling was discussed in the Chapter 3. The algorithm needs to be tested with a variety of data sets to check if it gives consistent results. Also, there are certain rules considered for sampling in order to optimize the tolerance and obtain lowest sample size (one example of such a rule is to force a measurement if there has not been a single measurement in the past N runs). The solution to this problem would be obtained by mixed integer programming [21]. This problem is suggested for further test and study for the future work.

The idea of thread recombination was introduced in the last chapter. Simulations were run for the threads formed with the simulated 5000 run data set. However, a simulation with real data is suggested for future work.

Based on the results obtained, several ideas to extend threaded control by reducing overall number of threads, by improving thread definitions and combinations have been introduced. Future work also includes implementing the extensions to threaded control suggested in this work in real production data, comparing the results without the use of those methods, and building new alternatives to threaded control. The non-threaded techniques JADE, RLS and Random walk model should be compared with the extended threaded technique when applied to real production data under different scenarios like disturbance, noise, low running threads.

References

- [1] R. Schaller, "Moore's law: past, present and future," *Spectrum, IEEE*, vol. 34, no. 6, pp. 52-59, 1997.
- [2] J. D. Plummer, M. D. Deal, and P. B. Griffin, *Silicon VLSI technology: fundamentals, practice and modeling*. Prentice Hall, 2000.
- [3] C. A. Bode, "Run-to-Run Control of Overlay and Linewidth in Semiconductor Manufacturing," 2001.
- [4] J. Stuber, F. Pagette, and S. Tang, "Device dependent run-to-run control of transistor critical dimension by manipulating photolithography exposure settings," in *AEC/APC Symp. XII vol. I, Int. SEMATECH, 2000*.
- [5] C. A. Bode, B. S. Ko, and T. F. Edgar, "Run-to-run control and performance monitoring of overlay in semiconductor manufacturing," *Control Engineering Practice*, vol. 12, no. 7, pp. 893-900, Jul. 2004.
- [6] E. Sachs, A. Hu, and A. Ingolfsson, "Run by run process control: combining SPC and feedback control," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 8, no. 1, pp. 26-43, 1995.
- [7] J. Moyne, E. D. Castillo, and A. M. Hurwitz, *Run-to-run control in semiconductor manufacturing*. CRC Press, 2001.
- [8] M. L. Miller, "Impact of multi-product and -process manufacturing on run-to-run control," in *Process, Equipment, and Materials Control in Integrated Circuit Manufacturing III*, vol. 3213, pp. 138-146, 1997.
- [9] W. Campbell, S. Firth, A. Toprac, and T. Edgar, "A comparison of run-to-run

control algorithms,” in *American Control Conference, 2002. Proceedings of the 2002*, vol. 3, pp. 2150-2155 vol.3, 2002.

- [10] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control*, 2nd ed. Wiley, 2003.
- [11] S. Firth, W. Campbell, A. Toprac, and T. Edgar, “Just-in-time adaptive disturbance estimation for run-to-run control of semiconductor processes,” *Semiconductor Manufacturing, IEEE Transactions on*, vol. 19, no. 3, pp. 298-315, 2006.
- [12] A. V. Prabhu and T. F. Edgar, “A new state estimation method for high-mix semiconductor manufacturing processes,” *Journal of Process Control*, vol. 19, no. 7, pp. 1149-1161, Jul. 2009.
- [13] A. V. Prabhu, *Performance Monitoring of Run-to-Run Control Systems Used in Semiconductor Manufacturing*. University of Texas Libraries, 2008.
- [14] Ming Tham, “Internal Model Control,” *Chemical and Process Engineering*, University of Newcastle upon Tyne, 2002.
- [15] B. Francis and W. Wonham, “The internal model principle of control theory,” *Automatica*, vol. 12, no. 5, pp. 457-465, Sep. 1976.
- [16] C. Bode, J. Wang, Q. He, and T. Edgar, “Run-to-run control and state estimation in high-mix semiconductor manufacturing,” *Annual Reviews in Control*, vol. 31, no. 2, pp. 241-253, 2007.
- [17] S. K. Firth, “Just-in-Time Adaptive Disturbance Estimation for Run-to-Run Control in Semiconductor Processes,” 2002.
- [18] M. Braun, S. Jenkins, and N. Patel, “A comparison of supervisory control algorithms for tool/process disturbance tracking,” in *American Control Conference*,

2003. *Proceedings of the 2003*, vol. 3, pp. 2626-2631 vol.3, 2003.

- [19] J. Wang, Q. Peter He, and T. F. Edgar, "State estimation in high-mix semiconductor manufacturing," *Journal of Process Control*, vol. 19, no. 3, pp. 443-456, Mar. 2009.
- [20] J. Wang, Q. Peter He, and S. Joe Qin, "Stability analysis and optimal tuning of EWMA controllers - Gain adaptation vs. intercept adaptation," *Journal of Process Control*, vol. 20, no. 2, pp. 134-142, Feb. 2010.
- [21] R. Good and M. Purdy, "An MILP Approach to Wafer Sampling and Selection," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 20, no. 4, pp. 400-407, 2007.
- [22] S.A. Harrison, M.W. Braun, and T.F. Edgar, "An evaluation of the effects of product mix and metrology delay on the performance of segregated versus threaded EWMA control," in *AEC/APC Symposium, 2003*.
- [23] Y. Zheng, Q. Lin, D. S. Wang, S. Jang, and K. Hui, "Stability and performance analysis of mixed product run-to-run control," *Journal of Process Control*, vol. 16, no. 5, pp. 431-443, Jun. 2006.
- [24] A. J. Toprac and Y. Wang, "Advanced method for run-to-run control of photolithography overlay registration in high-mix semiconductor production," in *Data Analysis and Modeling for Process Control II*, vol. 5755, pp. 1-8, 2005.
- [25] X. Wan et al., "Overlay advanced process control for foundry application," in *Metrology, Inspection, and Process Control for Microlithography XVIII*, vol. 5375, pp. 735-743, 2004.
- [26] M. Ma, C. Chang, D. S. Wong, and S. Jang, "Identification of tool and product effects in a mixed product and parallel tool environment," *Journal of Process*

Control, vol. 19, no. 4, pp. 591-603, Apr. 2009.

[27] O. Vanli, N. Patel, M. Janakiram, and E. Del Castillo, "Model Context Selection for Run-to-Run Control," *Semiconductor Manufacturing, IEEE Transactions on*, vol. 20, no. 4, pp. 506-516, 2007.

[28] H. J. Lee, *Advanced Process Control and Optimal Sampling in Semiconductor Manufacturing*. University of Texas Libraries, 2008.

Vita

Ninad N. Patwardhan was born and brought up in the city of Pune, India. He completed his Bachelor's in Engineering from Modern College of Engineering affiliated to University of Pune, India. After completion of his Bachelor's degree he worked for 2 years in the Automation and Process Control Industry, as a Systems Engineer in Emerson Export Engineering Centre, India and as a Project Engineer in Spectrum Automation and Controls, India. He joined The University of Texas at Austin as a Master's student in Electrical Engineering Department in fall 2008.

Email: ninad.patwardhan@gmail.com

This thesis was typed by the author.