

Copyright  
by  
Bhalinder Singh Gill  
2011

The Dissertation Committee for Bhalinder Singh Gill  
certifies that this is the approved version of the following dissertation:

**Development of Virtual Metrology in Semiconductor  
Manufacturing**

Committee:

---

Thomas F. Edgar, Supervisor

---

Dragan Djurdjanovic

---

Venkat Ganesan

---

Glenn Y. Masada

---

Grant Willson

---

John D. Stuber

**Development of Virtual Metrology in Semiconductor  
Manufacturing**

by

**Bhalinder Singh Gill, B.Tech.**

**DISSERTATION**

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

**DOCTOR OF PHILOSOPHY**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2011

Dedicated to my parents.

## Acknowledgments

First, I would like to thank Dr. Edgar for providing me an opportunity to pursue my Ph.D. degree at the University of Texas at Austin under his guidance. Besides sharing his technical expertise, he helped me to become a better engineer and a better technical writer. I sincerely feel that working with him has improved my time management and multi-tasking capabilities. All the discussions with him came out to be very fruitful and always ended on a cheerful note.

I would also wish to thank Dr. Djurdjanovic, Dr. Ganesan, Dr. Masada, Dr. Willson, and Dr. Stuber for serving as my committee members. In addition, I thank all the professors at the University of Texas at Austin who enhanced my knowledge of the process control systems and semiconductor manufacturing.

This dissertation would not have been possible without the support of Texas Instruments, Inc. (TI) and the initiative taken by Dr. John Stuber. I really appreciate Dr. Stuber's effort to carry out research with the universities, which provide the students an exposure to the real world manufacturing. He has always helped me to extract data from the databases at TI and has provided valuable feedback on the technical reports. I would like to thank Mike Bowen for the fab tour and for pulling out data from the plasma etch sys-

tem. Maja Imamovic also assisted me in accessing data for one of the process excursions.

I extend my thanks to the brilliant and talented professors at the Indian Institute of Technology, Guwahati for equipping me with the fundamental concepts of chemical engineering. I would like to thank my undergraduate research supervisor Dr. Prabirkumar Saha for giving me an opportunity to work with him and providing me an exposure to research in the field of process control systems. I would also like to thank Achim Küpper and Dr. Sebastian Engell at process control laboratory, Universität Dortmund, Germany for providing me an internship in process controls area. Both these research experiences triggered a desire in me to pursue doctoral studies in this area.

I was also fortunate enough to study together with many exceptional graduate students and visiting scholars including: Hyung Lee, Amogh Prabhu, Yang Zhang, Sidharth Abrol, Kye Hyun Baek, Pedro Han, Sepideh Ziaii, Ninad Patwardhan, Ela Joag, Ivan Castillo, Ben Spivey, Doug French, Blake Parkinson, Xiaojing Jiang, Anh Nguyen, Ramiro Palma, Kriti Kapoor, Kody Powell, Jong Kim, Vinay Bhardwaj, Wesley Cole, and Akshay Sriprasad. This is an excellent group of people to be associated with from both academia and personal perspective. I would like to thank Hyung, Amogh, and Sidharth for helping me to make a good career choice by sharing their job search experiences. I certainly owe a lot to Ivan for the innumerable technical discussions we had and his perpetual willingness to help me with the fault detection approaches and latex programs for writing the dissertation.

My stay in Austin over the last four years has been made enjoyable by many interesting and cheerful friends. I feel lucky to have met and shared great times with Akshay, Al, Aldo, Alok, Amit, Anne, Anish, Anush, Apurv, Aruna, Ashley, Chetan, Cynthia, Deepjyoti, Deona, Divya, Gautam, Geetha, Guneet, Guntej, Harpreet, Harsh, Harshdeep, Hui, Hyojin, Jagdeep, Jiajia, Jin, Julie, Karthik, Katie, Kiran, Kriti, Kyra, Lynne, Mansi, Ming, Namrata, Nandita, Nani, Neha, Nikhil, Nikita, Nupur, Poolkeshi, Pradeep, Pranav Bhandarkar, Pranav Karve, Preeti, Ramya, Ravneet, Rebecca, Rekha, Ripudaman, RJ, Sahana, Sahil, Sai, Sarabjot, Sean, Sharayu, Shatam, Shira, Sindhu, Som, Sravnthi, Sumala, Tai, Thu, Tina, Vidya, Vinay, Yetendra, and Ying. I will miss you all and I will miss living in Austin.

Achieving this goal would not have been possible without the support of my parents. They always inspire and encourage me to reach high goals. The support of my sister and my brother-in-law keeps me going through the ups and downs of life. I thank God for blessing me with such a wonderful and loving family.

# Development of Virtual Metrology in Semiconductor Manufacturing

Publication No. \_\_\_\_\_

Bhalinder Singh Gill, Ph.D.  
The University of Texas at Austin, 2011

Supervisor: Thomas F. Edgar

Virtual Metrology (VM) predicts end-of-batch properties (metrology data) from measurable input data composed of pre-process metrology and fault detection and classification (FDC) system outputs. This dissertation aims at moving a step closer to the realization of VM in semiconductor manufacturing by providing solutions to the challenges that present VM technology faces. First, various VM methods are introduced and compared in terms of prediction accuracy using four industrial datasets collected from a plasma etch system at Texas Instruments, Inc.. Kalman filter estimation is employed in a novel way to serve as a VM model for predicting outputs of a static process. Recursive PLS regression (R-PLSR) and Kalman filter show the best prediction results as they update the model whenever new measurements are available. Next, two PLS variants (PLS with EWMA mean update and recursive PLS) are proposed as robust VM algorithms that can predict process outputs fairly accurately in the presence of unexpected process drifts and noise. The obtained results

reinforce VM technology by suggesting appropriate prediction methods when unexpected process changes occur.

For a successful implementation of VM, the data entering the VM model needs to be free from faults. Fault-free (reconstructed) data are obtained by performing fault detection, fault identification, and fault reconstruction. A novel fault detection method based on statistics pattern analysis (SPA) is presented. The SPA method provides better fault detection performance for different types of faults as compared to the MPCA-based methods. Next, three well-known fault identification methods present in literature are implemented. An equation that relates the RBC with the SVI is derived. The contribution plot method identifies a smaller number of faults correctly as compared to the RBC and the SVI methods. Fairly good estimates of the fault magnitude are obtained when the faults are identified correctly.

An approach that combines physical measurements with the VM estimates to develop a more robust approach than using VM alone is presented. EWMA-R2R control is implemented using three well-known sampling methods in order to demonstrate the superior performance of two novel control schemes: B-EWMA R2R control and VM-assisted EWMA-R2R control. A new reliance index, which is attractive from a mathematical and practical point of view, is proposed. The VM-assisted EWMA-R2R control yields the best control results among the control schemes employed in this study. The simulation results demonstrate that VM has the potential to reduce measurement costs significantly while promising better process control.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>viii</b>
<b>List of Tables</b>	<b>xiv</b>
<b>List of Figures</b>	<b>xvi</b>
<b>Chapter 1 Introduction</b>	<b>1</b>
1.1 Metrology in Semiconductor Manufacturing . . . . .	1
1.2 Virtual Metrology (VM) . . . . .	2
1.3 Challenges With The Current VM Technology . . . . .	3
1.4 Research Objectives . . . . .	5
<b>Chapter 2 Virtual Metrology Methods and Their Application to a Plasma Etch Process</b>	<b>14</b>
2.1 Introduction . . . . .	14
2.2 Industrial Datasets . . . . .	17
2.3 Methods . . . . .	21
2.3.1 Multiple Linear Regression . . . . .	22
2.3.2 Principal Component Regression . . . . .	22
2.3.3 Partial Least Squares Regression . . . . .	24
2.3.4 Recursive Partial Least Squares Regression . . . . .	25
2.3.5 Time Series Analysis . . . . .	25
2.3.6 Kalman Filter Estimation . . . . .	26
2.4 Results and Discussion . . . . .	28
2.4.1 Lot-level etch rate predictions . . . . .	29
2.4.2 Wafer-level sheet resistance predictions . . . . .	31
2.4.3 Wafer-level critical dimension (CD) predictions . . . . .	37
2.5 Conclusions . . . . .	44

<b>Chapter 3</b>	<b>Tailored PLS Algorithms for Handling Unexpected Drifts and Noise in Virtual Metrology</b>	<b>47</b>
3.1	Introduction . . . . .	48
3.2	Process Model . . . . .	49
3.3	Design of Experiments . . . . .	51
3.4	Prediction Methods . . . . .	52
3.4.1	PLS with EWMA mean update . . . . .	53
3.4.2	Recursive PLS . . . . .	55
3.5	Results and Discussion . . . . .	56
3.6	Conclusions . . . . .	65
<b>Chapter 4</b>	<b>Detection of Faults in Virtual Metrology Sensors Using MPCA</b>	<b>66</b>
4.1	Introduction . . . . .	67
4.2	Fault Detection Approaches . . . . .	68
4.2.1	Principal component analysis (PCA) . . . . .	71
4.2.2	Multiway principal component analysis (MPCA) . . . . .	78
4.3	Details of the Benchmark Dataset . . . . .	81
4.4	Fault Detection Using MPCA . . . . .	84
4.4.1	Fault detection results using MPCA . . . . .	86
4.4.2	Effect of fault magnitude/size on fault detection performance . . . . .	89
4.4.3	Effect of confidence level ( $\alpha$ ) on fault detection performance	90
4.4.4	False alarms . . . . .	91
4.4.5	Limitations of MPCA . . . . .	93
4.4.6	Fault detection using MPCA with EWMA filtering of residuals . . . . .	97
4.5	Conclusions . . . . .	104
<b>Chapter 5</b>	<b>Detection, Identification, and Reconstruction of Faults in Virtual Metrology Sensors</b>	<b>107</b>
5.1	Introduction . . . . .	107
5.2	Fault Detection Using Statistics Pattern Analysis (SPA) . . . . .	109
5.2.1	Motivation . . . . .	109

5.2.2	SPA framework . . . . .	111
5.2.3	Fault detection results using SPA . . . . .	114
5.3	Fault Identification and Reconstruction . . . . .	127
5.3.1	Contribution plot approach . . . . .	128
5.3.2	Reconstruction-based contribution (RBC) method . . . . .	129
5.3.3	Sensor validity index (SVI) method . . . . .	133
5.3.4	Fault identification results . . . . .	135
5.3.5	Fault reconstruction . . . . .	143
5.4	Conclusions . . . . .	145

**Chapter 6 Improvements in Run-to-Run Process Control Using Virtual Metrology 150**

6.1	Introduction . . . . .	151
6.2	Sampling Methods . . . . .	155
6.2.1	Uniform sampling . . . . .	155
6.2.2	Random sampling . . . . .	156
6.2.3	Dynamic sampling . . . . .	158
6.3	Run-to-Run Control . . . . .	166
6.3.1	Exponentially-Weighted-Moving-Average R2R control . . . . .	167
6.4	Models . . . . .	171
6.4.1	SISO model . . . . .	171
6.4.2	MIMO model . . . . .	172
6.5	Results and Discussion . . . . .	174
6.5.1	EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling for SISO model . . . . .	174
6.5.2	Effect of EWMA forgetting factor . . . . .	180
6.5.3	EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling for MIMO model . . . . .	180
6.5.4	Improvement in EWMA-R2R control performance using Bayesian update of EWMA forgetting factor . . . . .	188
6.5.5	VM-assisted EWMA-R2R control . . . . .	190
6.5.6	New approach for calculating reliance index of VM estimates . . . . .	196
6.6	Conclusions . . . . .	204

<b>Chapter 7 Summary and Future Work</b>	<b>209</b>
7.1 Summary of Contributions . . . . .	209
7.2 Recommendations for Future Work . . . . .	218
<b>Bibliography</b>	<b>222</b>
<b>Vita</b>	<b>246</b>

## List of Tables

2.1	Comparison of VM methods for the lot-level prediction of etch rate from OES data. . . . .	29
2.2	Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2. . . . .	34
2.3	Signal-to-noise ratio of the measured sheet resistance and etch rate for Dataset 2. . . . .	37
3.1	Summary of the key features of the prediction methods employed in this work. . . . .	56
3.2	Summary of the prediction performance of the PLS methods for the output $y_1$ . Recursive PLS provides the best predictions for all types of drifts. . . . .	62
3.3	Summary of the prediction performance of the PLS methods for the output $y_2$ . Recursive PLS provides the best predictions for all types of drifts. . . . .	62
3.4	Mean squared error (MSE) values for the predictions of the output variable $y_1$ for different sizes of measurement noise. . .	64
3.5	Mean squared error (MSE) values for the predictions of the output variable $y_2$ for different sizes of measurement noise. . .	64
4.1	Process variables used for fault detection in an Aluminium stack etch process. . . . .	82
4.2	Influence of the fault magnitude/size on the fault detection performance of MPCA (19 sensor faults were introduced). The fault magnitude is expressed in terms of the percentage of the mean value of process variables. Faults with larger magnitudes are detected more easily than the faults with relatively smaller magnitudes. . . . .	90
4.3	Influence of the confidence level ( $\alpha$ ) on the fault detection performance of MPCA (19 sensor faults were introduced). A higher confidence level will lead to detection of lesser number of faults because of higher control limits, but these detections are very likely to be the actual faults, not false alarms. . . . .	91

5.1	Comparison of three fault detection methods studied in this work: MPCA, MPCA with EWMA filtering of residuals, and SPA in terms of number of false alarms raised. . . . .	121
5.2	Comparison of three fault detection methods studied in this work: MPCA, MPCA with EWMA filtering of residuals, and SPA in terms of number of different types of faults detected. A total of 74 faults comprising of 19 mean faults (one for each process variable), 19 variance faults, 18 skewness faults, and 18 kurtosis faults were introduced. . . . .	125
5.3	Comparison of three fault identification methods studied in this work: contribution plots, RBC, and SVI in terms of number of correctly identified faults for different fault types. A total of 74 faults comprising of 19 mean faults (one for each process variable), 19 variance faults, 18 skewness faults, and 18 kurtosis faults were introduced. . . . .	141
5.4	The estimates of the fault magnitudes for 19 mean faults. Fairly accurate estimates are obtained when faults are correctly identified. . . . .	144
6.1	EWMA-R2R controller performance for SISO model using different sampling methods . . . . .	178
6.2	EWMA-R2R controller performance for MIMO model using different sampling methods . . . . .	184
6.3	Controller performance for SISO model using different sampling methods . . . . .	190
6.4	Controller performance for MIMO model using different sampling methods . . . . .	190
6.5	Controller performance for MIMO model using different sampling methods . . . . .	204

## List of Figures

1.1	The working of VM-assisted EWMA-R2R control . . . . .	3
1.2	Summary of the approach adopted in Chapters 4 and 5. . . . .	8
2.1	A thin film made up of material of resistivity $\rho$ . . . . .	18
2.2	Variation of 18 OES signals with time for a sample wafer. Each curve corresponds to an OES signal. There is not much variation in the OES signal values; the mean values of the signals are sufficient metrics to build a regression model. . . . .	20
2.3	Comparison of VM methods for the lot-level prediction of etch rate from OES data for Dataset 2 (training). . . . .	32
2.4	Comparison of VM methods for the lot-level prediction of etch rate from OES data for Dataset 2 (validation). . . . .	33
2.5	Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2 (training). . . . .	35
2.6	Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2 (validation). . . . .	36
2.7	CD values predicted by PLS model are fairly close to the measured CDs and have a Mean Absolute Percentage Error (MAPE) of 1.5159. Most of the outliers come from first two wafers of the lots. For the gate etch process under investigation, target CD was 81 nm. . . . .	40
2.8	A positive correlation exists between the measured CDs and the predicted CDs. A $R^2$ value of 0.4324 was obtained. The equation of linear fit is $y = x + 8.78E - 13$ . . . . .	41
2.9	PLS model coefficients for 38 input process variables. Variable numbers 4, 6, 9, 11, 13, 27, 29, 35, and 37 have significantly larger coefficients than the rest of the process variables. Most likely, these variables are causing the undesired CD values for the first two wafers in the lots under consideration. . . . .	42
2.10	Magnitude of variable number 9 for the first wafer, second wafer, and a normal wafer in each of 41 lots under consideration. For all the lots, the first wafer has a very different value than a normal wafer. Therefore, variable number 9 is one of the variables that cause a different CD value for the first wafer as compared to other wafers in the lot. . . . .	43

3.1	Operating points for PLS model building based on a full-factorial design. . . . .	52
3.2	Implementation of PLS to predict the values of process outputs. . . . .	53
3.3	Selecting the data for updating the PLS model. . . . .	55
3.4	Predictions made by the PLS methods in the presence of a linear rise. . . . .	57
3.5	Predictions made by the PLS methods in the presence of a ramp change. . . . .	60
3.6	Predictions made by the PLS methods in the presence of a linear drift starting arbitrarily during the process. . . . .	61
3.7	PLS with EWMA mean update provides better predictions than recursive PLS in the presence of large measurement noise. . . . .	63
4.1	Characteristics of semiconductor manufacturing processes. This figure shows unequal batch lengths and unsynchronized/misaligned and multimodal trajectories of a process variable, pressure, for two etch steps for four processed wafers. . . . .	79
4.2	Data preprocessing of three-dimensional data collected from a semiconductor manufacturing process to obtain a two-dimensional matrix that can be analyzed using PCA. This procedure is known as multiway PCA or MPCA. . . . .	81
4.3	Processing times for the normal (fault-free) wafers in the benchmark dataset. These times represent the duration of the main etch and over etch steps only. The data need to be preprocessed before detecting faults using MPCA. . . . .	85
4.4	Fault detection using MPCA. 14 faults present in the mean values of the process variables are detected. . . . .	87
4.5	Fault detection using MPCA. Zoomed-in view of Figure 4.4. . . . .	87
4.6	Fault detection using MPCA. Zoomed-in view of Figure 4.5. . . . .	88
4.7	SPE indices for 107 normal wafers using MPCA. Twelve false alarms were observed, i.e., the SPE indices were found to be more than the SPE control limit for twelve normal wafers. . . . .	92
4.8	Histograms of two process variables from the benchmark dataset measured at a fixed time stamp for all the normal wafers (a) $BCl_3$ flow rate; (b) $Cl_2$ flow rate. Clearly the distributions of these process variables are not close to a Gaussian distribution. . . . .	96
4.9	Filtered residuals obtained by setting the EWMA forgetting factor ( $\Gamma$ ) equal to 0.1 (a) Histogram (b) q-q plot. The filtered residuals conform better to the normal distribution than the unfiltered residuals (generated from a uniform distribution). . . . .	99

4.10	Filtered residuals obtained by setting the EWMA forgetting factor ( $\Gamma$ ) equal to 0.9 (a) Histogram (b) q-q plot. EWMA filtering fails to improve the normality of the residuals. The filtered residuals resemble the distribution of the unfiltered residuals (uniform distribution) more than the normal distribution. . . .	101
4.11	$S\bar{P}E$ indices for 107 normal wafers using MPCA with EWMA filtering of residuals using a forgetting factor ( $\gamma$ ) value of 0.1. The number of false alarms is reduced to ten; twelve false alarms were observed when no filtering was used. It is evident that all the false alarms occur at the start of filtering and no false alarms are generated once $S\bar{P}E$ falls below the $S\bar{P}E$ control limit. . .	103
5.1	Summary of the approach adopted in this chapter. . . . .	108
5.2	Illustration of the SPA framework. (a) Original batches of unequal length; (b) Statistics pattern (SP) generation; (c) Fault detection using dissimilarity quantification. . . . .	111
5.3	Histograms and q-q plots of the means for two process variables from the benchmark dataset for all the normal wafers (a and b) $BCl_3$ flow rate; (c and d) $Cl_2$ flow rate. Clearly, the distributions of the means are quite close to a Gaussian distribution. .	115
5.4	Histograms and q-q plots of the variances for two process variables from the benchmark dataset for all the normal wafers (a and b) $BCl_3$ flow rate; (c and d) $Cl_2$ flow rate. Clearly, the distributions of the variances are quite close to a Gaussian distribution. . . . .	116
5.5	Histograms and q-q plots of the skewnesses for two process variables from the benchmark dataset for all the normal wafers (a and b) $BCl_3$ flow rate; (c and d) $Cl_2$ flow rate. Clearly, the distribution of the skewnesses for $BCl_3$ flow rate is quite close to a Gaussian distribution. . . . .	117
5.6	Histograms and q-q plots of the kurtoses for two process variables from the benchmark dataset for all the normal wafers (a and b) $BCl_3$ flow rate; (c and d) $Cl_2$ flow rate. The distributions of the kurtoses are not very close to a Gaussian distribution as the Central Limit Theorem holds weakly for higher-order statistics. . . . .	118
5.7	Fault detection using SPA raises only one false alarm while monitoring data for 107 normal wafers. . . . .	120
5.8	Fault detection using SPA. All 19 faults present in the mean values of the process variables are detected. . . . .	123
5.9	Fault detection using SPA. Zoomed-in view of Figure 5.8. . . .	124

5.10	Fault detection using SPA. Zoomed-in view of Figure 5.9. . . .	124
5.11	Fault identification for a fault in the mean of process variable RF power. SP statistic number 11, which corresponds to the mean of RF power, shows the largest contribution indicating correct identification. . . . .	136
5.12	Fault identification for a fault in the variance of process variable $BCl_3$ flow rate. SP statistic number 20, which corresponds to the variance of process variable $BCl_3$ flow rate, shows the largest contribution indicating correct identification. . . . .	137
5.13	Fault identification for a fault in the skewness of process variable TCP tuner. SP statistic number 221, which corresponds to the skewness of process variable TCP tuner, shows the largest contribution indicating correct identification. . . . .	138
5.14	Fault identification for a fault in the kurtosis of process variable TCP impedance. SP statistic number 241, which corresponds to the kurtosis of process variable TCP impedance, shows the largest contribution indicating correct identification. . . . .	139
6.1	Variation of mean squared error (MSE) with sampling interval for uniform sampling. As the sampling interval increases, measurements are done less frequently causing sluggish adaptation of R2R controller to the process drift. . . . .	157
6.2	Normal data and the shifted data . . . . .	161
6.3	Normal data and shifted data to demonstrate how Bayes' theorem is used in dynamic sampling . . . . .	164
6.4	Implementation of R2R control on a linear process . . . . .	170
6.5	EWMA-R2R Control Using Different Sampling Methods for SISO Model . . . . .	175
6.6	Dynamic sampling results for the SISO process with $\lambda = 0.3$ .	179
6.7	Dynamic sampling results for the SISO process with $\lambda = 0.1$ .	181
6.8	Dynamic sampling results for the SISO process with $\lambda = 0.7$ .	182
6.9	EWMA-R2R Control Using Different Sampling Methods for MIMO Model . . . . .	183
6.10	Dynamic sampling results for $b_1$ , $u_1$ , and $y_1$ of MIMO process with $\lambda = 0.3$ . . . . .	185
6.11	Dynamic sampling results for $b_2$ , $u_2$ , and $y_2$ of MIMO process with $\lambda = 0.3$ . . . . .	186

6.12	Comparison of different control schemes for SISO model. B-EWMA-R2R control using dynamic sampling provides the best control performance . . . . .	191
6.13	Comparison of different control schemes for MIMO model. B-EWMA-R2R control using dynamic sampling provides the best control performance . . . . .	192
6.14	The working of VM-assisted EWMA-R2R control . . . . .	193
6.15	Reliance index is calculated as the overlapping area between the reference distribution and the VM distribution . . . . .	199
6.16	Simulation results showing the superior performance of VM-assisted EWMA-R2R control as compared to other control schemes.	201
6.17	Reliance indices for the VM estimates of two process outputs. A threshold value of 0.7 was used in this simulation. . . . .	202

# Chapter 1

## Introduction

### 1.1 Metrology in Semiconductor Manufacturing

In semiconductor manufacturing, a wafer undergoes hundreds of different steps to yield the final product. It is imperative to have a good knowledge of the wafer characteristics at the end of each of these processes. Otherwise, faulty wafers are detected too late, which leads to the loss of resources. Offline metrology using technologically advanced tools (e.g., ellipsometer, atomic force microscope (AFM)) ensures that the process is on target by measuring the critical dimensions of the post-process wafers.

In the semiconductor industry, the feature sizes have been shrinking considerably over the past decade. Also, the rising demand for the products has led to a tremendous increase in the throughput of fabs. It becomes impractical to do offline metrology after each processing step for every wafer because of high cost and time delay issues. The time spent in transporting wafers from processing tool to measurement tool, measuring critical dimensions and sending it back to the next processing tool can hamper the throughput of a fabrication facility.

Broadly speaking, the current sampling strategy of a semiconductor

fabrication facility is to sample one wafer per lot (a lot consists of about 25 wafers); i.e., a sampling rate of only 4%. In order to control the manufacturing processes better, more frequent sampling is required. Hence, high resolution and fast metrology equipment are needed to meet these challenges, which lead to the concept of Virtual metrology (VM).

## 1.2 Virtual Metrology (VM)

VM is a potential solution for providing low-cost, fast, and accurate metrology information and has many applications in lithography [28], etch [105], and deposition processes [38]. VM predicts end-of-batch properties (metrology data) from measurable input parameters. These input parameters could consist of pre-process metrology and fault detection and classification (FDC) system outputs. The predictions for end-of-batch properties are made by supplying the input parameter information to a model. This model could be physically-based (first principles), empirically determined, or a combination of physical and empirical.

One of the major attractive features of VM is that it utilizes FDC data that are already being used to detect and classify faults in the manufacturing processes. An accurate prediction by VM would be an added value to the FDC system as there is no need to install any new sensors. Almost all the critical equipment in a semiconductor fabrication facility operate in parallel with a FDC system; so once an accurate model is developed, it can be coupled with the manufacturing tools without making major changes in operations.

### 1.3 Challenges With The Current VM Technology

Researchers have proposed several approaches to develop VM for semiconductor manufacturing processes, but there is no standard and generalized method as of now. Hung and Lin [52] adopted a radial basis function neural network to construct the virtual metrology model. They studied a chemical vapor deposition (CVD) process and found that the proposed model had several advantages over the one based on back-propagation neural network. Cheng et al. [52] proposed dual phase virtual metrology to consider both promptness and accuracy. They also calculated the accompanying reliance index (RI) and global similarity index (GSI) [16] and presented an illustrative example involving fifth-generation thin-film-transistor liquid crystal display (TFT-LCD) chemical vapor deposition equipment. The authors used multiple regression, neural networks, and time series algorithms to build the conjecture models.

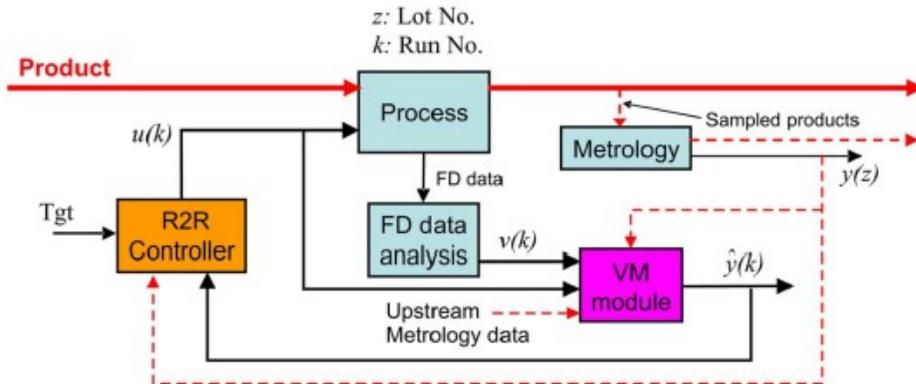


Figure 1.1: The working of VM-assisted EWMA-R2R control

Khan et al. [62] proposed a VM approach to implement wafer-to-wafer

(run-to-run, R2R) control on a factory level, which is illustrated in Figure 1.1. The authors utilized the FDC data along with the process inputs and upstream metrology data to predict outputs using the VM module. Whenever new metrology data were available, the VM module was updated. The authors presented solutions for the issues that will arise when VM becomes an integral part of the factory-wide advanced process control solution for the wafer-to-wafer control. Khan et al. also developed a recursive moving-window approach using partial least squares (PLS) to update the VM module whenever new metrology data become available [41, 63]. Pan and Tai [91] used multiple linear regression and PLS regression to build prediction models for the film thickness and critical dimensions for the online production of semiconductor manufacturing. A comparison of various popular VM algorithms including back-propagation neural networks (BPNN), simple recurrent neural networks (SRNN), and multiple regression (MR) was reported by Su et al. [120]. The process under investigation was a fifth-generation thin-film-transistor liquid crystal display (TFT-LCD) chemical vapor deposition, same as in Cheng et al. [16].

As semiconductor devices are continually shrinking and critical dimension (CD) approaches 32 nm, the control of gate CD, which depends on plasma etch process, is a top priority in advanced semiconductor manufacturing [91]. But most of the work in the literature until now on VM aims at predicting the outputs in a CVD process [16, 52, 120], which is more easily understood than the plasma etch process. Due to complex physical and chemical phenomena, it

has been difficult to accurately predict plasma processes [64]. However, some authors have explored VM for etching process as summarized below.

Cheng et al. [16] presented an illustrative example involving 300 mm semiconductor foundry etching equipment, but no comparison of the predicted values with the actual measurements was reported. Zeng and Spanos [149] applied six statistical techniques to predict etch bias for a plasma etch process, but their results suffer from lack of accuracy. Recently, a VM system was developed for an etching process based on various data mining techniques [59]. The authors used the actual metrology values from a preceding metrology process and equipment sensor data to make the predictions. As stated in the conclusion section of their work, this kind of approach would restrict the usability of the VM system because metrology is not done at the end of each process in semiconductor manufacturing. Some other works [75, 91] also suffer from the same limitation. Further research should be conducted only using the sensor data to build an accurate and reliable VM system.

## **1.4 Research Objectives**

The present VM technology faces many challenges that must be addressed before it can be put into practice. Accuracy of the VM predictions is one of the most important requirements as it directly affects the quality of the products. Process noise, measurement noise, process drifts, and process shifts are the main barriers in the development of an accurate VM model. Other issues, for example, degradation of the quality of the VM predictions

because of the faults present in the input data, and optimal sampling of the wafers to be measured will arise when VM is put into practice. This proposal aims at providing solutions to these challenges and moving a step closer to the realization of VM in semiconductor manufacturing.

First, we introduce various methods used for implementing Virtual Metrology (VM) and compare them in terms of prediction accuracy using industrial data collected from a plasma etch process in Chapter 2. Specifically, multiple linear regression (MLR), principal component regression (PCR), partial least squares regression (PLSR), recursive partial least squares regression (R-PLSR), time series analysis, and Kalman filter estimation are compared in terms of prediction accuracy using four industrial datasets collected from a fabrication facility at Texas Instruments, Inc.. Predictions are made for etch rate, sheet resistance, and critical dimension (CD) using the optical emission spectroscopy (OES) data and 38 other process variables. Kalman filter estimation is employed in a novel way to serve as a VM model for predicting outputs of a static process.

Unexpected process drifts and noise can severely hamper the accuracy of predictions made by popular VM algorithms such as PCA, PLS [63], neural networks [64, 75], and Kalman filter [37]. Erroneous predictions provide false information about the process, which might lead to inferior process control and low product yield. Hence, the transformation of the existing algorithms into more robust algorithms is of utmost importance for realizing VM in semiconductor industry. In Chapter 3, we focus on three variants of partial least

squares regression and provide simulation results using the data generated from a generic semiconductor process model present in VM literature. The process model incorporates the effect of different types of process drifts and noise.

While building VM models in the Chapters 2 and 3, we assumed that the sensor data represent the true behavior of the process and are free from sensor faults. Any undesirable process behavior is referred to as a fault and can be further classified into sensor faults, actuator faults, and process faults. Any of these faults can arise while manufacturing a product. Sensor faults are the most relevant faults for VM as VM relies on the sensor data to predict the process outputs. A sensor fault means that the value of a process variable registered by the sensor is significantly different from the true value of the process variable.

The assumption of fault-free sensor data becomes invalid when a malfunctioning sensor corrupts the sensor data. The probability of the occurrence of a sensor fault in a process increases linearly with the number of installed sensors. Currently, semiconductor manufacturing processes deploy a large number of sensors to monitor the process behavior, which leads to a greater risk of the occurrence of sensor faults. When a sensor fault occurs, the corresponding sensor data are erroneous and do not represent the true behavior of the process. The quality of sensor data, which serve as inputs for VM models, has a direct effect on the quality of predicted values. In the presence of faulty input data, an accurate VM model will provide erroneous predictions for the

outputs. This situation is known as Garbage-In-Garbage-Out in process modeling terms. For using VM effectively, we need to make sure that the data to be fed into the VM model are free from faults. Chapters 4 and 5 focus on removing the effect of sensor faults from the sensor data and feeding the corrected (reconstructed) sensor data to the VM model.



Figure 1.2: Summary of the approach adopted in Chapters 4 and 5.

Figure 1.2 summarizes the adopted approach. First, the sensor data are analyzed to detect sensor faults. Once the sensor faults are detected, the next step is to figure out which sensor contains the fault. It is possible that multiple sensors are simultaneously faulty, but the likelihood of the occurrence of simultaneous multiple sensor faults is fairly low as the sensors are independent physical entities. Our focus will be on single sensor faults only. After knowing which sensor is faulty, the magnitude of the fault will be estimated. Fault-free sensor data can be constructed by removing the effect of the identified sensor faults from the faulty sensor data. Mathematically, fault-free sensor data are obtained by subtracting the estimated magnitudes of the faults from the faulty sensors present in the data.

In Chapter 4, we provide a literature review of the popular fault detection and identification approaches. Specifically, we first present fault detection using principal component analysis (PCA). PCA is able to detect faults for

a two-dimensional data matrix only, the two dimensions being time and process variables in most cases. However, the data collected from semiconductor manufacturing processes are three-dimensional, with an additional dimension for different wafers. Hence, PCA cannot be directly applied for fault detection on data collected from a semiconductor manufacturing process. Instead, multiway principal component analysis (MPCA) is employed to address this limitation of PCA.

After discussing MPCA and its limitations when employed for a semiconductor manufacturing process, we will present a statistics pattern analysis (SPA) based fault detection method in Chapter 5 that performs PCA on the statistics of the process variables, unlike the traditional PCA and MPCA methods that perform PCA on the temporal values of the process variables. The advantages of SPA method over the PCA and MPCA methods are discussed in the context of semiconductor manufacturing. Next, we present and discuss three well-known fault identification methods present in literature. Specifically, these include contribution plot approach, reconstruction-based contribution (RBC) approach, and sensor validity index (SVI) approach. The magnitude of the fault is estimated by minimizing the fault detection indices, SPE,  $T^2$ , or  $\phi$ . The fault detection, identification, and reconstruction performance of the above methods are compared using a benchmark etch dataset. This comparison will enable us to determine the approaches that are the well-suited for correcting faults present in sensor data, which serve as inputs for VM models.

The idea of using estimates made by VM as a substitute for physical metrology seems very alluring at the first sight. VM may significantly reduce the measurement costs as it does not require any actual physical measurements. Substituting the physical measurements completely by VM might be the most economical solution to the problem of high measurement costs, but it might fail in the presence of process disturbance and shifts. If any undesired process change happens, VM model might not be able to compensate for the unknown change in the process and might not be able to predict the outputs accurately. The process operator will be under the false impression that the process is running normally, while the actual processed products will not be on the target.

On the other hand, a combination of physical measurements and VM might be a more robust approach. Instead of blindly relying on the estimates made by VM, the combined approach aims at monitoring the quality of VM estimates and performs a physical measurement whenever the quality of VM estimates falls below a threshold value. More metrology events increase the measurement costs and decrease the product throughput (by increasing cycle time), whereas too few metrology events might hamper the product quality. Therefore, the frequency of metrology events needs to be optimized. Thus, the implementation of the combined approach requires the development of optimal sampling plans that will tell the semiconductor manufacturers when to perform a physical measurement to supplement VM predictions.

In the context of deciding which wafers or products should be physically measured, the terms sampling and scheduling represent the same con-

cept. Scheduling the metrology events is equivalent to sampling the wafers to be measured. Whenever the VM prediction accuracy falls below a certain threshold, an actual measurement should be done by the metrology tool and the VM model should be updated. An intuitive solution is to do make frequent physical measurements when VM predictions are quite different from the metrology values and update the prediction model.

In Chapter 6, first we simulate a Single-Input-Single-Output (SISO) process with process drift and noise. Run-to-Run (R2R) control is employed to adjust the recipe settings (inputs) to ensure that the output stays on the target in the presence of process drift and noise. After discussing R2R control for the SISO case in detail to obtain a good understanding, we implement R2R control on a Multiple-Input-Multiple-Output (MIMO) model available in the VM literature. In general, the implementation of R2R control includes the estimation of process gain matrix, process drift, or both. In semiconductor manufacturing, process drift is a major issue of concern as process gain matrix remains almost constant owing to the physics and chemistry behind the process. So, in this work the process drift is estimated using the measurements done according to the sampling plan. Whenever a measurement is made, the value of process drift is estimated by exponentially-weighted-moving-average (EWMA), a weighted average of the previous estimate of the process drift and the process drift value suggested by the current measurement.

Devising an optimal sampling plan is critical in order to ensure that the process outputs are on target, while not spending a large amount of money by

measuring many products. We implement three commonly known sampling methods, uniform sampling, random sampling, and dynamic sampling, in order to demonstrate the superior performance of a novel reliance index based sampling method that utilizes VM estimates. The most common sampling strategy is uniform sampling, which measures a product after a fixed interval of time or products. Random sampling does not have a fixed measurement interval, but measures the products at random intervals that have specified lower and upper limits. Both of these methods do not take advantage of the known past and current behavior of the process. Dynamic sampling is based on the intuitive idea of measuring more products when the process seems to drift away from the target and measuring fewer products when the process outputs are fairly close to the target. Bayesian detection approach will be employed to implement dynamic sampling in this work. The Bayesian detection approach calculates a posterior probability distribution using a prior probability distribution and the observed data. When the probability that the currently observed data is coming from a drifting process exceeds a threshold value, the sampling frequency is increased. An improved dynamic sampling approach, which updates the value of EWMA forgetting factor  $\lambda$  using Bayesian detection, is proposed in this work.

In the three sampling methods mentioned above (uniform sampling, random sampling, and dynamic sampling) the estimates of the process drift and the recipe settings (inputs) of the process are only updated when a physical measurement is made as dictated by the sampling plan. Using the predictions

made by VM model, it is possible to make these updates even when a physical measurement is not done. VM enables us to update the estimate of process drift and the recipe settings of the process after processing each product wafer irrespective of the fact whether the wafer was physically measured or not. An accurate VM model will ensure reduced measurement costs and better controller performance. After processing each wafer, a decision whether the most recent wafer should be measured or not needs to be made. This can be decided by calculating a reliance index that quantifies how much a manufacturer can rely on the VM estimate. If the value of calculated reliance index is below a certain threshold, a physical measurement needs to be made as the manufacturer cannot rely on the VM estimate. Some work on a reliance index is present in VM literature but it suffers from a few shortcomings. A new reliance index, which is more attractive from a mathematical and practical point of view, is proposed in this work.

The summary of the contributions of this research is provided in Chapter 7 along with the recommendations for future work.

## Chapter 2

# Virtual Metrology Methods and Their Application to a Plasma Etch Process

In this chapter, we will introduce various methods used for implementing Virtual Metrology (VM) and compare them in terms of prediction accuracy using industrial data collected from a plasma etch process. Specifically, multiple linear regression (MLR), principal component regression (PCR), partial least squares regression (PLSR), recursive partial least squares regression (R-PLSR), time series analysis, and Kalman filter estimation will be compared in terms of prediction accuracy using four industrial datasets. Kalman filter estimation will be employed in a novel way to serve as a VM model for predicting outputs of a static process.

### 2.1 Introduction

In semiconductor manufacturing, a wafer undergoes hundreds of different steps to yield the final product. It is imperative to have a good knowledge of the wafer characteristics at the end of each of these processes. Otherwise, faulty wafers are detected too late, which leads to the loss of resources. Offline metrology using technologically advanced tools (e.g., ellipsometer, atomic

force microscope (AFM)) ensures that the process is on target by measuring the critical dimensions of the post-process wafers. Due to the associated high cost, offline metrology after each processing step for every wafer may be impractical.

VM is a potential solution for this problem and has many applications in lithography [28], etch [105, 112], and deposition processes [38]. VM predicts end-of-batch properties (metrology data) from measurable input parameters. These input parameters could consist of pre-process metrology and fault detection and classification (FDC) system outputs. The predictions for end-of-batch properties are made by supplying the input parameter information to a model. This model could be physically-based (first principles), empirically determined, or a combination of physical and empirical.

As semiconductor devices are continually shrinking and critical dimension (CD) approaches 32 nm, the control of gate CD, which depends on plasma etch process, is a top priority in advanced semiconductor manufacturing [91]. But most of the work in literature until now on VM aims at predicting the outputs in a CVD process [16, 52, 120], which is relatively easier and better understood than plasma etch process. Due to complex physical and chemical phenomena, it has been difficult to accurately predict plasma processes [64]. However, some authors have explored VM for etching process, as summarized below.

Cheng et al. [16] presented an illustrative example involving 300 mm semiconductor foundry etching equipment, but no comparison of the predicted

values with the actual measurements was reported. Zeng and Spanos [149] applied six statistical techniques to predict etch bias for a plasma etch process, but their results suffer from lack of accuracy. Recently, a VM system was developed for an etching process based on various data mining techniques [59]. The authors used the actual metrology values from a preceding metrology process and equipment sensor data to make the predictions. As stated in the conclusion section of their work, this kind of approach would restrict the usability of the VM system as metrology is not done at the end of each process in semiconductor manufacturing. Some other works [75, 91] also suffer from the same limitation. Further research should be conducted only using the sensor data to build an accurate and reliable VM system.

Cheng et al. [15] developed a business model to measure the profitability of VM based on in-depth manufacturing practices and metrology operations required for semiconductor manufacturing. This paper also proposed a novel manufacturing system that integrates automatic virtual metrology (AVM) into the manufacturing execution system (MES). The interfaces among AVM, other MES components, and run-to-run (R2R) modules in the novel manufacturing system are also defined such that the total quality inspection system can be achieved and the R2R capability can be migrated from lot-to-lot control to wafer-to-wafer control.

## 2.2 Industrial Datasets

In this work, four datasets collected from a plasma etch system at Texas Instruments' DMOS6 wafer fab in Dallas, Texas will be used to compare the VM methods (presented in Section 2.3) in terms of their prediction accuracy. The datasets contain the recorded values of 18 Optical Emission Spectroscopy (OES) signals collected every 0.1 second (i.e., data are collected at a frequency of 10 Hz). Our goal is to predict the output properties (etch rate and sheet resistance) using the OES signals as the input data. The datasets also contain the actual measurement values of the etch rate and sheet resistance in order to evaluate the predictions accuracy of the VM methods. The actual measurements of the etch depth were done using atomic force microscope (AFM). If the etch time is known, the actual value of the etch rate can be calculated. The sheet resistance measurements were made using the test structures similar to those discussed in Smith et al. [108].

Sheet resistance is one of the most important electrical-test parameters used in the semiconductor manufacturing industry to assess the electrical quality of a product. It is defined as the resistance of a square sheet of material with current flowing parallel to the plane formed by the square sides.

Figure 2.1 shows a thin film of length  $L$ , width  $W$ , and thickness  $t$ , made up of a material of resistivity  $\rho$  with current flowing parallel to its length. The resistance of this thin film is calculated using Equation 2.1.

$$R = \frac{\rho L}{A} = \frac{\rho L}{Wt} \quad (2.1)$$

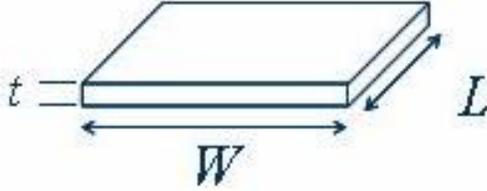


Figure 2.1: A thin film made up of material of resistivity  $\rho$ .

In case of a square film,  $L = W$ , and the resistance becomes equal to the sheet resistance.

$$R = R_s = \frac{\rho}{t} \quad (2.2)$$

From Equation 2.2, we can observe that sheet resistance increases on reducing film thickness. For a given etching time in a trench etch process, a higher value of sheet resistance would indicate a reduction in etch rate because the trench containing the conducting film would be shallower.

Four datasets were collected from a plasma etch system at Texas Instruments' DMOS6 wafer fab in Dallas, Texas for the wafers processed during the periods March-April 2009, September-October 2009, July-August 2010, and January-February 2011; these datasets will be referred to as Dataset 1, Dataset 2, Dataset 3, and Dataset 4, respectively in this work. For the Datasets 1, 2, and 3, lot-level etch rate measurements were available. In other words, only one wafer was actually measured for each lot (a collection of 25 wafers) for the first three datasets. For Dataset 2, the sheet resistance values of every wafer were also available. The input data for the first three datasets comprised of 18 OES signals. Dataset 4 was collected from a gate etch process to figure

out the reason behind the significantly different values of critical dimensions (CDs) of the first two wafers of a lot as compared to the rest of the lot. Dataset 4 was composed of an input data matrix made up of 38 process variables and output CDs for 41 lots (441 wafers).

In order to correlate the OES data with the end-of-batch properties (etch rate and sheet resistance), the data needs to be preprocessed. OES data are available at intervals of 0.1 second, but only one value of sheet resistance is available per wafer. Etch rate values are available for only wafer per lot. Therefore, the temporal behavior of the OES signals recorded during the etch process has to be summarized by a statistic before building a VM model that can predict etch rates and sheet resistance values. To do so, it is assumed that the amount of film removed (etched) is directly proportional to the area under the OES curves in the OES signal vs. time plots. The OES curves for a sample wafer are shown in Figure 2.2. Physically, the higher intensities of the OES signals mean that more molecules are jumping from the excited states to the ground state. This is a result of the higher excitation caused by the plasma leading to high etch rates. Therefore, it is reasonable to assume that the amount removed (etched) is a linear combination of the areas under the OES signal curves for different wavelengths as shown in Equation 2.3.

$$Etch\ rate \times etch\ time = \sum_{i=1}^{18} a_i Area_i + k \quad (2.3)$$

For discrete measurements, the area under each OES curve is the summation of the areas of the rectangles with height equal to the OES signal value

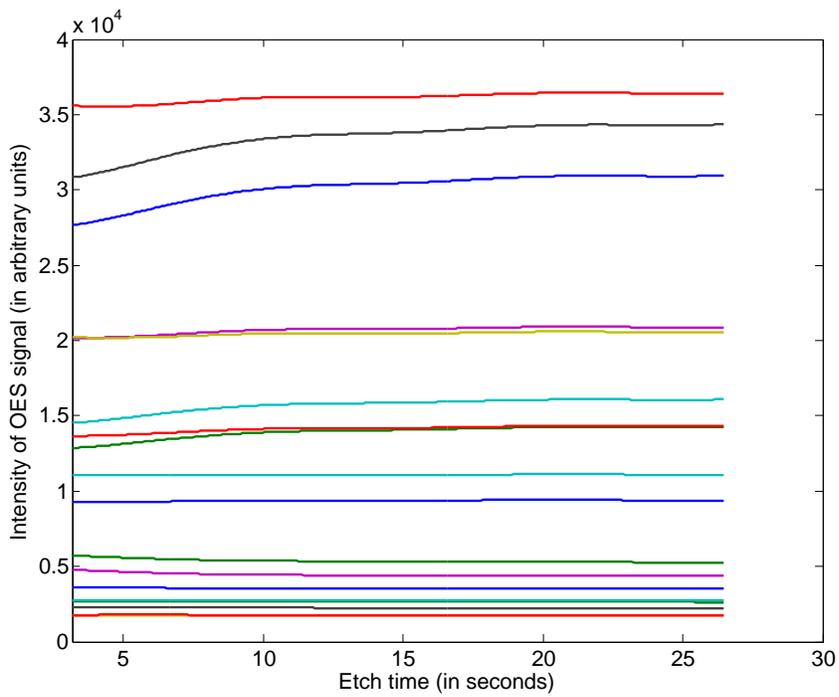


Figure 2.2: Variation of 18 OES signals with time for a sample wafer. Each curve corresponds to an OES signal. There is not much variation in the OES signal values; the mean values of the signals are sufficient metrics to build a regression model.

and breadth equal to the sampling time  $t_{sample}$ .

$$Area_i = \sum_{j=1}^n OES_{i,j} t_{sample} = O\bar{E}S_i \text{ etch time} \quad (2.4)$$

Using Equations 2.3 and 2.4, we get:

$$Etch \text{ rate} = \sum_{i=1}^{18} a_i O\bar{E}S_i + k \quad (2.5)$$

Equations 2.3 - 2.5 suggest that it is equivalent to say whether the amount removed is directly proportional to the area under the OES curves or the etch rates are directly proportional to the mean values of the OES signals. It can be noticed from Figure 2.2 that OES signal values remain almost constant during the etch time; therefore, the mean value of each OES signal is a good choice for the metric to be used in regression.

Equation 2.2 shows that sheet resistance  $R_s$  is inversely proportional to the film thickness. For a given etching time, lower etch rate will yield smaller film thickness. Hence, sheet resistance is inversely proportional to the etch rate and the OES signals. An equation analogous to Equation 2.5 can be written to calculate sheet resistance using the OES signals. The coefficients  $a$  and  $k$  are found using the regression methods discussed in the next section.

$$R_s^{-1} = \sum_{i=1}^{18} a_i O\bar{E}S_i + k \quad (2.6)$$

## 2.3 Methods

In VM literature, multiple linear regression, principal component regression, and partial least squares regression have been commonly adopted to

predict the process outputs using the input data. These methods can be used as benchmarks for evaluating the other three methods (recursive partial least squares, time series analysis, and Kalman filter estimation) presented in this chapter.

### 2.3.1 Multiple Linear Regression

A multiple linear regression model that relates inputs  $x$  with the output  $y$  is shown in Equation 2.7. In this work, the inputs  $x$  are the OES data, which consist of the intensity values of the light emitted at 18 different intervals of wavelengths.  $x_i$  represents the intensity of light emitted at wavelengths encompassed by the  $i^{th}$  wavelength interval and  $y$  represents the process outputs, etch rate and sheet resistance.

$$y = \sum_{i=1}^{18} C_i x_i + \nu \quad (2.7)$$

$\nu$  is a Gaussian white noise term with mean zero and variance equal to  $R$ . The coefficient matrix  $C$  is calculated by minimizing the sum of squares of the differences between the actual measurements and the values of  $y$  obtained from Equation 2.7.  $C$  can be utilized to make the predictions of etch rate and sheet resistance values when new OES data are available.

### 2.3.2 Principal Component Regression

$$X = TP^T + E \quad (2.8)$$

Principal component analysis (PCA) aims at decomposing a data matrix  $X$  into a score matrix  $T$  and a loading matrix  $P$ . The columns of the loading matrix  $P$  represent the principal components. These components are orthogonal to each other and represent the directions that capture the maximum variation in the data matrix  $X$ . Mathematically, the decomposition is shown in Equation 2.8.  $E$  refers to the residuals, the part of  $X$  that is not explained by the principal components. Choosing an optimum number of principal components is critical to obtain a useful PCA model. Too few principal components might not capture all the important characteristics of the data, while too many principal components might incorporate more noise in the model. As a general rule of thumb, the number of principal components is chosen to be the number that explains more than 80% of the variance in  $X$ . An interested reader is referred to Geladi and Kowalski [32] for details.

In principal component regression (PCR), multiple linear regression is carried out between the scores  $T$  of the principal components and the outputs  $y$ . Theoretically, PCR is superior to MLR when there are numerous noisy and correlated predictor (input) variables. Correlated input variables cause collinearity problems and lead to inaccurate estimation of the coefficients  $C$ . PCR removes the correlation between the input variables by providing orthogonal (independent) principal components. The outputs  $y$  are related to the scores  $T$  as shown in Equation 2.9.

$$y = CT + \nu \tag{2.9}$$

### 2.3.3 Partial Least Squares Regression

In PCR, the principal components are calculated without any reference to the output variables [62]. So, the principal components that explain a major part of the variation in independent variables may not be related to the variation in the output variables. This leads to an inaccurate estimation of the regression coefficients. PLS does not suffer from this problem as it has an inner relation that correlates the scores of the independent variables with the scores of the dependent variables, enabling better prediction power than PCR. Furthermore, PLS has been shown to be a robust multivariate linear regression technique [66].

Suppose that we want to build a PLS model with  $n$  principal components for predicting the output variable matrix  $y$  using the input variable matrix  $X$ . Then the outer relation for  $X$  is given by Equation 2.10.

$$X = TP^T + E = \sum_{i=1}^n t_i p_i + E \quad (2.10)$$

$T$ ,  $P$ , and  $E$  refer to the scores, loadings, and residuals of  $X$ , respectively.  $t_i$  and  $p_i$  are the  $i^{th}$  columns of  $T$  and  $P$ , respectively and represent the  $i^{th}$  principal component. A similar outer relation can also be written for  $y$ .

$$y = UQ^T + F = \sum_{i=1}^n u_i q_i + F \quad (2.11)$$

$$u_i = b_i t_i + e_i \quad (2.12)$$

The scores of matrices  $X$  and  $y$  have an inner relation given by Equation 2.12. Here,  $b_i$  refers to  $i^{th}$  regression coefficient. After calculating the regression

coefficients, the outputs for the new input data can be predicted. A detailed PLS algorithm can be found in Geladi and Kowalski [32].

#### **2.3.4 Recursive Partial Least Squares Regression**

Recursive PLS regression, a recursive version of the PLS regression, updates the model whenever new data are available. It has better adapting capability than PLS and proves beneficial when the process changes occur. More details can be found in Qin et al. [99]. Recursive PLS regression will be revisited in Chapter 3.

#### **2.3.5 Time Series Analysis**

The datasets under investigation were collected from consecutive lots, i.e., sequential in time. Time series analysis can determine any sequential pattern present in the data. This can be done by implementing the practical procedure proposed by Box and Jenkins [10], which consists of three stages: identification, estimation, and diagnostic checking. After finding an optimal model using these three stages, predictions/forecasts can be made.

At the identification stage, two functions are used to measure the degree of correlation between the observations within a data series. These functions are known as estimated autocorrelation function (acf) and estimated partial autocorrelation function (pacf). They are helpful in providing a crude idea about the patterns present in the data under investigation. These functions can be compared with those of different ARIMA (Auto-Regressive-Integrated-

Moving Average) models. The model whose theoretical acf and pacf closely resemble the estimated acf and pacf of the data series can be chosen a tentative model for the data under investigation.

At the estimation stage, precise estimates of the parameters in the identified model are calculated. The nonlinear least-squares (NLS) technique most commonly used for estimation is a combination of two NLS procedures, Gauss-Newton linearization and the gradient method. This combination is often referred to as Marquardt's compromise [78]. Mathematical details of this method are provided in [10, 92].

A statistically adequate model is the one whose random shocks are statistically independent, meaning not autocorrelated. This is checked by finding the residual acf. The ultimate application of ARIMA modeling is to forecast future values of a time series. Once we know the model structure and the estimated parameters, it is possible to predict the future values of etch rates and sheet resistance.

### **2.3.6 Kalman Filter Estimation**

To our best knowledge, no technique that takes lot-to-lot model error into consideration has been proposed for VM so far in literature. Predictions made in the past can be compared with the actual measurements and relevant adjustments can be made for the future predictions. Mathematically, this means to add a term to the prediction equation that compensates for this bias, multiplied by a gain factor. This gain factor should be chosen such that

the error between the predicted and actual values and the error covariance matrix is minimized. These are features of Kalman filter, which is a well known estimation technique and is discussed in this subsection.

$$y = \sum_{i=1}^{18} C_i x_i + \nu \quad (2.13)$$

A linear regression model that relates inputs  $x$  with the output  $y$  is shown in Equation 2.13. Let us assume that the output measurements are corrupted by Gaussian white noise  $\nu$  with mean zero and variance equal to  $R$ .

Although the Kalman filter [58] was designed in early 1960s to estimate the states of dynamic or time-varying systems, it can be used on static systems as well. Pachter and Altman [89] used this technique to determine the structure of certain protein molecules using geometric information provided by nuclear magnetic resonance (NMR) studies. The formulation of the Kalman filter for a static system is provided below [40].

$$K = P_{old} C^T (C P_{old} C^T + R)^{-1} \quad (2.14)$$

$$x_{new} = x_{old} + K(y - C x_{old}) \quad (2.15)$$

$$P_{new} = P_{old} - K C P_{old} \quad (2.16)$$

$$y_{est} = C x_{new} \quad (2.17)$$

In the above equations,  $K$  is the Kalman gain,  $P$  is the state error covariance matrix, and  $R$  is the measurement noise covariance matrix. The initial value of  $P$  was chosen to be the error covariance matrix of the states in

the training set.  $R$  was chosen to be the variance of the measurement values of etch rates and sheet resistance in the data used for model building.  $x_{old}$  is the state matrix made up of OES signals and  $x_{new}$  is the updated state matrix.  $y$  represents the measured values of etch rates and sheet resistance;  $y_{est}$  denotes the estimated values of etch rates and sheet resistance.  $C$  is the coefficient matrix that relates the OES signals with the outputs.

## 2.4 Results and Discussion

In this section, we will present the prediction results of the VM methods discussed in Section 2.3. The VM methods are implemented on the four datasets collected from a plasma etch system at Texas Instruments, Inc. (see Section 2.2 for details). For all these datasets, 70% of the data were utilized for model building and the rest of the data were used for model validation. The first three datasets contain 18 OES signals and etch rate values. OES signals are available for all the wafers in the datasets but only one value of etch rate is available per lot (a collection of 25 wafers). First, we will be comparing the VM methods in terms of their ability to predict etch rates at lot-level.

Sheet resistance data are available for all the wafers in Dataset 2. So, wafer-level predictions of sheet resistance can be made using the OES signals and will be presented next. Dataset 4 was collected from a gate etch process to figure out the reason behind the significantly different values of critical dimensions (CDs) of the first two wafers of a lot as compared to the rest of the lot. Last, wafer-level predictions of CD will be made using an input data

matrix of 38 process variables. The process variables causing a non-uniformity in the CDs of the first and second wafers of the lots will also be identified.

### 2.4.1 Lot-level etch rate predictions

Equation 2.5 shows that the etch rate of a wafer is linearly proportional to the values of OES signals averaged over wafer’s etching time. This step provides data that are suitable for making wafer-level predictions. As only one value of etch rate is available per lot for the first three datasets, the data need to be preprocessed further before implementing VM methods. The averaged values of the OES signals are further averaged over all the wafers in a lot. The preprocessed data now contains one etch rate value and one OES value for each of 18 OES signals per lot.

The prediction performance of the VM methods is quantified by calculating mean absolute percentage error (MAPE) as defined below.

$$MAPE = \frac{100}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \% \quad (2.18)$$

Table 2.1: Comparison of VM methods for the lot-level prediction of etch rate from OES data.

VM Method	MAPE (Dataset 1)	MAPE (Dataset 2)	MAPE (Dataset 3)
Multiple linear regression	9.6253	4.1146	20.0643
Principal component regression	4.0331	3.0607	2.8069
Partial least squares regression	3.9044	3.0560	2.7494
Recursive partial least squares regression	2.7165	3.0514	2.6956
Kalman Filter	2.2599	3.0413	2.7380
Time series analysis	1.5771	6.1600	3.5711

In Equation 2.18,  $y_i$  is the actual metrology value of the output,  $\hat{y}_i$  is the predicted value of the output, and  $n$  is the total number of predictions. Table 2.1 contains the *MAPE* values obtained by using different prediction methods for the first three datasets. Due to the correlated nature of the inputs (OES signals), multiple linear regression (MLR) did not provide good prediction accuracy. For a system of linear equations, condition number is a measure of sensitivity of the solution vector to noise in the data. Mathematically, it is the ratio of the largest singular value of the data matrix to the smallest. A data matrix with a low condition number is said to be well-conditioned, while a problem with a high condition number is said to be ill-conditioned. For Datasets 1, 2, and 3, the condition numbers were found to be 2390.6, 1398.7, and 1754.8, respectively, which are orders of magnitude greater than 1. This explains why MLR estimates are unreliable and not very accurate for all three datasets. As expected, partial least squares regression (PLSR) provided better results than principal component regression (PCR) and MLR for all three datasets. 14 principal components explained more than 80% of the variation in the data. Recursive PLS regression provided the best prediction results among all the PCA-based methods.

Time series analysis provides good predictions for Dataset 1, but not for Datasets 2 and 3. This is because of the presence of slight upward drift in the etch rate in the validation sets of Datasets 2 and 3, which is not identified by the time series model. The predictions drift away from the actual measurements as lot-to-lot model error is not taken into consideration. This suggests

that it is very important to update the VM model by feeding back the model error, so that any departure away from the target can be compensated in time.

Recursive PLS regression (R-PLSR) and Kalman filter show the best prediction results as they update the model whenever new measurements are available. Figure 2.3 shows the prediction results for etch rate for the training data of Dataset 2. Dataset 2 contains input and output data for 51 lots (1121 wafers). 70 % of the data (35 lots) were used for model building/training and the rest were used for validation. The parameters trained during the training phase are the model coefficients of PLS and time series analysis, measurement noise covariance  $R$  for Kalman filter, and the forgetting factor for recursive PLS regression. As recursive PLS regression and Kalman filter update the model coefficients whenever new data are available, training these coefficients would not be useful. For the validation data, the prediction results of the three best methods for Dataset 2 are shown in Figure 2.4 for direct comparison with those of sheet resistance in Figure 2.6.

#### **2.4.2 Wafer-level sheet resistance predictions**

Although the lot-level predictions of etch rate seem fairly accurate good as depicted by small *MAPE* values in Table 2.1, the correlation between the OES signals and the etch rates was not found to be very strong; a maximum  $R^2$  value of 0.31 was observed. To obtain better correlation between the input and the output variables, a quality variable that was measured for each wafer was identified. Therefore, sheet resistance data were collected for 1121 wafers

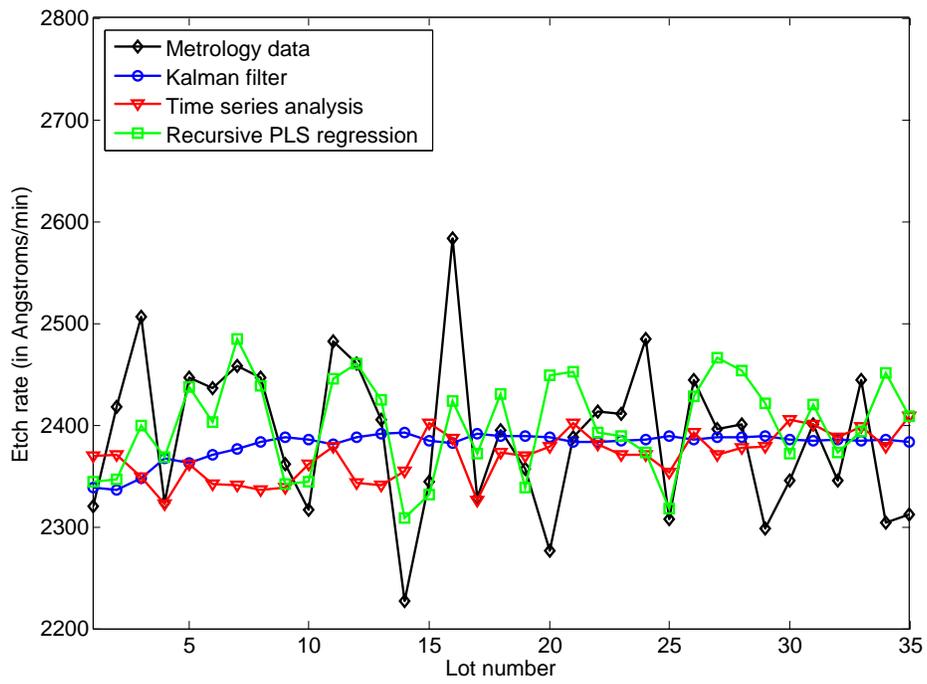


Figure 2.3: Comparison of VM methods for the lot-level prediction of etch rate from OES data for Dataset 2 (training).

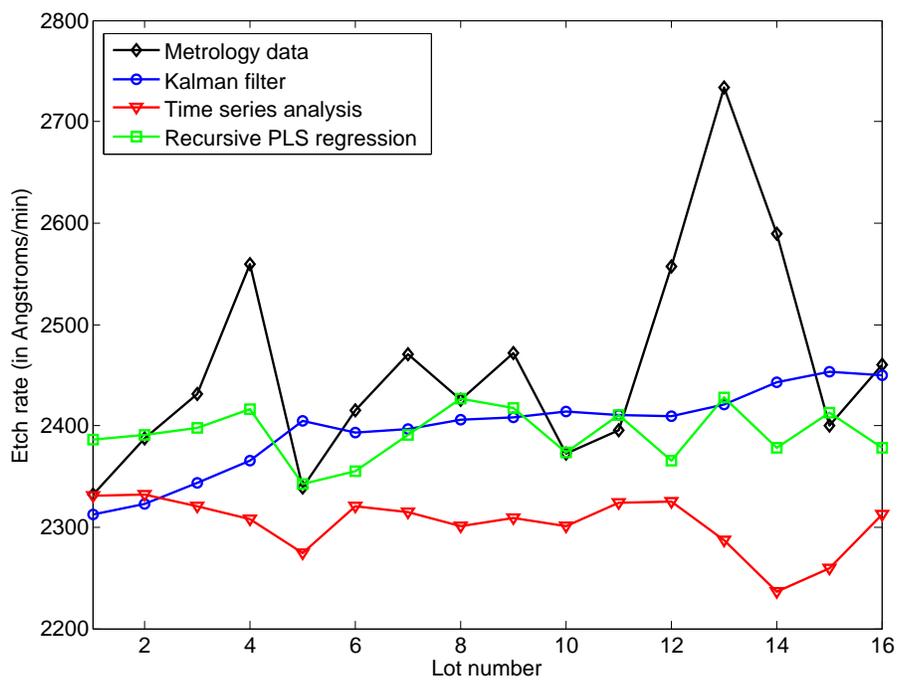


Figure 2.4: Comparison of VM methods for the lot-level prediction of etch rate from OES data for Dataset 2 (validation).

and correlated with OES data using various VM methods. Models were built using 70 % of the total data (784 wafers) and the rest of the data (337 wafers) were used for validation. Sheet resistance data were available only for Dataset 2. Table 2.2 compares various VM methods in terms of  $MAPE$  and  $R^2$  values for wafer-level predictions of sheet resistance for Dataset 2.

Table 2.2: Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2.

VM Method	MAPE (Dataset 2)	$R^2$ (Dataset 2)
Multiple linear regression	8.7206	0.2387
Principal component regression	7.1892	0.5462
Partial least squares regression	6.8592	0.5847
Recursive partial least squares regression	5.0700	0.7149
Kalman Filter	5.2882	0.6953
Time series analysis	8.4565	0.4002

While comparing Tables 2.1 and 2.2, it can be observed that the predictions for sheet resistance have larger  $MAPE$  values than those of the predictions for etch rate. This is because of larger variance of the sheet resistance data as compared to the etch rate data. Table 2.3 provides signal-to-noise (mean-to-standard deviation) ratios (SNR) for sheet resistance and etch rate for Dataset 2. SNR of etch rate is more than thrice of SNR of sheet resistance.  $MAPE$  is not a good choice for performance metric if the output variables have significantly different SNR. Correlation is better described by the correlation coefficient  $R^2$  in such situations.

Figures 2.5 and 2.6 show the prediction results for sheet resistance for the training and validation data for Dataset 2, respectively. It is evident from Table 2.2 and Figure 2.6 that the OES data have much better correlation with

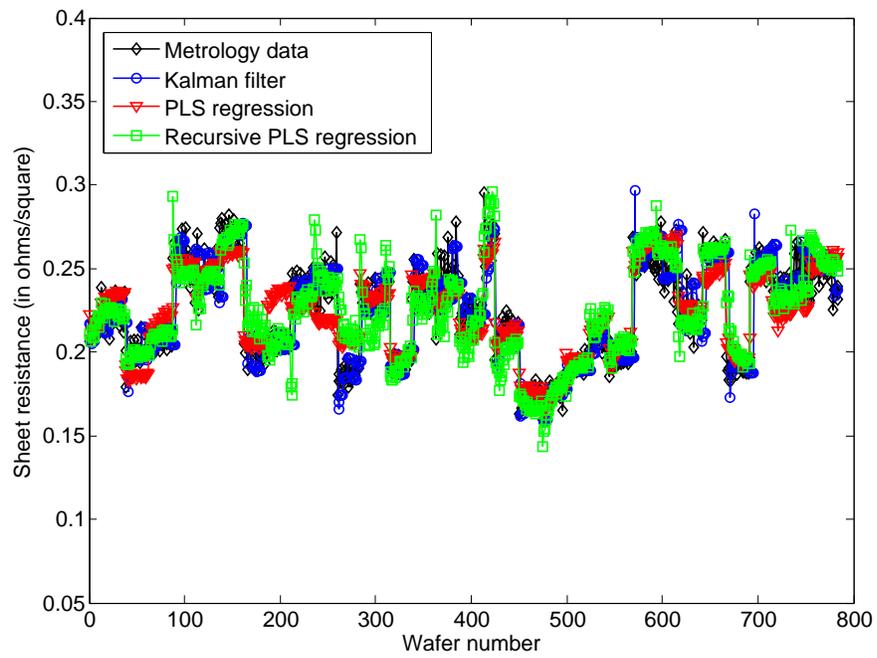


Figure 2.5: Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2 (training).

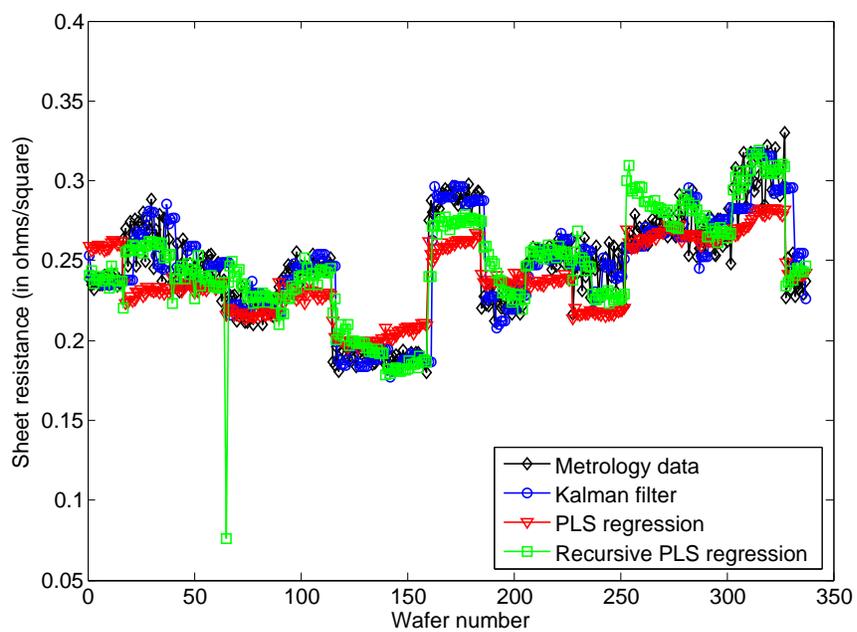


Figure 2.6: Comparison of VM methods for the wafer-level prediction of sheet resistance from OES data for Dataset 2 (validation).

Table 2.3: Signal-to-noise ratio of the measured sheet resistance and etch rate for Dataset 2.

Output variable	Signal-to-noise ratio (SNR)
Etch rate	26.5812
Sheet resistance	7.3751

the sheet resistance data as compared to the etch rate data. Sheet resistance was observed to be a strong function of four OES signals that capture emissions from the wavelength intervals 481-483 nm, 504-506 nm, 702.5-704.5 nm, and 776-778 nm. These wavelength intervals correspond to emissions from  $F$ ,  $O_2$ ,  $SiF_4$ ,  $CO$ ,  $Si$ , and  $He$  ions. Hence, we can deduce that sheet resistance strongly depends on the optical emissions from these ions. An etch recipe composed of  $CF_4$  and  $He$  was used for etching Organo-silicate Glass ( $SiO_2$  with hydrocarbons at interstitial sites) in this work. In short, the modeling results were found to be in agreement with the process chemistry.

### 2.4.3 Wafer-level critical dimension (CD) predictions

Dataset 4 was collected to figure out the reason behind the non-uniformity in the etch CDs of wafers in a gate etch process. The first two wafers of the processed lots had significantly different CDs than the rest of the lot. Dataset 4 was composed of an input data matrix made up of 38 process variables (names of process variables not provided as it is proprietary information) and output CDs for 41 lots (441 wafers). If we are able to predict the CDs precisely using the process variables, we can also figure out which process variables have the most significant effect on the output. These process variables would be the

ones which are more likely to cause undesired non-uniformities in the output etch CD.

The collected data needed multiple steps of preprocessing because of the issues with the data mentioned below. The etching times for the wafers in a lot were not the same and the process variables recorded for the wafers were also different. These issues were overcome by building data matrices that contained information of common process variables for the same number of time stamps for all wafers in a lot. Another challenging feature of Dataset 4 was that the data were only recorded when the value of a process variable changed during the process so that database memory usage was minimized. The data for the time stamps in between two available values were filled in to ensure a data value exists at each time stamp of processing. Normalization of time was also required for each wafer because the time stamps in the collected data are recorded as the time passed since a fixed time instant in the past.

After preprocessing the data, a PLS model was built to predict CDs of the wafers using 38 process variables. As trace data were available for the process variables, a mean value was calculated for each process variable by averaging the values over the processing time. This data compression step is required because only one value of CD exists for a wafer. So, the size of input data matrix was  $441 \times 38$  and the size of the output matrix was  $441 \times 1$ . Cross validation was done to choose the number of PLS factors and the best fit was obtained when 35 factors were used. The prediction performance of the VM methods was quantified by calculating mean absolute percentage error

(MAPE) (see Equation 2.18).

The PLS model provided fairly good predictions with the MAPE value of 1.5159, which is lower than the acceptable error of 2 percent mentioned in VM literature. Figure 2.7 shows the predicted values of CD using the PLS model for 441 wafers. The predicted and measured values for the first and second wafers of each lot are marked to show that most of the outliers come from first two wafers. To obtain a good prediction model, the outliers are left out of the dataset generally. But for this study, the data for first two wafers for each lot are retained in order to investigate the cause behind the CD non-uniformity. The measured values of CD are plotted against the predicted CDs in Figure 2.8 to show the positive correlation that exists between the predicted and measured CDs. A  $R^2$  value of 0.4324 was obtained. The equation of linear fit was found to be  $y = x + 8.78 E - 13$ .

As a PLS model is built using normalized data (zero mean and unit standard deviation), the PLS model coefficients represent an unbiased contribution of each input variable to the output. Figure 2.9 shows the PLS model coefficients for 38 input variables for the gate etch process. Variable numbers 4, 6, 9, 11, 13, 27, 29, 35, and 37 have significantly larger coefficients than the rest of the process variables. In other words, these nine process variables have significantly larger effect on the CD values than the rest of the process variables. Most likely, these variables are causing the undesired CD values for the first two wafers in the lots under consideration.

In order to verify the results from the PLS model, we can plot the

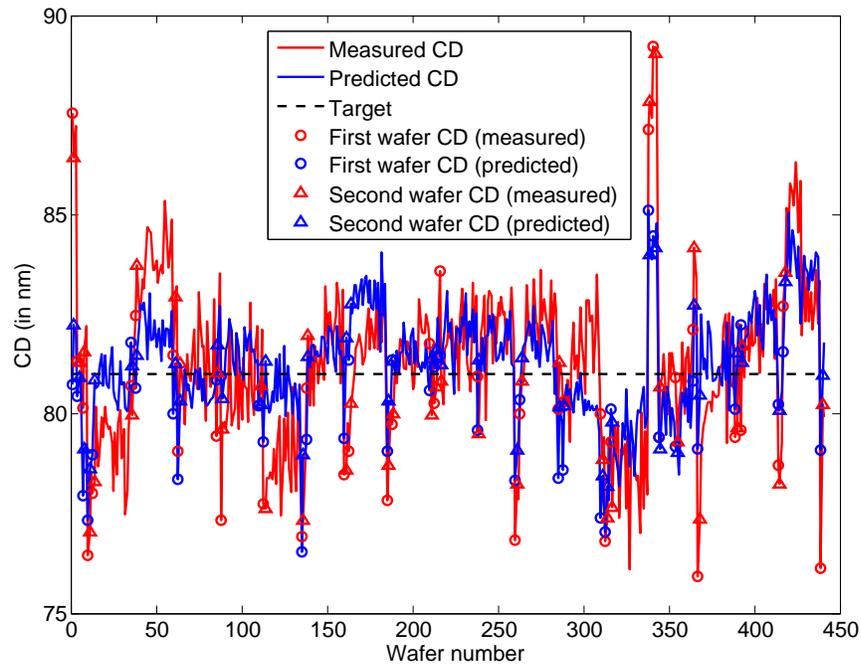


Figure 2.7: CD values predicted by PLS model are fairly close to the measured CDs and have a Mean Absolute Percentage Error (MAPE) of 1.5159. Most of the outliers come from first two wafers of the lots. For the gate etch process under investigation, target CD was 81 nm.

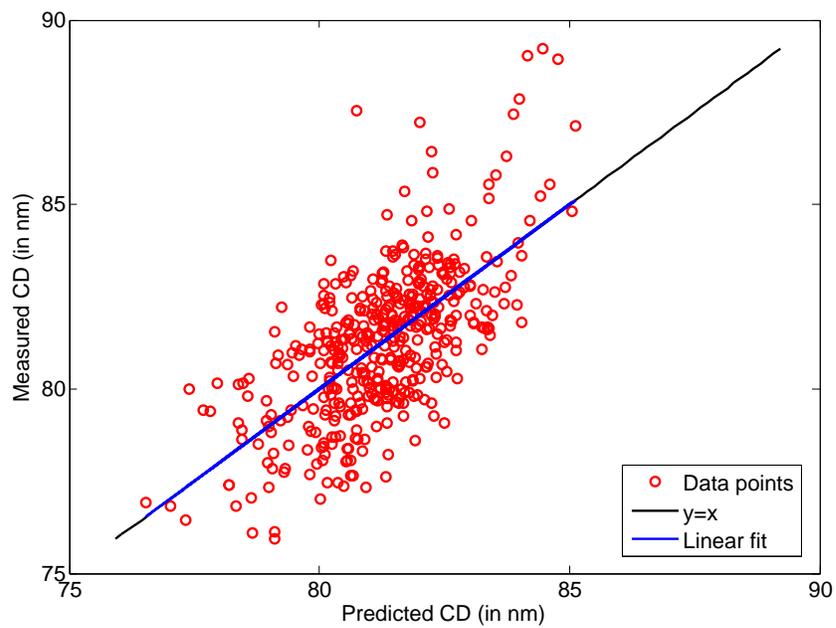


Figure 2.8: A positive correlation exists between the measured CDs and the predicted CDs. A  $R^2$  value of 0.4324 was obtained. The equation of linear fit is  $y = x + 8.78E - 13$ .

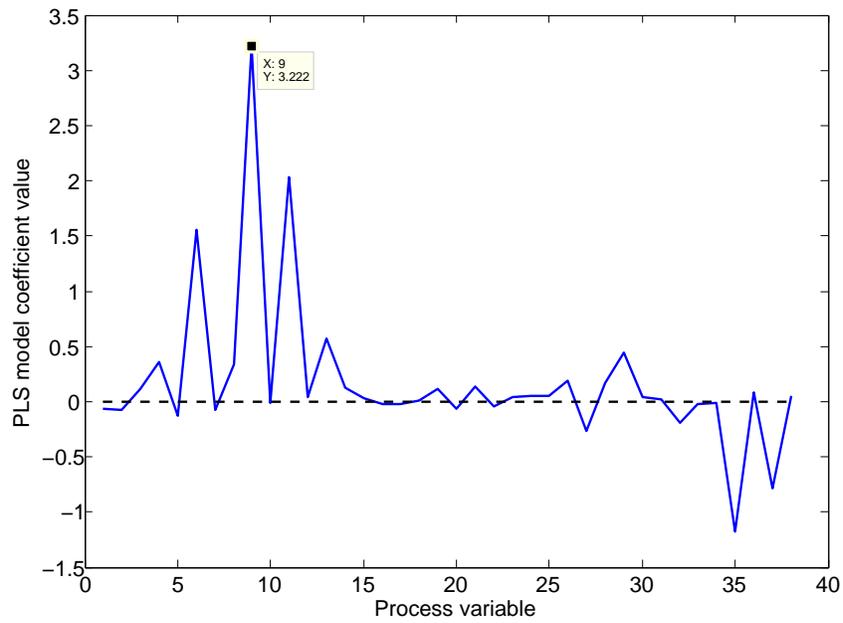


Figure 2.9: PLS model coefficients for 38 input process variables. Variable numbers 4, 6, 9, 11, 13, 27, 29, 35, and 37 have significantly larger coefficients than the rest of the process variables. Most likely, these variables are causing the undesired CD values for the first two wafers in the lots under consideration.

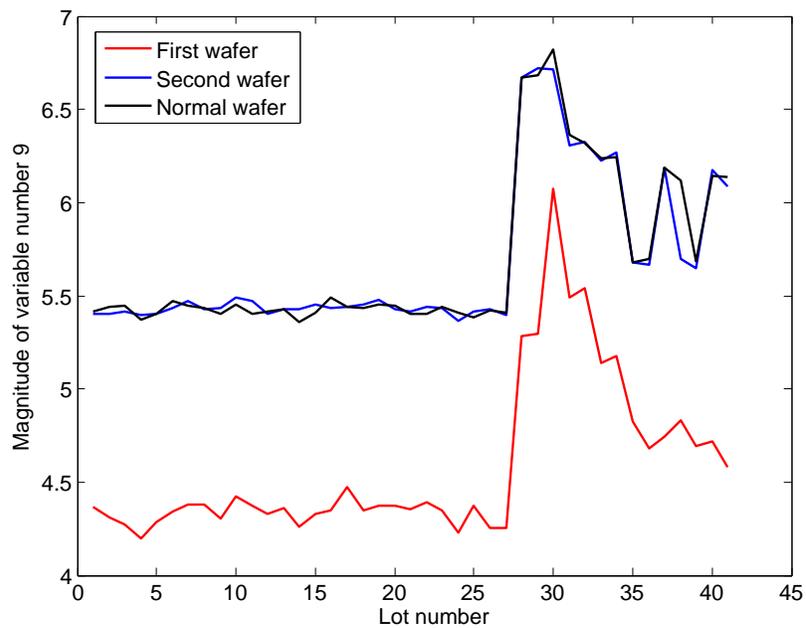


Figure 2.10: Magnitude of variable number 9 for the first wafer, second wafer, and a normal wafer in each of 41 lots under consideration. For all the lots, the first wafer has a very different value than a normal wafer. Therefore, variable number 9 is one of the variables that cause a different CD value for the first wafer as compared to other wafers in the lot.

process variables for different wafers in a lot. A large PLS coefficient means that the process variable has a significant effect on the etch CD. For most of the lots, CD of the first wafer was found to be different than the rest of the lot. Figure 2.10 shows that the value of variable number 9 is very different for the first wafer in all the lots as compared to other wafers in the lot. This supports the PLS result which states that variable number 9 is a significant input variable for predicting etch CD. In Figure 2.10, the first 27 lots were processed on a different etch tool than that processed the next 14 lots. This is the reason for the jump in the value of variable number 9 after lot 27.

## 2.5 Conclusions

In this chapter, various VM methods were introduced and compared in terms of prediction accuracy using four industrial datasets collected from a plasma etch system at Texas Instruments, Inc. Specifically, multiple linear regression (MLR), principal component regression (PCR), partial least squares regression (PLSR), recursive partial least squares regression (R-PLSR), time series analysis, and Kalman filter estimation were implemented to predict process outputs such as etch rate, sheet resistance, and critical dimension (CD). Kalman filter estimation was employed in a novel way to serve as a VM model for predicting outputs of a static process.

First, lot-level predictions were made for etch rate using 18 optical emission spectroscopy (OES) signals for the first three datasets. Due to the correlated nature of the inputs (OES signals), multiple linear regression (MLR)

did not provide good prediction accuracy. As expected, partial least squares regression (PLSR) provided better results than principal component regression (PCR) and MLR for all three datasets. Recursive PLS regression provided the best prediction results among all the PCA-based methods. Time series analysis provided good predictions for Dataset 1, but not for Datasets 2 and 3. This is because of the presence of slight upward drift in the etch rate in the validation sets of Datasets 2 and 3, which was not identified by the time series model. The predictions drifted away from the actual measurements as lot-to-lot model error was not taken into consideration. This suggested that it is very important to update the VM model by feeding back the model error, so that any departure away from the target can be compensated in time. Recursive PLS regression (R-PLSR) and Kalman filter showed the best prediction results as they update the model whenever new measurements are available. However, the correlation between the OES signals and etch rate was not found to be very strong because only one value of measured etch rate was available per lot.

Next, to obtain better correlation between the input and the output variables, a quality variable that was measured for each wafer was identified. Sheet resistance data were collected for 1121 wafers and correlated with OES data using various VM methods mentioned above. Recursive PLS regression and Kalman filter showed the best wafer-level predictions for sheet resistance using the OES data. It was observed that the OES data have much better correlation with the sheet resistance data as compared to the etch rate data. Sheet resistance was observed to be a strong function of the OES signals that

represent the optical emissions from the gases present in the etch recipe. In other words, the modeling results were found to be in agreement with the process chemistry.

Last, Dataset 4 was collected from a gate etch process to figure out the reason behind the non-uniformity in the etch CDs of wafers. The first two wafers of the processed lots had significantly different CDs than the rest of the lot. Dataset 4 was composed of an input data matrix made up of 38 process variables and output CDs for 41 lots (441 wafers). After preprocessing the data, a PLS model was built to predict CD values using the process variables. The model predictions were found to be fairly good with a MAPE value (see Equation 2.18) of 1.5159 and a  $R^2$  value of 0.4324. Nine process variables that had the most significant effect on the CD were identified. Most likely, these process variables were responsible for causing the undesired CD values for the first two wafers in the lots under consideration.

## Chapter 3

# Tailored PLS Algorithms for Handling Unexpected Drifts and Noise in Virtual Metrology

Chapter 2 provided a comparison of various modeling techniques using industrial data for the etch process. Specifically, multiple linear regression, principal component regression, partial least squares regression, recursive partial least squares regression, time series analysis, and Kalman filter estimation were compared in terms of prediction accuracy using three industrial datasets. Kalman filter estimation was employed in a novel way to serve as a VM model for predicting outputs of a static process. Recursive partial least squares and Kalman filter estimation yielded better predictions than the rest of the methods.

In this chapter, we will focus on three variants of partial least squares regression and provide simulation results using the data generated from a generic semiconductor process model present in VM literature. The process model will incorporate the effect of different types of process drifts and noise.

### 3.1 Introduction

In semiconductor manufacturing, a wafer undergoes hundreds of processing steps before yielding the final product. Unexpected process drifts, shifts, and noise deteriorate the processing quality of each of these steps, which ultimately affects the final product quality. In order to achieve high product yield, it is imperative to quantify product quality at the end of each processing step. An ideal solution would be to take measurements for every wafer after each processing step. However, this approach can not be adopted by the industry as it will result in very high metrology costs leading to reduced profits.

Several researchers [52, 59, 91, 149] have suggested Virtual Metrology (VM) as an alternative approach, which aims at the estimation of end-of-batch properties from measurable input/process variables. Unexpected process drifts and noise can severely hamper the accuracy of predictions made by popular VM algorithms such as PCA, PLS [63], neural networks [64, 75], and Kalman filter [37]. Erroneous predictions provide false information about the process, which might lead to inferior process control and low product yield. Hence, the transformation of the existing algorithms into more robust algorithms is of utmost importance for realizing VM in semiconductor industry.

Some work [62, 63] has been done in the past to include the effect of process drifts and noise in the virtual metrology model, but the authors assumed that the magnitude of process drift and noise was known beforehand. Building a PLS model with process drift as one of the input vectors makes it

easier to track the known drift while making predictions. This model had limited prediction capability as only the known drifts can be tracked. Due to the lack of robustness and generalization, the model would fail to track unknown process drifts and noise. Therefore, in this chapter, an effort has been made to tailor the existing PLS algorithm to track unexpected drifts and noise, which would result in a robust and adaptive VM algorithm. The objective of this work is to provide generic guidelines that will assist us to select an appropriate VM algorithm when the process under investigation is infected by drifts and noise.

### 3.2 Process Model

In order to arrive at generic conclusions, a linear process model (shown in Equation 3.1) that describes a typical semiconductor process was simulated. This model was proposed by Khan et al. [63]; the authors implemented it to prove that VM has the ability to improve run-to-run control of a semiconductor process. Han et al. [41] also simulated the same model to compare different VM algorithms, but the presented results lacked consistency and did not provide generalized guidelines for the selection of appropriate VM algorithm.

$$y_k = u_k A + \eta_k + \varepsilon_k \quad (3.1)$$

In Equation 3.1,  $y_k$  is the response vector (quality variables) obtained at the end of run  $k$ ,  $u_k$  is the process input (recipe settings) vector that is set at the start of run  $k$ ,  $A$  is the process gain matrix that relates the inputs to

the outputs,  $\eta_k$  is the process drift vector, and  $\varepsilon_k$  is a white noise sequence.

Along with the linear model shown above, Han et al. [41] also simulated a nonlinear model given by Equation 3.2. This was done to investigate the effect of nonlinearity on the performance of prediction methods.

$$y_k = u_k A u_k^T + \eta_k + \varepsilon_k \quad (3.2)$$

The simplified example simulated in this work assumes that the processes represented by Equations 3.1 and 3.2 can be described by two input variables ( $u_1$  and  $u_2$ ), six process variables ( $v_1, v_2, v_3, v_4, v_5$ , and  $v_6$ ), and two output variables ( $y_1$  and  $y_2$ ). The process variables and the process gain matrix are chosen such that the simulation study closely mimics the true behavior of a semiconductor manufacturing process. The general characteristics of a semiconductor manufacturing process are: some process variables are correlated with each other; not all the process variables depend on the inputs; there is no 1-1 relationship between the inputs and the outputs; the output variables experience different amounts of drifts and noise. Taking these features into consideration, the process variables and the process gain matrix can be represented by Equations 3.3 - 3.9.

$$v_1 = 0.3u_1 + 0.4u_2 + 0.7 \quad (3.3)$$

$$v_2 = 0.2v_1 \quad (3.4)$$

$$v_3 = 0.2u_1 + 0.2\eta_1 + 0.1 \quad (3.5)$$

$$v_4 = 0.7u_1 + 0.5u_2 + 0.3\eta_1 + 0.8\eta_2 + 0.4 \quad (3.6)$$

$$v_5 = 0.2v_1 - 0.1v_4 \quad (3.7)$$

$$v_6 = 1.5 \quad (3.8)$$

$$A = \begin{bmatrix} 0.5 & -0.2 \\ 0.25 & 0.15 \end{bmatrix} \quad (3.9)$$

As semiconductor manufacturing processes suffer from different kinds of drifts, it is essential to incorporate various drift types in the process models represented by Equations 3.1 and 3.2. The process drift vector  $\eta_k$  can be easily modified to reflect a desired drift type. In this work, four types of process drifts are investigated: no drift, linear rise, ramp change, and drift starting arbitrarily during the process. Linear rise refers to a drift type that causes a linear increase in the outputs, ramp change also exhibits a linear increase in the outputs but the drift vanishes after some time before reappearing again, and the drift starting arbitrarily during the process does not appear immediately when the process starts, but takes some time to show its effect on the outputs.

### 3.3 Design of Experiments

Design of experiments (DOE) is a standard approach in semiconductor industry employed to study the effect of process inputs on process outputs. The effect of an input variable is determined by conducting experiments in which only the value of the variable under consideration is altered; other variables remain unchanged. DOE data is used to build the initial VM model which later predicts the values of outputs from new input data. In this study, a full-factorial design for two input variables was carried out. Figure 3.1 shows

the operating points A, B, C, D, and E used to build PLS model. It should be noted that DOE data used for model building does not suffer from process drifts and noise, which ensures that any drift or noise present in the validation set is unknown to the PLS model. This fact will enable us to evaluate the robustness of VM algorithms in presence of unexpected process drifts and noise.

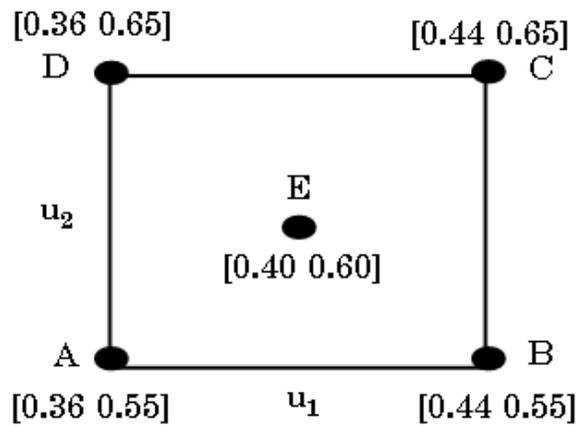


Figure 3.1: Operating points for PLS model building based on a full-factorial design.

### 3.4 Prediction Methods

The most popular VM algorithms till date comprise of PCA, PLS [63], neural networks [64, 75], and Kalman filter [37]. Several researchers [63, 73, 91] employed PLS to predict the future properties of a wafer from the current processing conditions. The essential features of PLS include fairly easy implementation, good prediction accuracy, and its ability to identify the input

variables which affect the outputs to a great extent. However, PLS estimates are only effective if all the variables affecting the outputs are incorporated in the model. This limitation incapacitates the traditional PLS algorithm [32] to track unknown process drifts and noise. Hence, in this chapter, we present two variants of PLS algorithm that can track the unknown process drifts and noise more effectively as compared to the conventional PLS algorithm.

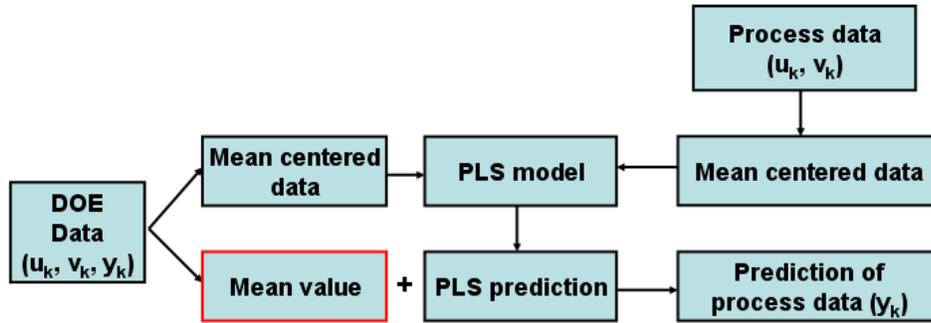


Figure 3.2: Implementation of PLS to predict the values of process outputs.

### 3.4.1 PLS with EWMA mean update

To build a PLS model, the data has to be mean-centered (normalized) first to ensure that only the variations around the mean value take part in the model building. These mean values are added back to the predictions made by the model in order to bring back the process outputs to unnormalized space. Traditionally, these means are calculated from DOE data and kept constant while making predictions. However, unknown process drifts and noise can significantly change the process mean values, which must be updated to achieve acceptable prediction accuracy. One simple way to update the mean

value is using Exponentially Weighted Moving Average (EWMA) as shown in Equations 3.10 and 3.11.

$$mean_{i+1} = (1 - \lambda)mean_i + \lambda m_{i+1} \quad (3.10)$$

$$y_{i+1} = z_{i+1} + mean_{i+1} \quad (3.11)$$

In these equations,  $mean_{i+1}$  denotes the updated mean value to be used for making predictions at time  $i + 1$ ,  $m_{i+1}$  refers to the raw mean value corresponding to the latest observations,  $\lambda$  is the EWMA weighing factor that is set by user,  $y_{i+1}$  stands for the predicted value of  $y$  in unnormalized units at time  $i + 1$ , and  $z_{i+1}$  denotes the PLS prediction of  $y$  in scaled (normalized) units at time  $i + 1$ .  $\lambda$  denotes the weighting given to the latest measurements to estimate the mean value of the predicted values. A high value of  $\lambda$  (close to 1) means that the latest measurements reflect the true behavior of the process and must be trusted to a high degree to estimate the mean of predicted values. However, in presence of large measurement noise, trusting the latest measurements might lead to inaccurate estimates of mean. A low value of  $\lambda$  (close to 0) represents a more conservative update of mean values. Providing a small weighting to the latest measurements makes the predictions more robust to measurement noise but might lead to a sluggish response in case of a process drift or shift. In this work, the value of  $\lambda$  was set to 0.3 as done in most industrial applications.

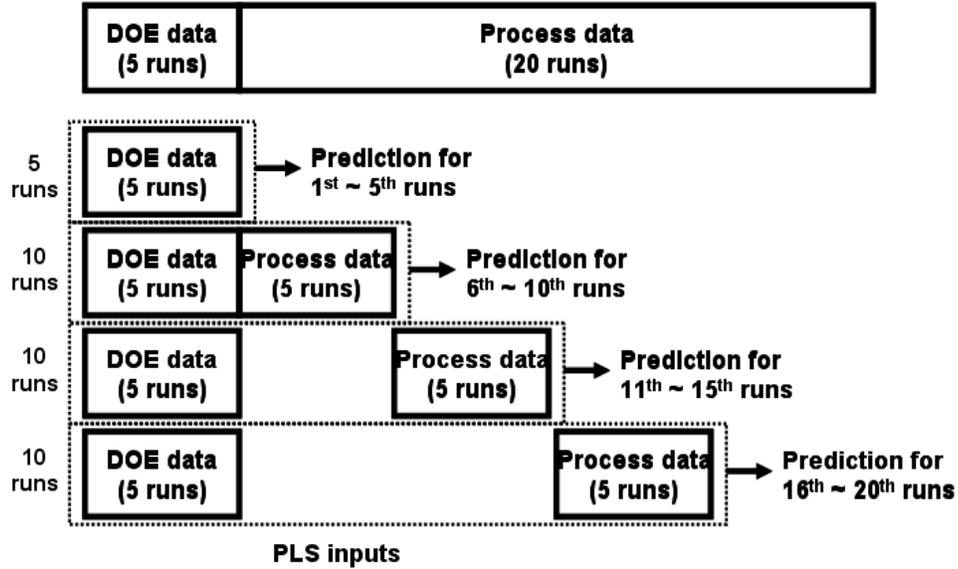


Figure 3.3: Selecting the data for updating the PLS model.

### 3.4.2 Recursive PLS

Recursive PLS (also known as Adaptive PLS) updates the PLS model whenever new measurements are available. The input and output matrices used for building the PLS model are augmented with the new data and a new model is calculated. A simplified way to update the model was proposed by Qin [99]; different types of recursive PLS algorithms such as block-wise and forgetting-factor-based algorithms were presented. The update methodology adopted in this work is shown in Figure 3.3. DOE data consists of five operating points shown in Figure 3.1. The value of the outputs at these operating points can be calculated using Equations 3.1 and 3.2 for the linear and the nonlinear model, respectively. Process drift and noise are not considered while calculating the outputs for DOE data in order to evaluate the prediction

methods in presence of unexpected drift and noise.

In this methodology, the PLS model is updated every five runs. The new model is based on five DOE runs and five latest process runs; for example, for predicting the outputs for process runs 11 through 15, the model is based on five DOE runs and the process runs 6 through 10. This way of selecting data matrices to update the PLS model ensures that both the DOE data and the latest process behavior contribute towards the new PLS model. The inclusion of DOE data provides the new model with drift-free and noise-free information about the process inputs and outputs, whereas the inclusion of the latest process runs tells the model about the current behavior of the process. Table 3.1 summarizes the characteristics of PLS, PLS with EWMA mean update, and recursive PLS.

Table 3.1: Summary of the key features of the prediction methods employed in this work.

Prediction method	PLS	PLS with EWMA mean update	Recursive PLS
Mean update	No	Yes	Yes
Model update	No	No	Yes
Decision options	Model inputs ( $v's$ or $v's+u's$ )	Model inputs and EWMA factor $\lambda$	Model inputs, EWMA factor $\lambda$ , and moving-window size

### 3.5 Results and Discussion

Simulations were carried out considering different kinds of unknown drifts and noise in process models represented by Equations 3.1 and 3.2. In order to compare the methods fairly in terms of their predictive power, the

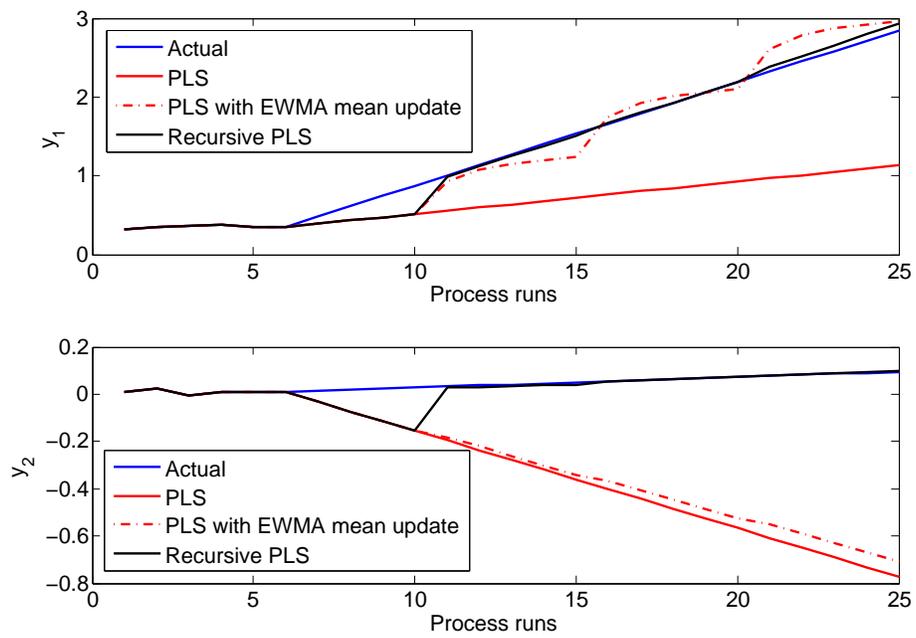


Figure 3.4: Predictions made by the PLS methods in the presence of a linear rise.

number of principal components required to build the PLS model for all three methods was set to two. Figure 3.4 compares the predictions made by PLS, PLS with EWMA mean update, and recursive PLS in presence of a linear drift and Gaussian white noise. The first five runs on x-axis represent DOE runs followed by 20 process runs. As the model updates start after five DOE runs and five process runs, the predictions made by all three methods overlap for process runs 6 through 10.

It can be observed that the PLS model is unable to track the linear drift due to no model update. For process output  $y_1$ , both PLS with EWMA mean update and recursive PLS track the actual outputs fairly well with recursive PLS performing slightly better. For output  $y_2$ , only recursive PLS tracks the outputs closely; PLS with EWMA mean update is unable to follow the drift because of no model update as explained below. The process gain matrix  $A$  shown in Equation 3.9 shows that the coefficient relating the input  $u_1$  with the output  $y_2$  is negative. When the process variables and the outputs are regressed to form the initial PLS model based on DOE data, the PLS coefficients for the output  $y_2$  have a negative value for the process variables that are a function of  $u_1$ . However, process variables  $v_3$  and  $v_4$  shown in Equations 3.5 and 3.6 are functions of not only  $u_1$ , but are also functions of the process drift vectors  $\eta_1$  and  $\eta_2$ . Therefore, in the case of a linear drift, the values of the process variables  $v_3$  and  $v_4$  also increase linearly leading to a negative slope in the predictions of  $y_2$ . On the other hand, recursive PLS tracks the linear drift for both the outputs because it updates the coefficients by rebuilding the model

every five runs.

Figures 3.5 and 3.6 compare the predictions made by PLS, PLS with EWMA mean update, and recursive PLS in presence of a ramp change and a linear drift starting arbitrarily during the process with Gaussian white noise, respectively. The results were found to be similar to those of linear drift. Recursive PLS and PLS with EWMA mean update provided fairly good predictions for  $y_1$ , but only recursive PLS was able to track the drifts for  $y_2$ .

The prediction performance of the three methods was quantified by calculating mean squared errors (MSE). Mean squared errors are the means of squared errors between the actual output values and the predicted values. Equation 3.12 shows the mathematical expression to calculate MSE.

$$MSE = \frac{\sum_{i=1}^n (y_{predicted,i} - y_{actual,i})^2}{n} \quad (3.12)$$

In Equation 3.12,  $y_{predicted,i}$  and  $y_{actual,i}$  correspond to the predicted and the actual values of the outputs for the  $i^{th}$  process run, respectively.  $n$  is the total number of process runs for which the predictions are made, which is equal to 20 in this study. Tables 3.2 and 3.3 provide the mean squared errors (MSE) in the predictions made by the PLS methods in presence of different kinds of drift for the outputs  $y_1$  and  $y_2$ , respectively. Similar behavior was observed for the nonlinear model shown in Equation 3.2.

While building a PLS model, it is critical to choose the input variables for the model carefully. Only the process inputs or both the process inputs

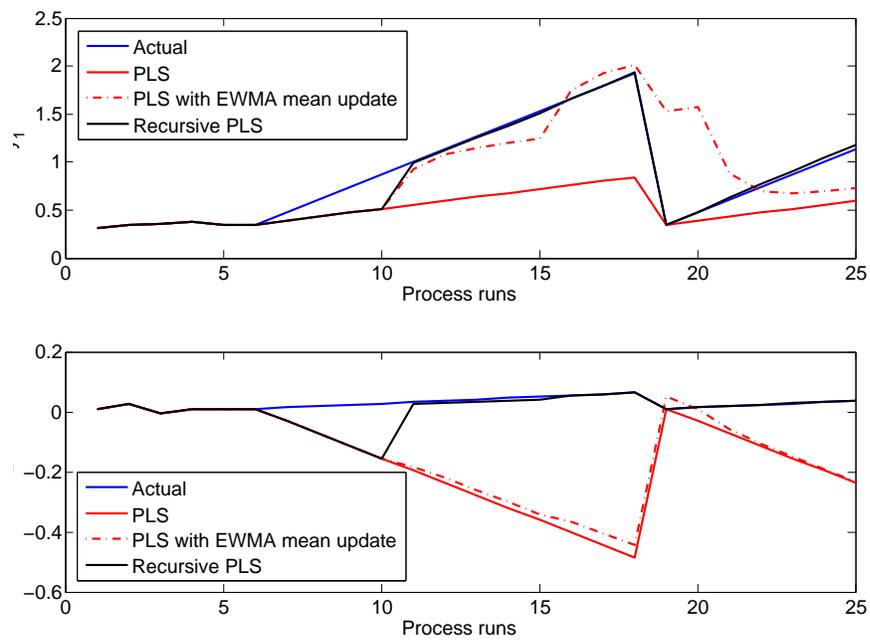


Figure 3.5: Predictions made by the PLS methods in the presence of a ramp change.

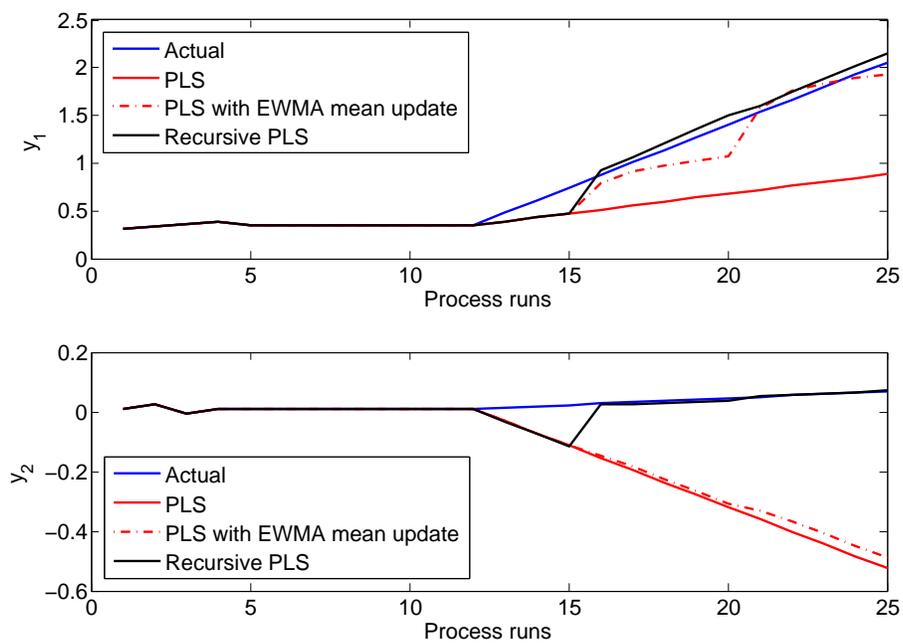


Figure 3.6: Predictions made by the PLS methods in the presence of a linear drift starting arbitrarily during the process.

Table 3.2: Summary of the prediction performance of the PLS methods for the output  $y_1$ . Recursive PLS provides the best predictions for all types of drifts.

Drift type	MSE (PLS)	MSE (PLS with EWMA mean update)	MSE (Recursive PLS)
No drift	0.0007	0.0005	0.0004
Linear rise	1.0046	0.0384	0.0137
Ramp change	0.3014	0.1686	0.0126
Arbitrary linear drift	0.3331	0.0174	0.0090

Table 3.3: Summary of the prediction performance of the PLS methods for the output  $y_2$ . Recursive PLS provides the best predictions for all types of drifts.

Drift type	MSE (PLS)	MSE (PLS with EWMA mean update)	MSE (Recursive PLS)
No drift	0.0003	0.0001	0.0001
Linear rise	0.2578	0.2220	0.0031
Ramp change	0.0773	0.0687	0.0031
Arbitrary linear drift	0.0855	0.0754	0.0015

and the process variables can be chosen as PLS inputs. Process variables of the model adapted in this work depend on the process inputs. Therefore, including the process inputs in the PLS inputs might be somewhat redundant and can prove detrimental to the prediction accuracy of the model. It was found that choosing only the process variables as PLS inputs provided lower MSE values as compared to the case where both the process inputs and the process variables were chosen as PLS inputs.

An important observation was made when the process was assumed to be infected with a large measurement noise (Gaussian white noise with zero mean and a large standard deviation). In such situations, PLS with EWMA mean update showed its ability to provide better predictions than recursive PLS. Recursive PLS tracked the noisy output measurements instead of pre-

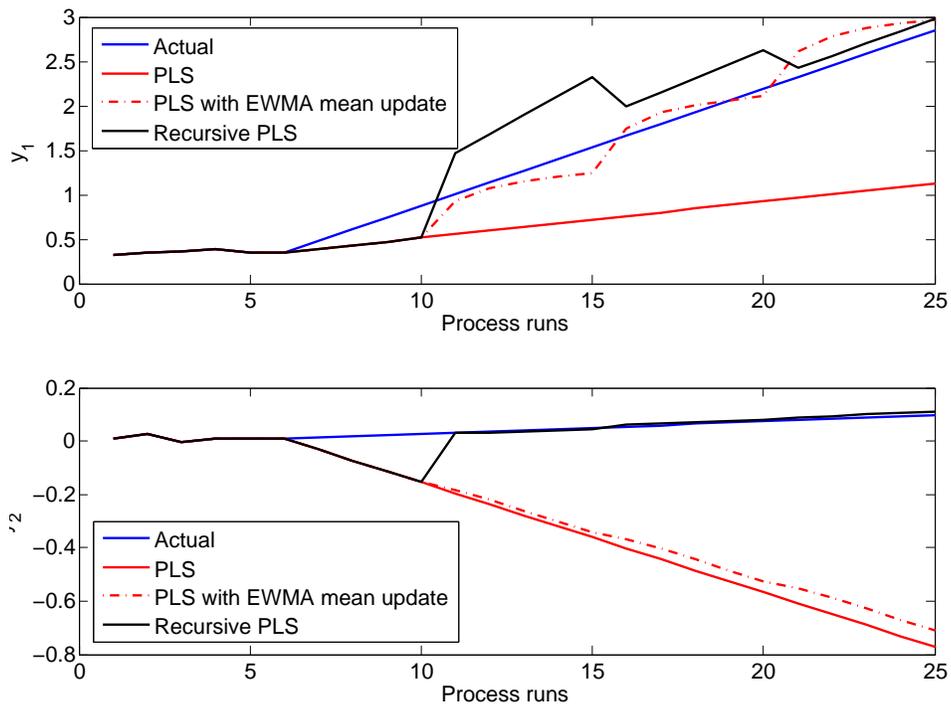


Figure 3.7: PLS with EWMA mean update provides better predictions than recursive PLS in the presence of large measurement noise.

dicting the actual process outputs. PLS with EWMA mean update, being more conservative in update, only updates the mean which is not significantly affected by the large measurement noise. Figure 3.7 compares the predictions made by the PLS methods in presence of a large measurement noise. It can be seen that PLS with EWMA mean update tracks the output  $y_1$  fairly well as compared to recursive PLS. For process output  $y_2$ , recursive PLS still provides the best estimate because of no model updating done in PLS with EWMA mean update method as discussed earlier in this section. Tables 3.4 and 3.5 provide MSE values for different noise sizes for the process outputs  $y_1$  and  $y_2$ , respectively.

Table 3.4: Mean squared error (MSE) values for the predictions of the output variable  $y_1$  for different sizes of measurement noise.

Measurement noise size (% of mean value of $y_1$ )	MSE (PLS)	MSE (PLS with EWMA mean update)	MSE (Recursive PLS)
10	1.0046	0.0384	0.0137
20	1.0046	0.0563	0.1270
30	1.0046	0.0973	0.2629

Table 3.5: Mean squared error (MSE) values for the predictions of the output variable  $y_2$  for different sizes of measurement noise.

Measurement noise size (% of mean value of $y_2$ )	MSE (PLS)	MSE (PLS with EWMA mean update)	MSE (Recursive PLS)
10	0.2578	0.2220	0.0059
20	0.2578	0.2398	0.0067
30	0.2578	0.2411	0.0165

### 3.6 Conclusions

In this work, two PLS variants (PLS with EWMA mean update and recursive PLS) were proposed as robust VM algorithms that can predict process outputs fairly well in the presence of unexpected process drifts and noise. Three types of process drifts were simulated and it was found that recursive PLS and PLS with EWMA mean update provided better predictions than traditional PLS algorithm for all drift types; recursive PLS being the best prediction method. However, in the presence of large measurement noise, PLS with EWMA mean update provided the best predictions as it is more conservative than recursive PLS in adapting to new measurements. These general guidelines reinforce VM technology by suggesting appropriate prediction methods when unexpected process changes occur. Other modeling features such as the selection of model inputs, tuning of the EWMA factor  $\lambda$ , and design of experiments were also discussed. The next step in this direction is to implement the proposed VM algorithms on industrial datasets infected with process drifts and noise.

## Chapter 4

# Detection of Faults in Virtual Metrology Sensors Using MPCA

The previous two chapters of this dissertation provided a comparison of various modeling techniques using both the industrial data and the simulated data. Chapter 2 compared multiple linear regression, principal component regression, partial least squares regression, recursive partial least squares regression, time series analysis, and Kalman filter estimation in terms of prediction accuracy using three industrial datasets. Kalman filter estimation was employed in a novel way to serve as a VM model for predicting outputs of a static process. Recursive partial least squares and Kalman filter estimation yielded better predictions than the rest of the methods.

Chapter 3 focused on three variants of partial least squares regression and provided simulation results using the data generated from a generic semiconductor process model present in VM literature. The process model incorporated the effect of different types of process drifts and noise. It was concluded that recursive partial least squares (also known as adaptive partial least squares) provides the best predictions as compared to other variants for a process suffering from drifts or shifts, while partial least squares with EWMA

update of the mean provided the best predictions if the process is corrupted with a large measurement noise.

## 4.1 Introduction

While building VM models in the earlier chapters, we assumed that the sensor data represent the true behavior of the process and are free from sensor faults. Any undesirable process behavior is referred to as a fault and can be further classified into sensor faults, actuator faults, and process faults (see Section 4.2 for more information). Any of these faults can arise while manufacturing a product. Sensor faults are the most relevant faults for VM as VM relies on the sensor data to predict the process outputs. A sensor fault means that the value of a process variable registered by the sensor is significantly different from the true value of the process variable.

The assumption of fault-free sensor data becomes invalid when malfunctioning of a sensor corrupts the sensor data. The probability of the occurrence of a sensor fault in a process increases linearly with the number of installed sensors. Currently, semiconductor manufacturing processes deploy a large number of sensors to monitor the process behavior, which leads to a greater risk of the occurrence of sensor faults. When a sensor fault occurs, the corresponding sensor data are erroneous and do not represent the true behavior of the process. The quality of sensor data, which serve as inputs for VM models, has a direct effect on the quality of predicted values. In the presence of faulty input data, an accurate VM model will provide erroneous predictions

for the outputs. This situation is known as Garbage-In-Garbage-Out in the process modeling terms. For using VM effectively, we need to make sure that the data to be fed into the VM model are free from faults. The objective of this chapter is to detect the sensor faults using MPCA. It is possible that multiple sensors are simultaneously faulty, but the likelihood of the occurrence of simultaneous multiple sensor faults is fairly low as the sensors are independent physical entities. In this chapter, we will focus on single sensor faults only.

In this chapter, we will provide a literature review of the popular fault detection and identification approaches. Specifically, we will first present fault detection using principal component analysis (PCA). PCA is able to detect faults for a two-dimensional data matrix only, the two dimensions being time and process variables in most cases. However, the data collected from semiconductor manufacturing processes are three-dimensional, with an additional dimension for different wafers. Hence, PCA cannot be directly applied for fault detection on data collected from a semiconductor manufacturing process. Instead, multiway principal component analysis (MPCA) is employed to address this limitation of PCA (see Section 4.2.2 for details).

## 4.2 Fault Detection Approaches

The fault detection approaches present in the relevant literature can be classified into two main categories: model-based and data-driven approaches. Model-based fault detection systems [8, 30, 33, 53, 55, 126] rely on dynamic models that are physically-based or empirically-defined. Model-based ap-

proaches can be further divided into two kinds of approaches. The first kind of approaches utilize state estimators and the concept of analytical redundancy, where residuals are derived by calculating the difference between the actual outputs of the monitored system and the outputs obtained from a mathematical model and a state estimator. The second kind of approaches utilize analytical redundancy relations (ARRs) (or parity equations) [5, 118], where residuals are obtained through differential-algebraic relationships that are generated by using a mathematical model. A major disadvantage of these model-based approaches is their reduced reliability when a process-model mismatch exists, which might be a result of the uncertainty present in the model parameters. Although a few improvements have been suggested in the literature [22, 29, 31, 142], model-based fault detection approaches are not mature and are undergoing active research. An interested reader is referred to a comprehensive review of nonlinear model-based fault detection methods by Castillo [12].

Data-driven approaches [30, 124, 125] use signal processing techniques on plant data to extract characteristic parameters and detect abnormal conditions. Data-driven approaches can be further divided into two kinds of approaches. First, computational intelligence methods [4, 90, 116, 150, 154] (also known as artificial intelligence approaches) incorporate heuristics and reasoning in the fault detection decisions. Popular computational intelligence methods reported in literature comprise of soft computing [95], neural networks [110, 117, 150, 154], fuzzy logic [54, 79, 90, 96], expert systems [3], pat-

tern recognition, and machine learning [122]. An important disadvantage of some of these computational techniques is their inability to provide physical reasoning for fault detection results because of their black box nature. The second kind of approaches consist of statistical process monitoring (SPM) techniques, which mostly employ principal component analysis (PCA) [2, 100, 143] and partial least squares (PLS) [76, 87, 143]. These techniques have the ability to handle a large number of measured process variables by compressing them into fewer directions so that the operating conditions can be visualized in lower dimensional plots.

The multivariate SPM methods such as multiway PCA (MPCA) and multiway PLS (MPLS) have long been used in monitoring batch processes in the traditional chemical and petrochemical industries [65, 67, 68, 76, 84–86, 103, 130]. Qin [100] provides an excellent review and analysis of the past and recent SPM methods. The characteristics associated with chemical batch processes, such as unequal batch and/or step length, unsynchronized or misaligned batch trajectory, and multimodal batch trajectory distribution, usually result in non-Gaussian distributed data and deteriorate the monitoring performance of MPCA and MPLS. To address these challenges, various data preprocessing steps are usually required for the MPCA and MPLS methods to achieve satisfactory monitoring performance. These preprocessing steps, including trajectory alignment/warping, trajectory mean shift, and data unfolding, are often performed off-line and could be difficult to automate.

In the last few decades, the multivariate SPM methods have been

adopted to monitor semiconductor manufacturing processes. The most commonly used multivariate SPM methods in the semiconductor industry are MPCA [23, 134, 135, 146–148] and MPLS [23, 137]. Similar to chemical process monitoring, two steps are involved in SPM for semiconductor processes: (1) correlation information among different variables is extracted by applying dimension reduction techniques to measurements of multiple variables; (2) fault detection is performed by examining whether a test sample follows the same correlation pattern exhibited by the normal training samples. The details of PCA and MPCA approaches for fault detection are presented in the following two subsections. Later in Section 4.4, MPCA will be implemented to detect faults in a benchmark dataset.

#### **4.2.1 Principal component analysis (PCA)**

In Chapter 2, we presented PCA method in detail and employed it as a VM model to predict output values from a given set of inputs. We learnt that PCA can extract and rank data correlations within a data matrix according to their importance. The amount of variance captured by a particular principal component was found to be equal to the ratio of the corresponding eigenvalue to the sum of all the eigenvalues. In industrial applications, where hundreds of process variables are routinely measured, the critical features of the plant are captured by relatively fewer strong correlations. In other words, instead of observing all the process variables, monitoring the important correlations is a more efficient way. This enables us to reject most of the process noise as it is

not important from process monitoring point of view.

PCA extracts important correlations from the data matrix and serves as an efficient way to monitor industrial processes. Using historical data, normal behavior of the process is recorded and the acceptable limits of the fault detection indices are calculated. When new data arrive, we can map it into the new principal component subspace and calculate the indices. If the indices are outside the acceptable limits, we conclude that the observed data are faulty.

Suppose process data are stored in a matrix  $X_{raw}$ ,  $n \times m$ , where  $n$  is the number of observations and  $m$  is the number of process variables/sensors [14, 100]. After scaling the matrix to zero mean and unit variance for each column (process variable/sensor), the data matrix  $X$  is decomposed by PCA as shown in Equation 4.1.

$$X = TP + \tilde{T}\tilde{P} \quad (4.1)$$

where  $T$  and  $P$  represent the scores and the loading vectors that explain the important process variations, corresponding to large eigenvalues of the covariance matrix of  $X$ ;  $\tilde{T}$  and  $\tilde{P}$  represent the scores and the loading vectors dominated by process noise, corresponding to small eigenvalues of the covariance matrix of  $X$ . The vector space spanned by important loading vectors  $P$  is referred to as principal component subspace while the space spanned by  $\tilde{P}$  is known as residual subspace.

The choice of the number of principal components that go to the principal subspace and the residual subspace is a critical issue. Choosing too few principal components for the principal subspace might not capture all the important correlations present in the process data, whereas choosing too many principal components increases the risk of incorporating more noise in the chosen principal components. The number of principal components chosen is bounded by a minimum value of one and a maximum value equal to the number of process variables in the data matrix  $X$ . One can utilize process knowledge to make a wise decision for choosing the number of principal components. For PCA, there are numerous methods available for this choice; according to the published literature [48, 123, 152], cross validation [136], variance of the reconstruction error [123], and parallel analysis are considered to be robust and reliable methods.

Determination of an appropriate number of principal components required and subsequently obtaining the scores and the loading vectors constitute the model building part of PCA-based fault detection approach. The next part will guide us how to quantify the degree of similarity of the newly observed data with the historical data and conclude whether a fault has occurred or not.

When a new observation is collected, it is compared with the historical data of the process to find the similarities between the two. In fault detection using PCA, first the newly collected data are normalized using the mean and standard deviation of the historical data as shown in Equation 4.2. This

normalization is required to calibrate the new data so that only the relative deviations from the historical data are considered while detecting faults.

$$x = \frac{x_{raw} - x_{mean}}{x_{std}} \quad (4.2)$$

where  $x$  is the new observation after normalization,  $x_{raw}$  is the raw (unnormalized) observation,  $x_{mean}$  consists of the mean values of the process variables calculated from historical data, and  $x_{std}$  consists of the standard deviations of the process variables calculated from historical data.

After normalization, the new observation is projected to the principal component subspace and the residual subspace. This can be thought as splitting up the observed data into a part that can be explained by the PCA model and a part that represents process noise. In Equations 4.3 - 4.5,  $\hat{x}$  and  $\tilde{x}$  stand for the projections of  $x$  on the principal component subspace and the residual subspace, respectively.  $P^T$  represents the transpose of the loading vectors  $P$  and  $I$  is the identity matrix.

$$x = \hat{x} + \tilde{x} \quad (4.3)$$

$$\hat{x} = PP^T x \quad (4.4)$$

$$\tilde{x} = (I - PP^T)x \quad (4.5)$$

Once the projection of the new observation on the principal component subspace and the residual subspace has been obtained, we can calculate certain

fault detection indices that quantify the amount of dissimilarity between the observed data and the historical data. Squared prediction error, SPE (or Q statistic) and Hotelling's  $T^2$  statistic are the two most commonly employed fault detection indices. Other indices that have been proposed in the literature are summarized in a resourceful review paper by Qin [100]. These indices are combined index  $\phi$  [106, 145], Hawkins' statistic [56], and Mahalanobis distance [77]. In this work, we will be focusing on SPE,  $T^2$ , and  $\phi$  fault detection indices.

The SPE index quantifies the projection of the observed data on the residual subspace. A large SPE index would mean that the projection of the observed data in the residual subspace is large, indicating that a large portion of the observed data cannot be explained by the principal component subspace. This situation enables us to conclude that the observed data are faulty as they behave quite different from the historical data. Equation 4.6 provides the mathematical expression for the calculation of SPE.

$$SPE = \|\tilde{x}\|^2 = \|(I - PP^T)x\|^2 = x^T \tilde{C}x \quad (4.6)$$

In order to use the calculated SPE value for detecting faults, we need to calculate a SPE limit. If the calculated SPE value is more than this limit, it can be concluded that the observed data are faulty; otherwise, the data are quite similar to historical data and are not faulty. A mathematical expression for calculating the SPE limit,  $\delta_\alpha^2$  was developed by Jackson and Mudholkar [57] and is shown in Equation 4.7. Here,  $\delta_\alpha^2$  is the upper SPE limit with a

confidence level  $\alpha$  and  $c_\alpha$  is the normal deviate corresponding to the upper  $1 - \alpha$  percentile.

$$\delta_\alpha^2 = \theta_1 \left( \frac{c_\alpha \sqrt{2\theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0} \quad (4.7)$$

where,

$$\theta_i = \sum_{j=l+1}^m \lambda_j^i \quad i = 1, 2, 3 \quad (4.8)$$

$$h_0 = 1 - \frac{2\theta_1 \theta_3}{3\theta_2^2} \quad (4.9)$$

In Equation 4.8,  $l$  refers to the number of principal components in principal component subspace,  $m$  is the total number of variables in data matrix  $X$ , and  $\lambda_j$  is the eigenvalue corresponding to the  $j^{th}$  principal component.

The variation in the principal component subspace is measured by Hotelling's  $T^2$  statistic (simply known as  $T^2$  statistic). Equation 4.10 provides the expression to calculate the  $T^2$  statistic. In this equation,  $P$  consists of the loading vectors,  $\Lambda$  stands for the eigenvalues of the covariance matrix of data  $X$ , and  $D = P\Lambda^{-1}P^T$ .

$$T^2 = x^T P \Lambda^{-1} P^T x = x^T D x \quad (4.10)$$

Assuming that the process behaves normally and the data follow multivariate Gaussian distribution, the relationship between  $T^2$  index and F distribution is given by Equation 4.11. Here,  $F_{l, n-l}$  represents an F distribution with

$l$  and  $n-l$  degrees of freedom,  $l$  refers to the number of principal components in the principal component subspace, and  $n$  is the number of observations in the data matrix  $X$ .

$$\frac{n(n-l)}{l(n^2-1)}T^2 \sim F_{l,n-l} \quad (4.11)$$

Equation 4.11 assumes that the population mean and covariance are estimated from the data. The upper control limit for  $T^2$  for a confidence level  $\alpha$ ,  $T_\alpha^2$ , is calculated as shown in Equation 4.12. If the  $T^2$  index of the observed data calculated from Equation 4.10 are greater than  $T_\alpha^2$ , it can be concluded that a fault has occurred. In the case when population mean is known and only covariance is estimated from the data, Equation 4.12 can be modified to Equation 4.13 [100].

$$T_\alpha^2 = \frac{l(n^2-1)}{n(n-l)}F_{l,n-l;\alpha} \quad (4.12)$$

$$T_\alpha^2 = \frac{l(n-1)}{n-l}F_{l,n-l;\alpha} \quad (4.13)$$

If the number of observations,  $n$  is sufficiently large so that the population mean and covariance can be accurately estimated from the data, the  $T^2$  index can be approximated by the  $\chi^2$  distribution with  $l$  degrees of freedom. In process monitoring, this is usually the case and the  $T_\alpha^2$  value is often calculated using Equation 4.14.

$$T_\alpha^2 = \chi_{l;\alpha}^2 \quad (4.14)$$

Yue and Qin [144, 145] proposed a combined index for fault detection, which combines SPE and  $T^2$  indices as shown in Equation 4.15. The details of upper control limit for  $\phi$  can be found in [145].

$$\phi = \frac{SPE(x)}{\delta_\alpha^2} + \frac{T^2(x)}{\chi_{l;\alpha}^2} = x^T \Phi x \quad (4.15)$$

where

$$\Phi = \frac{I - PP^T}{\delta_\alpha^2} + \frac{P\Lambda^{-1}P^T}{\chi_{l;\alpha}^2} = \frac{\tilde{P}\tilde{P}^T}{\delta_\alpha^2} + \frac{P\Lambda^{-1}P^T}{\chi_{l;\alpha}^2} \quad (4.16)$$

#### 4.2.2 Multiway principal component analysis (MPCA)

In the last section, we showed how PCA can be utilized for fault detection purposes. We also introduced three commonly used fault detection indices, SPE,  $T^2$ , and  $\phi$  along with their upper control limits. It can be recalled that the data matrix  $X$  considered was two-dimensional with the observations and process variables being the two dimensions. However, in the case of batch processes like semiconductor manufacturing, an additional dimension for different batches/wafers needs to be included. This gives rise to a three-dimensional data matrix.

The data associated with the semiconductor manufacturing processes often has unequal batch lengths as shown in Figure 4.1 and Figure 4.2a. In other words, data are available for different number of time stamps for different batches. Closed-loop control is the primary cause of unequal batch lengths.

To control and minimize the variation in the etch depth, semiconductor fabs implement run-to-run controllers, which adjust the etch times after every run. This adjustment of etch times leads to the variation in the durations of etch steps and the overall wafer processing time, which results in the misalignment of wafer trajectories. The preventive maintenance (PM) events such as *in-situ* cleaning and part replacement cause shifts in the states of equipment giving rise to multimodal batch trajectory distribution. Because of these issues with data collected from semiconductor manufacturing processes, PCA approach cannot be applied directly and the data need to be preprocessed first as described below.

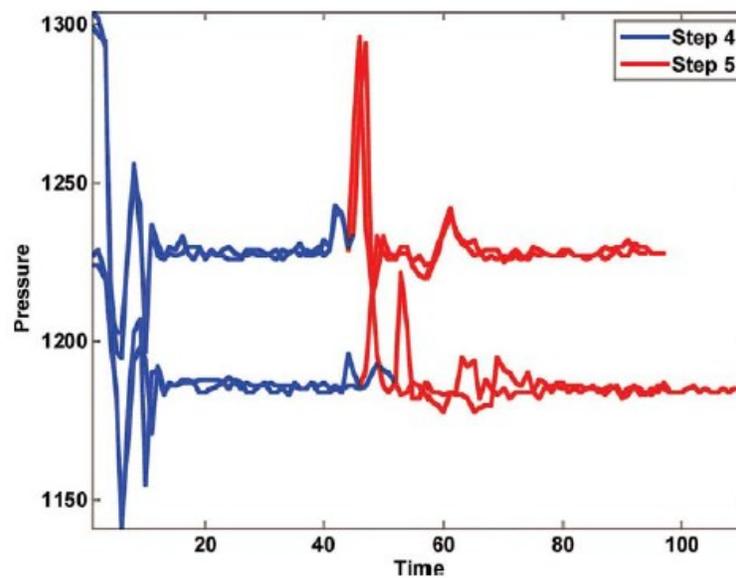


Figure 4.1: Characteristics of semiconductor manufacturing processes. This figure shows unequal batch lengths and unsynchronized/misaligned and multimodal trajectories of a process variable, pressure, for two etch steps for four processed wafers.

Several steps of data preprocessing are required before applying PCA to three-dimensional data collected from a semiconductor manufacturing process. First, trajectory alignment is applied to make batches synchronized using dynamic time warping (DTW) [151, 153]; then, it is ensured that the batch lengths are equal; finally, trajectory mean shift (e.g., subtracting trajectory mean from observed data) is applied to make all trajectories follow a unimodal distribution. The preprocessed three-dimensional matrix (shown in Figure 4.2b) needs to be further unfolded into a two-dimensional matrix, where each row will now represent a batch of the preprocessed three-dimensional matrix. The two-dimensional matrix is shown in Figure 4.2c, which is ready to be analyzed by PCA. The above procedure is termed multiway PCA or MPCA. If the original three-dimensional matrix had a size of  $I \times J \times K$  (corresponding to  $I$  batches,  $J$  process variables, and  $K$  time stamps), the unfolded two-dimensional matrix will have a size of  $I \times JK$ . This type of unfolding is known as batch-wise unfolding as it helps us to analyze the differences between the batches and allows us to detect faulty batches. Other types of unfolding are also possible and have been discussed by Zhang [151].

After preprocessing, the resulting two-dimensional matrix is analyzed using the procedure outlined in Section 4.2.1. In this chapter, we will be implementing MPCA to detect faults in a benchmark dataset, which is presented in the next section.

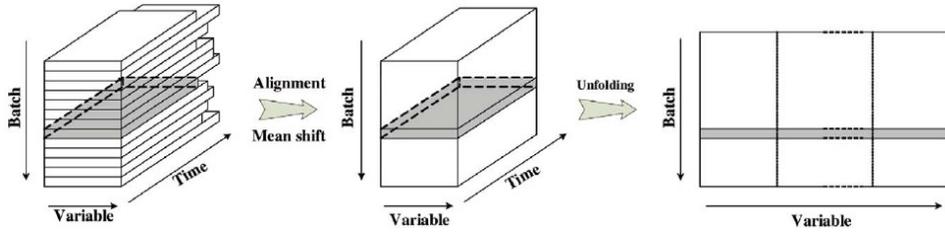


Figure 4.2: Data preprocessing of three-dimensional data collected from a semiconductor manufacturing process to obtain a two-dimensional matrix that can be analyzed using PCA. This procedure is known as multiway PCA or MPCA.

### 4.3 Details of the Benchmark Dataset

The benchmark dataset used in this work was collected from an Aluminium stack etch process performed on a Lam 9600 plasma etch tool at Texas Instruments Inc. [132, 134]. This process etches a TiN/Al0.5% Cu/TiN/oxide stack with an inductively coupled  $BCl_3/Cl_2$  plasma. The key parameters of interest are the linewidth of the etched Al line, uniformity across the wafer, and the oxide loss. The dataset consists of 108 normal wafers processed during three experiments which were several weeks apart and 21 wafers with intentionally induced faults processed during the same experiments. The complete process recipe consists of six steps but only two etch steps (main etch and over etch) are considered in this study. There is a significant difference in the means and variances of the monitored variables among different experiments because of process drifts and maintenance events.

The original dataset contains 40 variables including process setpoints, measured variables, and controlled variables such as gas flow rates, chamber

pressure, and RF power. The inclusion of irrelevant variables in the analysis degrades the performance of a fault detection technique. Therefore, in this work only 19 non-setpoint process variables are used for fault detection as suggested by Wise et al. [134]. The physics and chemistry of the problem suggest that these variables should be relevant to the process and the final product state. The data for these process variables were collected by the installed sensors that record the values at an interval of 1 second. These process variables are listed in Table 4.1.

Table 4.1: Process variables used for fault detection in an Aluminium stack etch process.

$BCl_3$ flow rate	$Cl_2$ flow rate	RF bottom power
RF bottom reflected power	End-point detector	Helium pressure
Chamber pressure	RF tuner	RF load
RF phase error	RF power	RF impedance
TCP tuner	TCP phase error	TCP impedance
TCP top power	TCP reflected power	TCP load
Vat valve		

RF stands for Radio Frequency; TCP stands for Transformer Coupled Plasma.

Each of the 21 intentionally induced faults present in the original dataset was created by changing the value of one of the process variables (say  $i^{th}$  process variable) to a value different than its setpoint value. The processing of a wafer with the changed value of the  $i^{th}$  process variable leads to corresponding changes in the other process variables, according to the relationship of the  $i^{th}$  process variable with the other process variables. In order to make the detection possible, the value of the intentionally altered  $i^{th}$  process variable was reset to the original setpoint value. If the value of the  $i^{th}$  process variable

was not reset to the original setpoint, the same relationship of the  $i^{th}$  process variable with other variables would hold and would not indicate a fault. By resetting the value to the original setpoint, the relationship of the  $i^{th}$  process variable with others is now different and the wafer corresponding to these data can be detected as faulty. The details of these faults are provided in Wise et al. [132, 134].

These faults were introduced with a limited intention of comparing several methods in terms of their fault detection performance. He [47] performed a fault detection study on this benchmark dataset. These intentionally induced faults serve the purpose of comparing different fault detection methods successfully, but are not suitable for performing fault identification. Fault identification aims at finding the process variable/process variables which caused a fault in a wafer/batch and serves as the basis for obtaining reconstructed data. As the values of the intentionally altered process variables were reset to the original set point values, it is highly unlikely that the altered process variables would be identified as the ones causing the faults. Rather, the other process variables, whose values changed because of their relationship with the altered process variables, have more chance of being identified as the ones responsible for the faults. Hence, the faults induced in the benchmark dataset are not suitable for performing fault identification.

We may recall that the original dataset contained data for 108 normal wafers. A significant amount of data for one of the wafers, wafer number 56, were missing and consequently, it was not considered in this work. In order

to perform fault identification effectively, artificial faults were introduced in the data corresponding to 107 normal wafers. For example, a sensor fault in the mean value of a process variable was simulated by adding a constant bias to the setpoint value of the process variable. These artificial faults appear more promising than the ones in the original dataset for performing fault identification as the values of the altered process variables are not reset to their setpoint values. Using the artificial faults, the chance of the altered process variable getting identified as the one causing the fault is much more than its chance while using the faults present in the original dataset.

The next section presents the implementation of MPCA to detect the artificial faults discussed in this section. Effects of EWMA filtering and the number of principal components held in the principal component subspace on the fault detection performance of MPCA are also discussed.

#### **4.4 Fault Detection Using MPCA**

As discussed in the last section, artificial faults were created using data for normal 107 wafers of the benchmark dataset. The processing times of these wafers were quite different from each other and are plotted in Figure 4.3. It should be noted that these times represent the duration of the main etch and over etch steps only. Preprocessing was carried out to ensure that all the batches are of same length, which will be the equal to the length of the batch with the shortest processing time (95 for this dataset). After preprocessing, three-dimensional data were unfolded to form a two-dimensional matrix. For

this dataset, the three-dimensional data are of size 107 (wafers) X 19 (process variables) X 95 (time stamps), which was unfolded to form a two-dimensional matrix of size 107 X 19\*95. The first 95 columns of the two-dimensional matrix hold values of the first process variable for 95 time stamps; the next 95 columns contain values of the second process variable for 95 time stamps and so on. Finally, the data in the two-dimensional matrix were normalized using Equation 4.2.

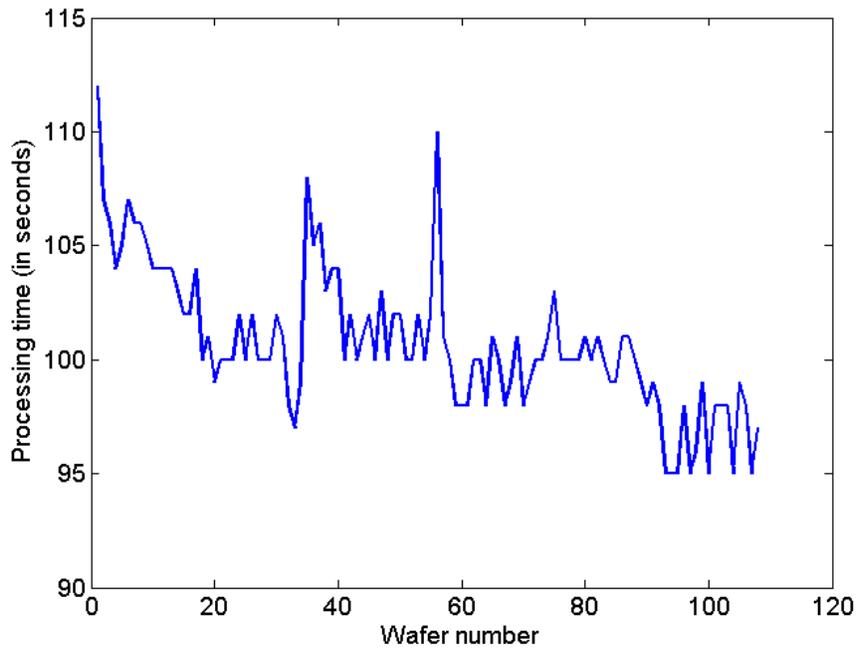


Figure 4.3: Processing times for the normal (fault-free) wafers in the benchmark dataset. These times represent the duration of the main etch and over etch steps only. The data need to be preprocessed before detecting faults using MPCA.

#### 4.4.1 Fault detection results using MPCA

After obtaining the normalized and preprocessed data for the normal 107 wafers of the benchmark dataset, a PCA model was built as described in Section 4.2.1. The control limits for the fault detection indices, SPE,  $T^2$ , and  $\phi$  were also calculated. Faulty data are required to meet our objective of performing fault detection using MPCA. Faulty data were created by introducing artificial faults in the data for 107 normal wafers. Specifically, for this section, constant biases were added to the normal/setpoint values of the process variables to simulate sensor faults. Addition of a constant bias to the values of a process variable for all time stamps is equivalent to adding a bias to the mean value of the process variable. In this study, 19 such sensor faults were introduced by adding a constant bias to one of the process variables at a time. In other words, the first sensor fault was created by adding a constant bias to the first process variable, the second sensor fault was created by adding a constant bias to the second process variable, and so on (see Table 4.1 for the list of 19 process variables). The sensor faults were introduced in this manner to investigate the detectability of the faults in each of these 19 process variables using MPCA. Sensor faults with a constant bias are the simplest and the most frequently occurring type of sensor faults; other kinds of sensor faults will be discussed later in Section 5.2.3. We are considering the simplest sensor faults first in order to observe the best fault detection performance of MPCA. If any other fault detection method provides better detection for these faults, we can conclude that it is a superior fault detection method than MPCA.

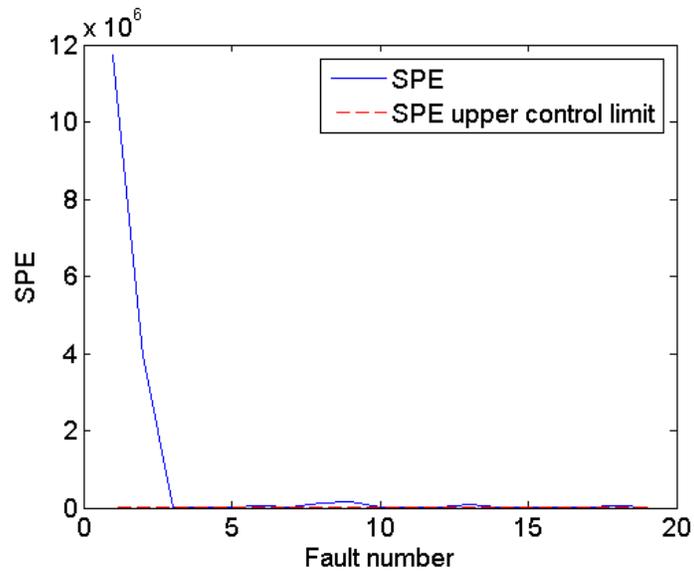


Figure 4.4: Fault detection using MPCA. 14 faults present in the mean values of the process variables are detected.

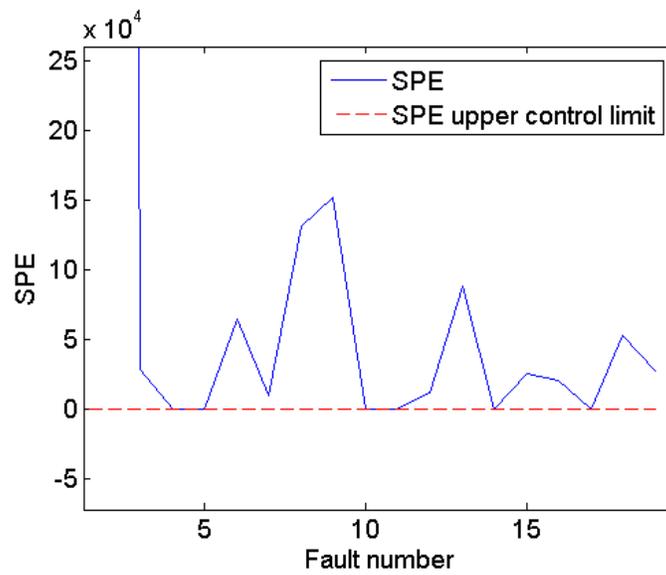


Figure 4.5: Fault detection using MPCA. Zoomed-in view of Figure 4.4.

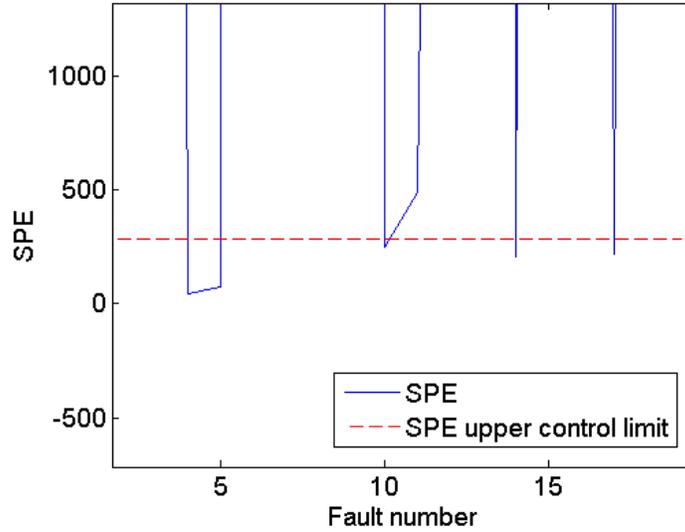


Figure 4.6: Fault detection using MPCA. Zoomed-in view of Figure 4.5.

Figures 4.4-4.6 show the fault detection results for 19 sensor faults using MPCA. Three fault detection indices, SPE,  $T^2$ , and  $\phi$  were calculated for these 19 faulty wafers along with the control limits of the indices. As SPE index quantifies the projection of the observed data on the residual subspace, more faults were detected using the SPE index as compared to the  $T^2$  index. In other words, SPE is a more useful index than the  $T^2$  index for detecting sensor faults. In this study, SPE,  $T^2$ , and  $\phi$  indices were able to detect 14, 13, and 14 faults out of 19 sensor faults, respectively.

In Figure 4.4, it is evident that SPE indices of the first and the second faults are very large as compared to those of the rest of the faults. It can be recalled that the first and the second faults were introduced by adding constant biases to the first and the second process variables, respectively. The first two

process variables had significantly smaller standard deviations as compared to the rest of the process variables, resulting in large values after normalizing using Equation 4.2. As the size of the normalized values of the process variables directly affects the SPE indices (see Equation 4.6), the SPE indices of the first two faults are found to be relatively larger than those of the other faults.

#### **4.4.2 Effect of fault magnitude/size on fault detection performance**

The magnitudes of the introduced faults were chosen to be 20 % of the mean values of the process variables. For example, to simulate the first fault, a constant bias of magnitude equal to 20 % of the mean value of the first process variable was added to the normal/setpoint value of the first process variable. The magnitude of the introduced fault affects the detection performance of a fault detection method. The influence of the magnitudes of sensor faults on the fault detection performance of MPCA is studied next. Sensor faults with magnitudes equal to 1%, 5%, 10%, and 40% of the mean values of the process variables were simulated. The number of faults of different magnitudes detected by SPE,  $T^2$ , and  $\phi$  indices are provided in Table 4.2. It is apparent that the faults with large magnitudes are detected more easily as compared to the ones with relatively smaller magnitudes. Large fault magnitudes lead to large values of fault detection indices, which overshoot the upper control limits indicating the occurrence of a fault.

Table 4.2: Influence of the fault magnitude/size on the fault detection performance of MPCA (19 sensor faults were introduced). The fault magnitude is expressed in terms of the percentage of the mean value of process variables. Faults with larger magnitudes are detected more easily than the faults with relatively smaller magnitudes.

Fault magnitude/size	Faults detected by SPE	Faults detected by $T^2$	Faults detected by $\phi$
1	9	2	6
5	13	9	13
10	13	11	13
20	14	13	14
40	15	13	14

#### 4.4.3 Effect of confidence level ( $\alpha$ ) on fault detection performance

A confidence level of 95 % was used to calculate the control limits of the fault detection indices for the simulation results provided in Figure 4.4.1 and Table 4.2. This means that when the value of a fault detection index for a wafer is more than its control limit, we can conclude with 95 % confidence that the wafer is faulty. For a Gaussian distribution, 95 % of the data points fall within two standard deviations from the mean. In other words, the wafers with a value of the fault detection index greater than two standard deviations away from the mean are considered as faulty. The performance of a fault detection method is a function of confidence level value ( $\alpha$ ). Particularly, the value of confidence level is set to calculate the control limits of the fault detection indices as shown in Equation 4.7 for calculating SPE control limit and Equations 4.12-4.14 for  $T^2$  control limit. A higher value of confidence level will leave a smaller room for the occurrence of false alarms. So, a higher confidence level typically corresponds to higher control limits. Due to higher control limits, less number

of faults will be detected. In essence, a higher confidence level will lead to detection of a lower number of faults, but these detections are very likely to be the actual faults, not false alarms. Table 4.3 shows the calculated control limits of the fault detection indices and the number of faults detected out of 19 sensor faults for different values of confidence level ( $\alpha$ ). The magnitudes of the introduced faults were chosen to be 20 % of the mean values of the process variables for these simulations.

Table 4.3: Influence of the confidence level ( $\alpha$ ) on the fault detection performance of MPCA (19 sensor faults were introduced). A higher confidence level will lead to detection of lesser number of faults because of higher control limits, but these detections are very likely to be the actual faults, not false alarms.

Confidence level (%)	70	80	90	95	99
SPE control limit	224.15	238.57	261.09	281.74	325.24
$T^2$ control limit	86.12	90.41	96.58	101.88	112.33
$\phi$ control limit	1.57	1.63	1.71	1.77	1.82
Faults detected by SPE	15	15	14	14	14
Faults detected by $T^2$	16	15	13	13	12
Faults detected by $\phi$	16	15	14	14	14

#### 4.4.4 False alarms

A false alarm refers to a situation when the process is operating under normal conditions, but the fault detection system indicates the presence of a fault. In other words, the values of fault detection indices are found to be above the control limits even when the process is operating normally. Besides being sensitive to faults, generating very few false alarms is an essential feature of a reliable fault detection method. Ideally, a fault detection system must show zero false alarms. Generating a large number of false alarms is detrimental

to the economics of a process as false alarms lead to wasting of resources and reducing the process throughput.

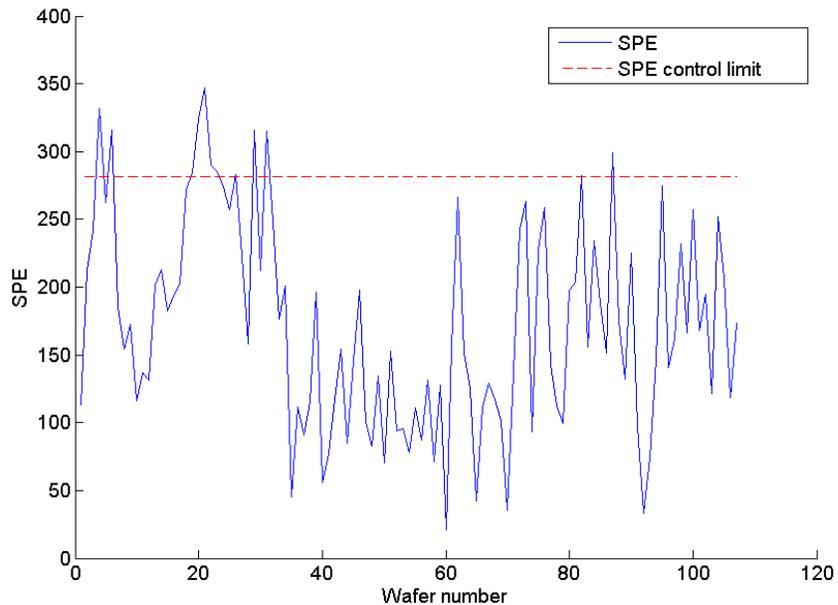


Figure 4.7: SPE indices for 107 normal wafers using MPCA. Twelve false alarms were observed, i.e., the SPE indices were found to be more than the SPE control limit for twelve normal wafers.

We can employ the PCA model built in Section 4.4.1 to observe whether MPCA causes false alarms. Instead of creating artificial faulty data as done in Section 4.4.1, the normal data for 107 were used to investigate if any false alarms were raised. MPCA wrongly recognized twelve normal wafers as faulty wafers, i.e., twelve false alarms were raised as shown in Figure 4.7. A confidence level of 95% was used to calculate the control limits for the fault detection indices, which means 5 out of 100 detections might be false alarms. MPCA

raised many more false alarms than those expected for a confidence level of 95%.

#### 4.4.5 Limitations of MPCA

In Section 4.4.1, MPCA was employed to detect artificially induced faults in the benchmark dataset. Several preprocessing steps were required before the data were sufficient for building the PCA model. MPCA provided fairly good fault detection results, but was unable to detect all the artificially induced faults (19 sensor faults were induced in this study). Section 4.4.4 was presented to assess the performance of MPCA when applied to normal data. MPCA raised 12 false alarms for the data of 107 normal wafers, which were many more than those expected for a confidence level of 95%. This section will discuss the limitations of MPCA, which led to the fault detection results presented in Sections 4.4.1 and 4.4.4.

The first limitation of MPCA is that the data need to be preprocessed before an effective PCA model can be built. Commonly used preprocessing steps for the data collected from semiconductor manufacturing include making the batch lengths equal, batch trajectory synchronization/alignment, and mean shift to ensure that data follow a unimodal distribution. Typically, these data preprocessing steps improve monitoring performance by making the preprocessed data conform more closely to a multivariate Gaussian distribution. However, there are some disadvantages associated with the preprocessing of data. First, the data preprocessing procedures such as trajectory align-

ment/synchronization often require human intervention, which makes the automation of process monitoring difficult. Second, the data preprocessing may distort process information and result in deteriorated monitoring performance [153].

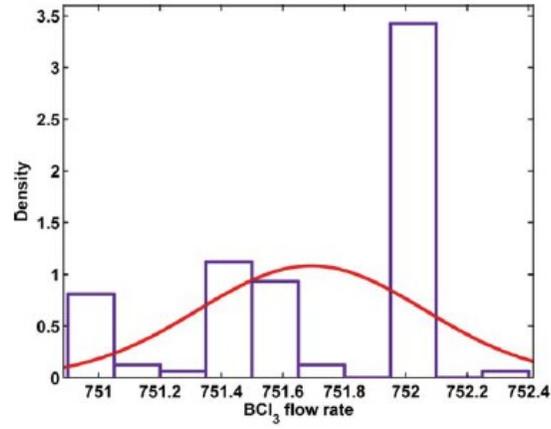
In the high-mix production environment of semiconductor manufacturing fabs, it is possible to have hundreds of different products running on the same piece of equipment. If data unfolding and alignment are needed to perform fault detection, different PCA models are needed for different products because different products usually have different batch durations. As a result, the data preprocessing steps required in the high-mix production environment make the model building and maintenance extremely labor intensive. In addition, due to fast pace of development in semiconductor technology, old products are continuously replaced with new more advanced products . Therefore, new models need to be developed continuously for the new products. For example, by the year 2006, there were more than 7,000 active fault detection and classification models at IBM [1] and over 30,000 models at Intel [81]. In essence, it is highly desirable to minimize or eliminate the required data preprocessing steps without sacrificing monitoring performance due to the overwhelming effort required to carry out these steps.

The second limitation of MPCA when applied to fault detection is its assumption of Gaussian distributed data. Specifically, the assumption of Gaussian distributed data is made while calculating the control limits for the SPE and  $T^2$  indices using Equation 4.7 and Equations 4.12-4.14, respectively.

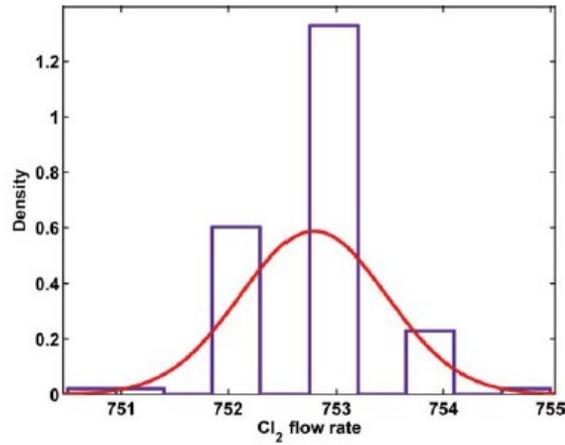
The assumption behind the control limit for SPE is that the variables in the unfolded matrix have a multivariate Gaussian distribution with population mean equal to zero [57, 86]. The control limit of  $T^2$  is determined with the assumption that the scores follow a multivariate Gaussian distribution with population mean equal to zero and estimated covariance matrix  $\Lambda$  [86, 121]. It should be noted that no assumption about the data distribution is made when PCA and MPCA are applied to reduce data dimensionality or to find patterns/clusters in the data.

For most semiconductor processes, the assumption of multivariate Gaussian distribution for the unfolded variables or the scores is violated because of the misaligned batch trajectories and the process mean shifts following tool maintenance events. Figure 4.8 shows the distributions of two process variables,  $BCl_3$  flow rate and  $Cl_2$  flow rate, present in the benchmark dataset. The fitted probability density functions (pdfs) of Gaussian distributions are also plotted. It is evident that the distributions of these process variables are far different from Gaussian distribution. The non-Gaussian distributed data is one of the major factors that affect the performance of MPCA when used for fault detection.

The third limitation of MPCA is that it is a second-order method, which only considers the mean and the variance-covariance of the data. Therefore, MPCA lacks the capability of providing higher-order representations of the data collected from semiconductor manufacturing processes, which is often non-Gaussian in nature [61, 71].



(a)



(b)

Figure 4.8: Histograms of two process variables from the benchmark dataset measured at a fixed time stamp for all the normal wafers (a)  $BCl_3$  flow rate; (b)  $Cl_2$  flow rate. Clearly the distributions of these process variables are not close to a Gaussian distribution.

#### 4.4.6 Fault detection using MPCA with EWMA filtering of residuals

We saw in the last section that the non-Gaussian nature of the data collected from semiconductor manufacturing processes poses serious challenges to the fault detection performance of MPCA. When the data are not normally distributed, undesirable false alarms are generated as shown in Figure 4.7. To reduce false alarms, Exponentially-Weighted-Moving-Average (EWMA) filtering can be applied to the residuals [104]. Here, the projections of the observed data on the residual subspace are referred to as residuals, which are calculated using Equation 4.5. Since an EWMA filter calculates a weighted average of a group of data samples in a moving window, the filtered residuals are closer to the normal distribution than the unfiltered residuals. This statement is supported by the popular Central Limit Theorem (CLT) [19] in statistics which states that:

“Whatever be the distributions of the independent variables  $\xi_v$  - subject to certain very general conditions - the sum  $\xi = \xi_1 + \xi_2 + \dots + \xi_n$  is asymptotically normal  $(m, \sigma)$ , where

$$m = m_1 + m_2 + \dots + m_n \quad (4.17)$$

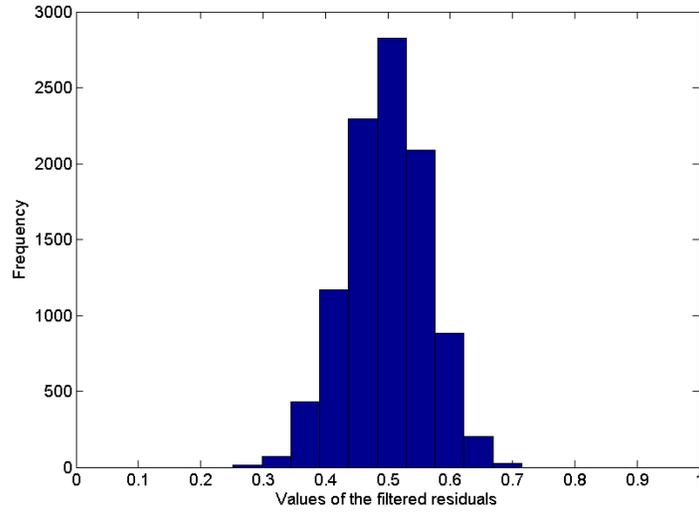
$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \dots + \sigma_n^2 \quad (4.18)$$

$m_v$  and  $\sigma_v$  are the mean and standard deviation of  $\xi_v$ , respectively”. Some extensions of the classical CLT have been established in the literature [20, 97].

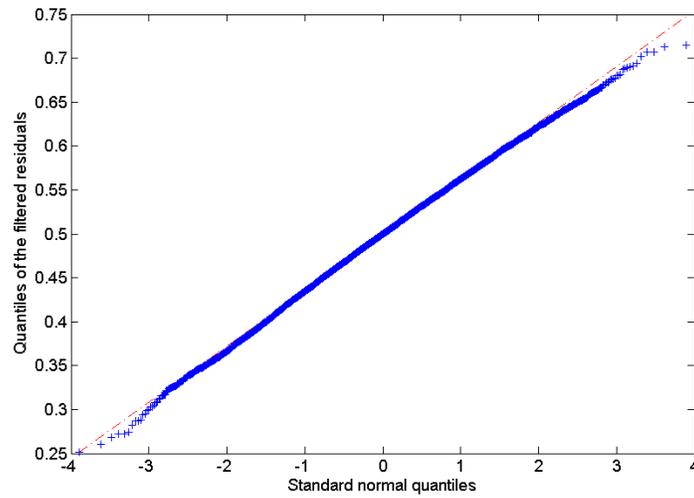
$$\bar{e}_k = (I - \Gamma)\bar{e}_{k-1} + \Gamma\tilde{x}_k \quad (4.19)$$

The general EWMA expression for residuals is given by Equation 4.19, where  $\bar{e}_k$  and  $\tilde{x}_k$  are the filtered and the unfiltered residual values for sample  $k$ , respectively.  $\Gamma$  denotes a diagonal matrix whose diagonal elements are the forgetting factors for the residuals. The value of  $\Gamma$  can be chosen to detect a particular kind of fault. Typically,  $\Gamma$  close to identity favors the detection of variance changes in the data, while  $\Gamma$  close to zero is more sensitive to the changes in mean values of the data. Therefore, the diagonal elements of  $\Gamma$  can be adjusted according to the type of fault to be detected in each process variable/sensor. Dunia et al. [25] have presented a few examples for choosing the forgetting factors for EWMA filters.

In order to demonstrate the capability of EWMA filtering to improve the normality of data obtained from a non-Gaussian distribution, we present the following simple example. Ten thousand random numbers were generated from the interval (0,1) using a uniform distribution to represent the residuals of Equation 4.19. The filtered values of the residuals were calculated by setting  $\Gamma$  equal to 0.1. Figure 4.9 shows the obtained filtered values of the residuals. It can be seen in Figure 4.9(a) that the histogram of the filtered residuals roughly resembles that of a normal distribution. The corresponding q-q plot is provided in Figure 4.9(b), which shows that the quantiles of the filtered residuals almost overlap the quantiles of the standard normal distribution. A smaller value of  $\Gamma$  leads to a smaller weighting of the most recent unfiltered



(a)



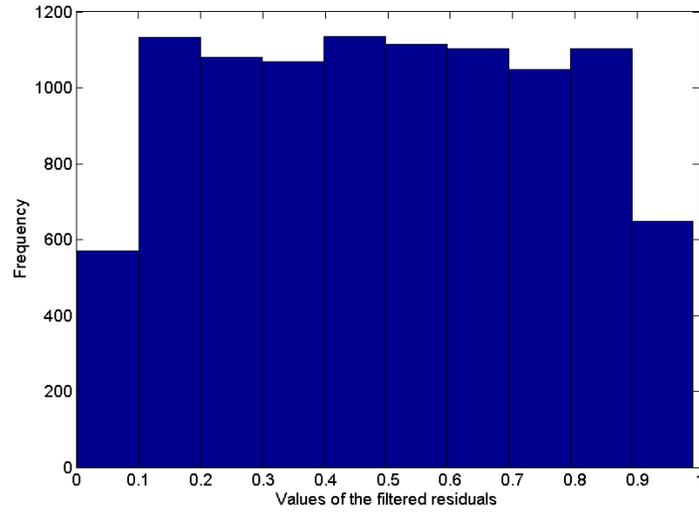
(b)

Figure 4.9: Filtered residuals obtained by setting the EWMA forgetting factor ( $\Gamma$ ) equal to 0.1 (a) Histogram (b) q-q plot. The filtered residuals conform better to the normal distribution than the unfiltered residuals (generated from a uniform distribution).

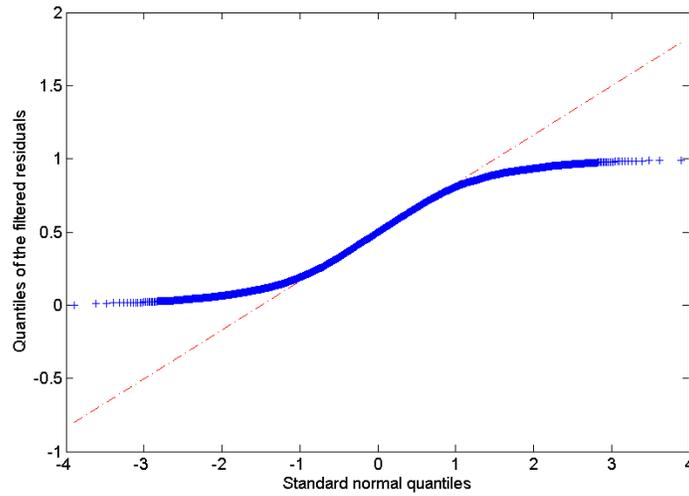
residual value. This means that the obtained filtered values will be averaged over more unfiltered values as compared to the filtered values obtained using a larger value of  $\Gamma$ . The increased averaging effect is the reason behind the improved normality of the residuals. Hence, we can conclude that EWMA filtering leads to an improvement in the normality of the residuals.

Qin et al. [104] and Dunia et al. [25] have provided examples to demonstrate the use of EWMA filtering of residuals. However, the value of  $\Gamma$  was chosen to be 0.1 in the examples studied in both of these references. No simulations or results were reported for the values of  $\Gamma$  close to unity. The simple example simulated in our study reveals that EWMA filtering is unable to improve the normality of the residuals if the chosen value of  $\Gamma$  is close to unity. A larger value of  $\Gamma$  leads to a larger weighting of the most recent unfiltered residual value, which reduces the averaging effect and causes the filtered residuals to behave more like the unfiltered residuals. If the value of  $\Gamma$  is set to unity, there will be no filtering and the obtained filtered residuals will have the same distribution as that of the unfiltered residuals (uniform distribution in this case). Figures 4.10(a) and 4.10(b) show the histogram and the q-q plot of the filtered residuals using a  $\Gamma$  value of 0.9, respectively. It is evident that the distribution of the filtered residuals is quite different from a normal distribution; in fact, it resembles a uniform distribution more. In essence, the ability of EWMA filtering to improve the normality of the residuals strongly depends on the value of the EWMA forgetting factor,  $\Gamma$ .

We mentioned earlier in this section that  $\Gamma$  close to unity favors the



(a)



(b)

Figure 4.10: Filtered residuals obtained by setting the EWMA forgetting factor ( $\Gamma$ ) equal to 0.9 (a) Histogram (b) q-q plot. EWMA filtering fails to improve the normality of the residuals. The filtered residuals resemble the distribution of the unfiltered residuals (uniform distribution) more than the normal distribution.

detection of variance changes in the data, while  $\Gamma$  close to zero is more sensitive to the changes in mean values of the data. Because EWMA filtering has the potential to improve the normality of the residuals only for small values of  $\Gamma$ , we can say that EWMA filtering improves the fault detection performance of MPCA only when the faults are present in the mean values of the data.

$$S\bar{P}E = \|\bar{e}\|^2 \quad (4.20)$$

$$\bar{\delta}_\alpha^2 = \frac{\gamma}{2 - \gamma} \delta_\alpha^2 \quad (4.21)$$

In order to detect faults using the filtered values of residuals, a new fault detection index  $S\bar{P}E$  is calculated using Equation 4.20. This equation is analogous to Equation 4.6 with the unfiltered residual  $\tilde{x}$  being replaced by the filtered residual  $\bar{e}$ . The corresponding control limit for  $S\bar{P}E$  is given by Equation 4.21. In this equation,  $\bar{\delta}_\alpha^2$  is the control limit for  $S\bar{P}E$ ,  $\gamma$  is the EWMA forgetting factor assuming  $\Gamma = \gamma I$ , and  $\delta_\alpha^2$  is the control limit for  $SPE$  calculated using Equation 4.7. It should be noted that  $\bar{\delta}_\alpha^2$  is smaller than  $\delta_\alpha^2$  for  $\gamma$  values between zero and one. The two limits are equal when  $\gamma$  is equal to one (no filtering). In other words,  $S\bar{P}E$  defines a tighter control region than  $SPE$  because of filtering. An interested reader can find the derivation of this limit in Qin et al. [104].

Figure 4.11 illustrates the benefit of performing EWMA filtering of the residuals obtained while employing MPCA for fault detection. Without any filtering, twelve false alarms were raised as mentioned in Section 4.4.4. EWMA

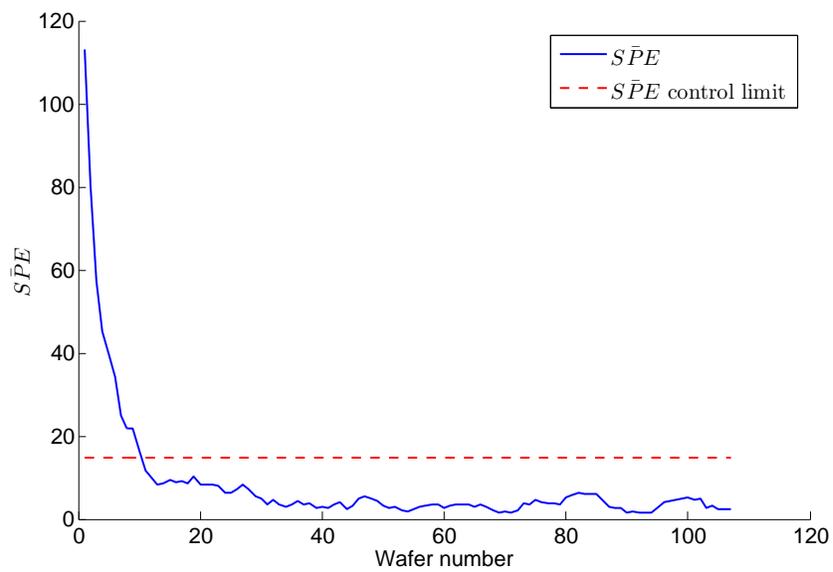


Figure 4.11:  $S\bar{P}E$  indices for 107 normal wafers using MPCA with EWMA filtering of residuals using a forgetting factor ( $\gamma$ ) value of 0.1. The number of false alarms is reduced to ten; twelve false alarms were observed when no filtering was used. It is evident that all the false alarms occur at the start of filtering and no false alarms are generated once  $S\bar{P}E$  falls below the  $S\bar{P}E$  control limit.

filtering smooths out the residuals, causing much smaller values of  $S\bar{P}E$ . In this study, ten false alarms were observed after EWMA filtering of residuals.

According to Equation 4.19, the filtered residuals of the previous wafer are required to calculate the filtered residuals for the current wafer. For the first wafer, no previous filtered residuals are available. Therefore, the filtered residual values are assumed to be same as the unfiltered values for the first wafer;  $S\bar{P}E$  value for the first wafer will be equal to a much larger SPE value. EWMA filtering starts to smooth out the unfiltered residuals second wafer onwards and causes a gradual reduction in the values of  $S\bar{P}E$ . As a result, false alarms occur at the start of filtering (for first ten wafers in this simulation). It is evident from Figure 4.11 that EWMA filtering successfully avoids generating false alarms once  $S\bar{P}E$  value falls below the  $S\bar{P}E$  control limit. Despite the demonstrated benefit of reducing false alarms, EWMA filtering of the residuals has a few limitations, which will be discussed in Section 5.2.3.

## 4.5 Conclusions

For a successful implementation of virtual metrology (VM), we need to make sure that the data entering the VM model are free from faults. Sensor faults are the most relevant faults in the context of VM as VM relies on the sensor data to predict the process outputs. When a sensor fault occurs, the corresponding sensor data are erroneous and do not represent the true behavior of the process. The quality of sensor data, which serve as inputs for VM models, has a direct effect on the quality of predicted values. In the

presence of faulty input data, an accurate VM model will provide erroneous predictions of the outputs. This situation is known as Garbage-In-Garbage-Out in the process modeling terms. The first step for the removal of effect of sensor faults from the sensor data is fault detection, which was done using MPCA in this chapter.

In order to compare the performance of various fault detection methods, a benchmark dataset (see Section 4.3) was utilized. The dataset consisted of processing data of 108 normal wafers and 21 faulty wafers. However, the faults in this dataset were introduced with a limited intention of comparing several methods in terms of their fault detection performance only. These intentionally induced faults serve the purpose of comparing different fault detection methods successfully, but are not suitable for performing fault identification and reconstruction (see Chapter 5 for details).

First, we presented fault detection using principal component analysis (PCA). PCA is able to detect faults for a two-dimensional data matrix only, the two dimensions being time and process variables in most cases. However, the data collected from a semiconductor manufacturing process are three-dimensional, with an additional dimension for different wafers. Hence, PCA cannot be directly applied for fault detection on data collected from a semiconductor manufacturing process. Instead, multiway principal component analysis (MPCA) is employed to address this limitation of PCA (see Section 4.2.2 for details).

MPCA was implemented to detect the artificial faults induced in the

benchmark dataset. The effect of the fault magnitude and the confidence level ( $\alpha$ ) on the fault detection performance of MPCA was also studied. It was found that MPCA raised several false alarms (i.e., MPCA indicated the presence of a fault for the fault-free data). Next, we presented a variation of MPCA that reduces false alarms by EWMA filtering of the residuals. MPCA with EWMA filtering of residuals detected less faults than MPCA for all the fault types. Due to the filtering of the residuals, the effect of the faults appears slowly in the filtered residuals. MPCA with EWMA filtering of residuals provided better detection when a small value of the EWMA forgetting factor ( $\Gamma$ ) was used, which favors the detection of mean faults. So, it detected more mean faults than the variance faults. The only advantage of using EWMA filtering is that it leads to fewer false alarms than MPCA. It was observed that both the MPCA-based methods were able to detect more mean and variance faults than the skewness and kurtosis faults. This is due to the fact that MPCA-based methods are second-order methods which only consider mean and variance of the data.

## Chapter 5

# Detection, Identification, and Reconstruction of Faults in Virtual Metrology Sensors

### 5.1 Introduction

The objective of this chapter is to remove the effect of sensor faults from the sensor data and feed the corrected (reconstructed) sensor data to the VM model. Figure 5.1 summarizes the approach that will be adopted in this chapter. First, the sensor data are analyzed to detect sensor faults. Once the sensor faults are detected, the next step is to figure out which sensor contains the fault. It is possible that multiple sensors are simultaneously faulty, but the likelihood of the occurrence of simultaneous multiple sensor faults is fairly low as the sensors are independent physical entities. In this chapter, we will focus on single sensor faults only. After knowing which sensor is faulty, the magnitude of the fault is estimated. Fault-free sensor data can be constructed by removing the effect of the identified sensor faults from the faulty sensor data. Mathematically, fault-free sensor data are obtained by subtracting the estimated magnitudes of the faults from the faulty sensors present in the data. Finally, the reconstructed (fault-free) data are fed into the VM model to predict the process outputs.

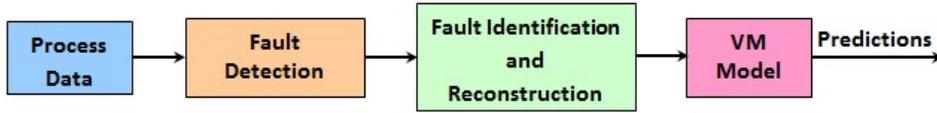


Figure 5.1: Summary of the approach adopted in this chapter.

In Chapter 4, we implemented MPCA for detection of sensor faults and discussed its limitations when employed for a semiconductor manufacturing process. In this chapter, we will present a statistics pattern analysis (SPA) based fault detection method that performs PCA on the statistics of the process variables unlike the traditional PCA and MPCA methods that perform PCA on the temporal values of the process variables. The advantages of SPA method over the PCA and MPCA methods will be discussed in the context of semiconductor manufacturing (see Section 5.2 for details).

Next, we will present and discuss three well-known fault identification methods present in literature. Specifically, these include contribution plot approach, reconstruction-based contribution (RBC) approach, and sensor validity index (SVI) approach (see Sections 5.3.1 - 5.3.3 for details). The magnitude of the fault will be estimated by minimizing the fault detection indices, SPE,  $T^2$ , or  $\phi$  (see Section 5.3.5 for details).

The fault detection, identification, and reconstruction performance of the above methods will be compared using a benchmark etch dataset. This comparison will enable us to determine the approaches that are the well-suited for correcting faults present in sensor data, which serve as inputs for VM models.

## 5.2 Fault Detection Using Statistics Pattern Analysis (SPA)

### 5.2.1 Motivation

As discussed in Section 4.2, data-driven fault detection approaches comprise of computational intelligence approaches and statistical process monitoring (SPM) methods. Pattern classification based monitoring (PCM) methods are an important class of computational intelligence approaches. Several PCM methods, which make use of the fault detection k-nearest-neighbor rule (FD-kNN) [13, 14, 43–46] and Mahalanobis distance [42, 109, 119] to reduce data preprocessing steps, have been developed recently. These PCM methods perform fault detection based on the simple idea that the trajectory of a normal sample is similar to those of the normal training samples, while the trajectory of a faulty sample exhibits some deviation from those of the normal training samples. The PCM methods utilize the complete training data to obtain a model for normal operating conditions. By doing so, the process nonlinearity and non-normality under normal operating conditions can be captured directly by the PCM methods. The number of the required training samples can be significantly reduced because batch unfolding is not needed; it may be recalled that batch unfolding in MPCA gives rise to a large number of variables, which require a huge amount of training data to build a PCA model. Another advantage of using PCM methods is that they seem to offer better fault detection performance as compared to the SPM methods in many cases [44, 69]. However, the PCM methods still require batch trajectory alignment

to make batch trajectories synchronized and of equal length, and usually require larger data storage space and longer computation times as compared to the SPM methods.

Recently, He and Wang [47] proposed a novel process monitoring framework called Statistics Pattern Analysis (SPA) to eliminate the data preprocessing steps mentioned above. The main difference between the traditional MPCA-based and the proposed SPA-based fault detection methods is that MPCA monitors the process variables, while SPA monitors various statistics of the process variables. In other words, singular value decomposition (SVD) is usually applied to the measurements of the process variables in MPCA. The obtained model captures the dominant correlations of process variables under normal operation, and the new measurements of the process variables are projected onto the MPCA model to perform fault detection. On the other hand, SVD is applied to the statistics calculated from process measurements under normal operation in SPA. The obtained model captures the dominant correlations of the statistics, and the statistics calculated from the new measurements are projected onto the model to perform fault detection. The statistics that capture different characteristics of the process can be selected to model the normal process operation, and process nonlinearity and non-normality can be quantified explicitly and used for process monitoring.

### 5.2.2 SPA framework

Batch statistics are monitored in the SPA-based fault detection method unlike traditional fault detection methods which monitor process variables. A statistics pattern (SP) is a collection of various statistics calculated from a batch trajectory, which capture the characteristics of each variable (e.g. mean and variance) as well as the interactions among different variables (e.g. covariance). The basic idea of the SPA framework is that the SPs of normal batches follow a similar pattern known as normal pattern, while the SPs of abnormal batches show some deviations from the normal pattern. The idea is supported by the fact that different batches processed on the same equipment are governed by the same physical/chemical mechanisms such as mass transport, kinetics, and thermodynamics.

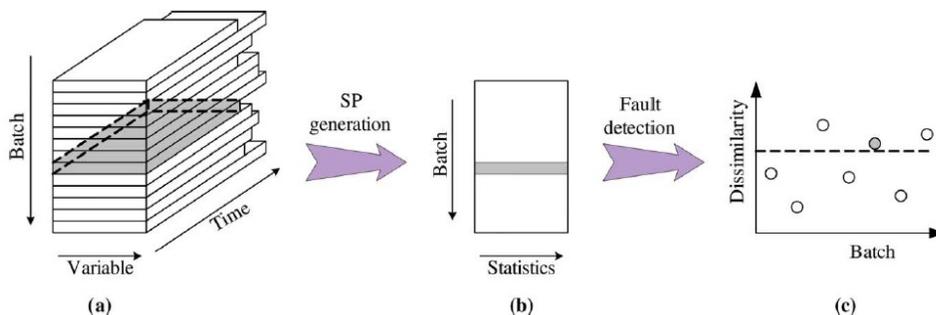


Figure 5.2: Illustration of the SPA framework. (a) Original batches of unequal length; (b) Statistics pattern (SP) generation; (c) Fault detection using dissimilarity quantification.

Figure 5.2 illustrates the two steps involved in performing fault detection using SPA. The first step is to generate a SP, which extracts the characteristics of a batch trajectory by calculating various statistics. The second

step is dissimilarity quantification and fault detection, where we quantify the dissimilarities among the SPs of normal batches and determine the control limit of the dissimilarity associated with a normal batch. The two steps are described in more detail as follows.

$$X = \begin{bmatrix} x_1(1) & x_1(2) & \cdots & x_1(d) \\ x_2(1) & x_2(2) & \cdots & x_2(d) \\ \vdots & \vdots & \ddots & \vdots \\ x_v(1) & x_v(2) & \cdots & x_v(d) \end{bmatrix} \quad (5.1)$$

Equation 5.1 shows a matrix  $X$ , which contains data collected from the processing of a batch. The size of matrix  $X$  is  $v \times d$ , where  $v$  is the number of the recorded process variables and  $d$  is the number of time stamps.  $X$  corresponds to the highlighted layer shown in Figure 5.2a.

In this work, SP is made up of four batch statistics: mean, variance/covariance, skewness, and kurtosis. The mathematical expressions of these statistics are provided in Equations 5.2 - 5.6. In Equation 5.2,  $SP_n$  stands for the SP of the  $n^{th}$  batch.  $\mu$ ,  $\Sigma$ ,  $\gamma$ , and  $\kappa$  contain the means, variances/covariances, skewnesses, and kurtoses, respectively, of  $v$  process variables for the  $n^{th}$  batch.

$$SP_n = [\mu \ \Sigma \ \gamma \ \kappa] \quad (5.2)$$

$$\mu = [E(x_i)] = \left[ \frac{1}{d} \sum_{k=1}^d x_i(k) \right] \quad (5.3)$$

$$\Sigma = [cov(x_i, x_j)] = \left[ \frac{1}{d-1} \sum_{k=1}^d (x_i(k) - \mu_i)(x_j(k) - \mu_j) \right] \quad (5.4)$$

$$\gamma = [\gamma_i] = \left[ \frac{E[(x_i - \mu_i)^3]}{E[(x_i - \mu_i)^2]^{3/2}} \right] \quad (5.5)$$

$$\kappa = [\kappa_i] = \left[ \frac{E[(x_i - \mu_i)^4]}{E[(x_i - \mu_i)^2]^2} - 3 \right] \quad (5.6)$$

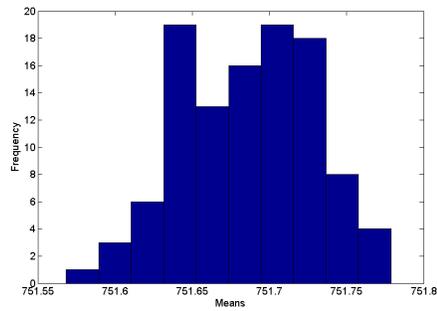
For  $v$  process variables,  $\mu$ ,  $\Sigma$ ,  $\gamma$ , and  $\kappa$  will have  $v$ ,  $v^2$ ,  $v$ , and  $v$  elements, respectively for each batch. Assuming we have data for  $m$  batches/wafers,  $SP_n$  shown in Equation 5.2 will form  $n^{th}$  row of the SP matrix representing all the batches, which will be of size  $m \times v(v+3)$ . The selection of batch statistics can be modified to capture specific process characteristics. For example, higher-order statistics (HOS) can be included to capture the process dynamics, nonlinearity, and non-normality. Although different batches may vary in batch lengths, same process variables are recorded for all of them. Therefore,  $SP_n$ s obtained from different batches will always have the same dimensions of  $1 \times v(v+3)$  and no data preprocessing will be needed for generating SP. The number of columns in the SP matrix can be reduced by utilizing the symmetric nature of the covariance matrix  $\Sigma$ . By including the upper triangular part of  $\Sigma$  only, the number of columns can be reduced to  $v \left( \frac{v+7}{2} \right)$ . As the batch duration  $d$  is usually much larger than the number of process variables  $v$ , the size of the SP matrix ( $m \times v \left( \frac{v+7}{2} \right)$ ) is much smaller than the size of the unfolded data matrix used in the MPCA method ( $m \times vd$ ).

After generating the SP matrix based on normal batches/wafers, we can build a PCA model as described in Section 4.2.1. In Section 4.2.1, we

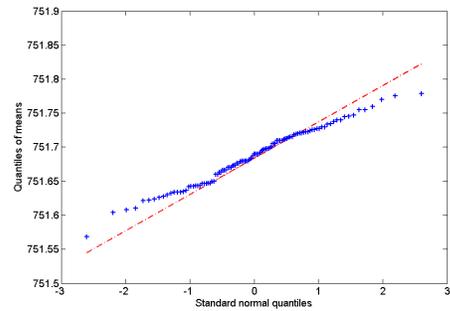
suggested to build a PCA model using the data matrix of process variables, but for SPA we will build a PCA on the SP matrix. Important correlations between different batch statistics will be extracted, which can be utilized for fault detection. Three fault detection indices,  $SPE$ ,  $T^2$ , and  $\phi$  along with their control limits can be calculated using SP matrix as described in Section 4.2.1. To employ SPA for fault detection,  $SP_n$  of the new batch is calculated first. Then, the fault detection indices are calculated and compared with their respective control limits to detect faults. If the fault detection indices of the new batch are below their respective control limits, the new batch is classified as a normal batch; otherwise, it is classified as a faulty batch.

### 5.2.3 Fault detection results using SPA

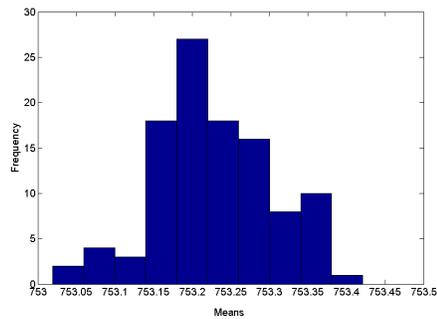
In the last subsection, we presented the details of the SPA framework. In this section, we will be performing fault detection using SPA on the benchmark dataset (see Section 4.3 for details). The benchmark dataset consists of data collected for 19 process variables for 107 normal wafers. SP of each batch is made up of 245 (19+190+18+18) elements which include 19 mean values of 19 process variables, 190 variances/covariances of 19 process variables (only upper triangular part of the covariance matrix required), 18 skewnesses for 18 process variables, and 18 kurtoses for 18 process variables (one process variable, RF bottom reflected power, has zero variance which makes its skewness and kurtosis indefinite). The SP matrix representing all the batches is of size  $107 \times 245$ . In Section 4.4.6, we presented Central Limit Theorem



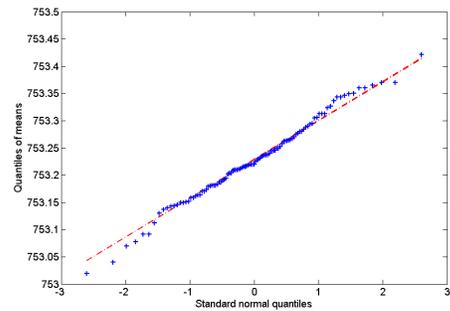
(a)



(b)

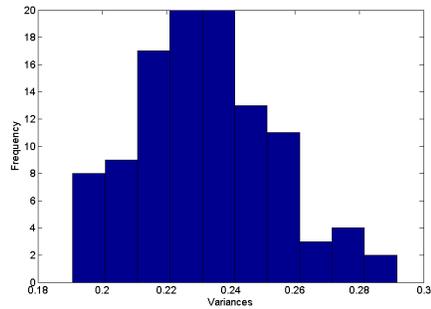


(c)

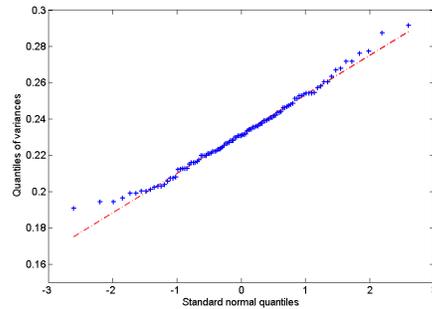


(d)

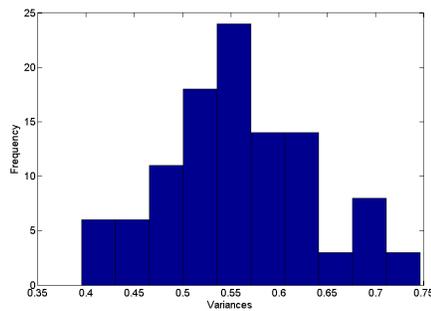
Figure 5.3: Histograms and q-q plots of the means for two process variables from the benchmark dataset for all the normal wafers (a and b)  $BCl_3$  flow rate; (c and d)  $Cl_2$  flow rate. Clearly, the distributions of the means are quite close to a Gaussian distribution.



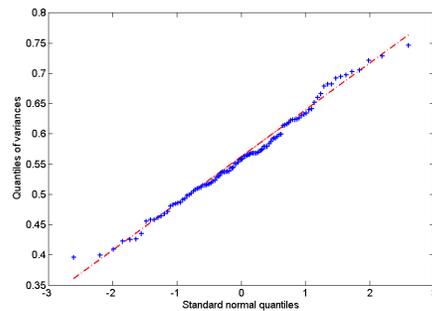
(a)



(b)

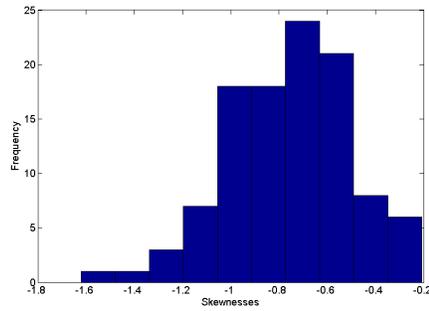


(c)

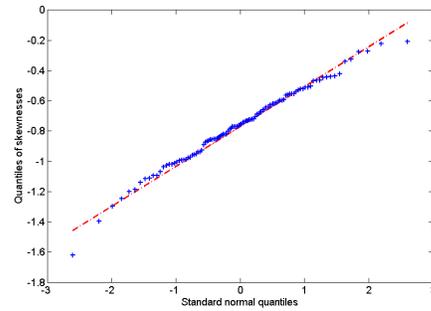


(d)

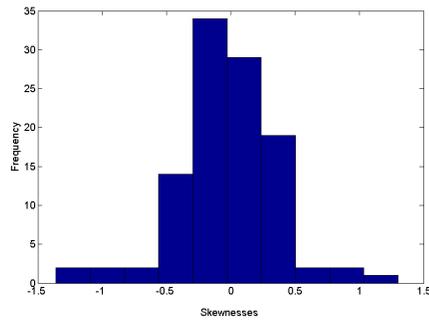
Figure 5.4: Histograms and q-q plots of the variances for two process variables from the benchmark dataset for all the normal wafers (a and b)  $BCl_3$  flow rate; (c and d)  $Cl_2$  flow rate. Clearly, the distributions of the variances are quite close to a Gaussian distribution.



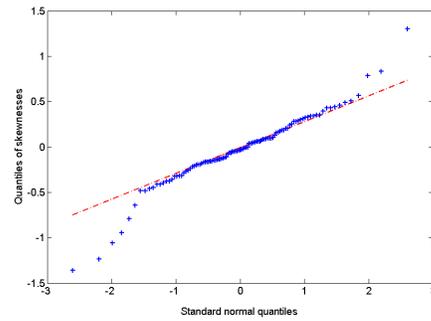
(a)



(b)

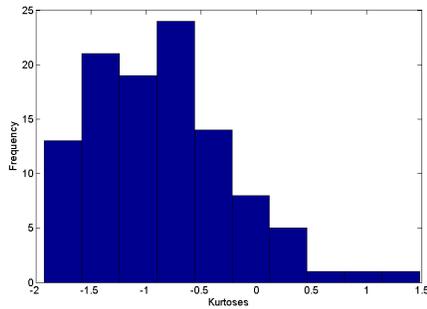


(c)

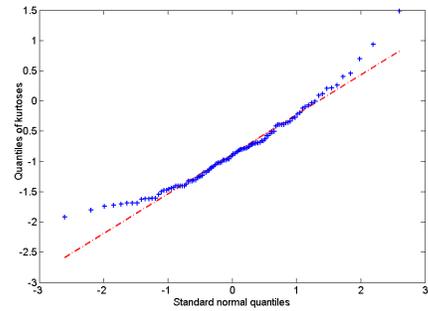


(d)

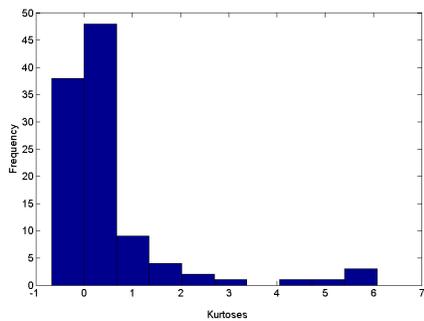
Figure 5.5: Histograms and q-q plots of the skewnesses for two process variables from the benchmark dataset for all the normal wafers (a and b)  $BCl_3$  flow rate; (c and d)  $Cl_2$  flow rate. Clearly, the distribution of the skewnesses for  $BCl_3$  flow rate is quite close to a Gaussian distribution.



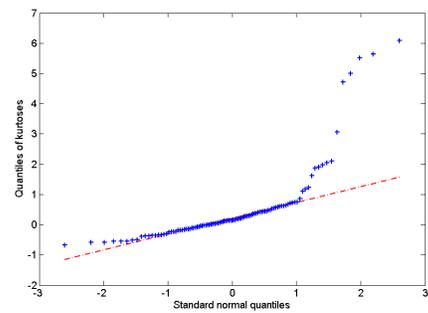
(a)



(b)



(c)



(d)

Figure 5.6: Histograms and q-q plots of the kurtoses for two process variables from the benchmark dataset for all the normal wafers (a and b)  $BCl_3$  flow rate; (c and d)  $Cl_2$  flow rate. The distributions of the kurtoses are not very close to a Gaussian distribution as the Central Limit Theorem holds weakly for higher-order statistics.

(CLT) which states that the means of random variables generated from any distribution asymptotically follow a Gaussian distribution. This result can be extended for other batch statistics such as variance, skewness, and kurtosis, but leads to reduced improvement in Gaussianity for higher-order statistics. This behavior is demonstrated by Figures 5.3-5.6, which show the histograms and the q-q plots for the means, variances, skewnesses, and kurtoses of two process variables,  $BCl_3$  flow rate and  $Cl_2$  flow rate for all 107 normal wafers. It is clear that the distributions of the batch statistics are more Gaussian than the distributions of the process variables (shown in Figure 4.8). The mean values for the two process variables show the maximum Gaussianity among the four batch statistics, with kurtosis (a fourth-order statistic) showing the least Gaussianity. Similar behavior of the four batch statistics was observed for other process variables.

Next, the ability of the SPA fault detection method to avoid false alarms was studied. From 107 normal wafers, only one false alarm was observed, which is much lower than the number of false alarms observed using MPCA and MPCA with EWMA filtering of residuals (12 and 10, respectively). The reduction in the number of false alarms is due to the following two features of SPA. First, better Gaussianity of the statistics in the SP matrix allows accurate estimation of the control limits. Second, incorporation of higher-order statistics (skewness and kurtosis) in SPA allows better representation of the process by the PCA model. While using MPCA for fault detection in Section 4.4.4, the non-Gaussian characteristics of the data were misinterpreted

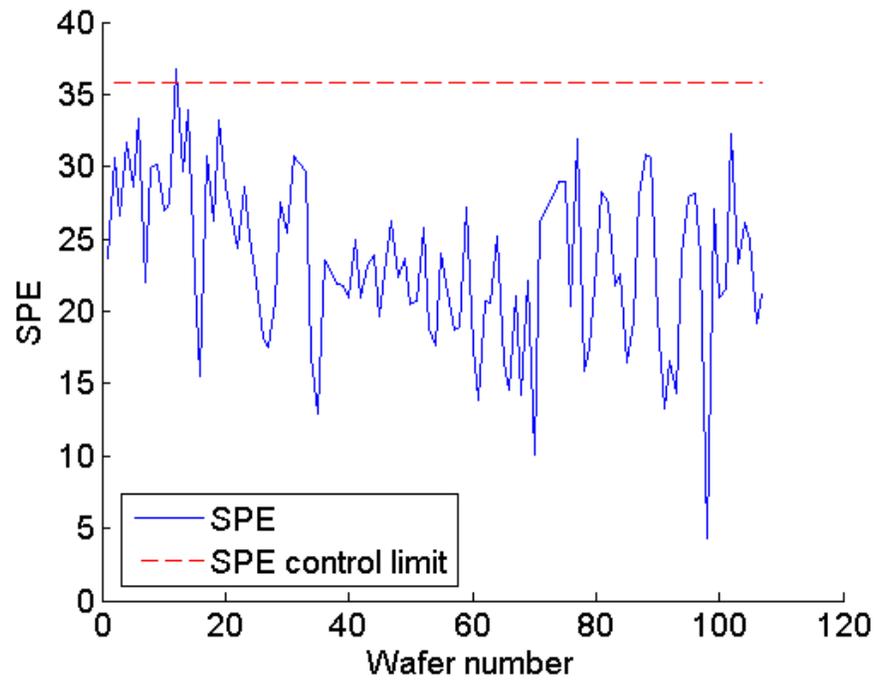


Figure 5.7: Fault detection using SPA raises only one false alarm while monitoring data for 107 normal wafers.

as faults leading to a large number of false alarms. SPA incorporates the non-Gaussian characteristics of the normal (fault-free) data in the PCA model leading to very few false alarms. Figure 5.7 shows that only one false alarm occurs when fault detection is done using SPA. Table 5.1 provides the number of false alarms raised while performing fault detection using MPCA, MPCA with EWMA filtering of residuals, and SPA.

Table 5.1: Comparison of three fault detection methods studied in this work: MPCA, MPCA with EWMA filtering of residuals, and SPA in terms of number of false alarms raised.

Fault detection method	Number of false alarms raised
MPCA	12
MPCA with EWMA filtering of residuals	10
SPA	1

The above study reveals that fault detection using MPCA-based methods raises more false alarms than fault detection using SPA. SPA is a superior fault detection method than MPCA with EWMA filtering of residuals due to the five limitations of the latter approach listed below.

1) EWMA filtering leads to an improved form of only one fault detection index, the SPE index (the improved form is  $S\bar{P}E$  index). No such improved indices exist for the  $T^2$  and  $\phi$  indices. 2) The improvement in the normality of data on using EWMA filtering is sensitive to the value of the EWMA forgetting factor ( $\Gamma$ ). We saw in Section 4.4.6 that EWMA filtering improves the normality of the data when small values of  $\Gamma$  were chosen and no improvement in the normality was observed for large values of  $\Gamma$ . 3) A value of  $\Gamma$  closer to zero favors the detection of faults in the mean values of the process variables,

while a value closer to one favors the detection of faults in the variance of the process variables. As the normality of the data deteriorates for  $\Gamma$  values closer to one, MPCA with EWMA filtering of residuals can provide good detection results only when the faults occur in the mean values of the process variables. The method is unable to detect faults in other higher-order statistics such as variance, skewness, and kurtosis. 4) EWMA filtering of the residuals leads to late detection of faults. When a fault occurs, its effect starts to appear slowly in the filtered residuals and causes a delay in the detection. 5) All MPCA-based fault detection methods require unfolding of the three-dimensional data before building a PCA model. This unfolding gives rise to a large number of process variables in the unfolded data matrix, which needs more storage space and longer computation times. Therefore, SPA is better suited for detecting faults in the three-dimensional data collected from semiconductor manufacturing processes than MPCA-based methods.

Next, the fault detection ability of SPA is tested by applying it to detect the artificial faults that were induced in Section 4.4.1. We may recall that 19 sensor faults were introduced by adding a constant bias to the mean value of one of the process variables at a time. In other words, the first sensor fault was created by adding a constant bias to the mean value of the first process variable, the second sensor fault was created by adding a constant bias to the mean value of the second process variable, and so on (see Table 4.1 for the list of 19 process variables). By doing so, faults in the means of the process variables were simulated. The sensor faults were introduced in this manner to

investigate the detectability of the faults in each of these 19 process variables using SPA. The magnitudes of the introduced faults were chosen to be 20 % of the mean values of the process variables. For example, to simulate the first fault, a constant bias of magnitude equal to 20 % of the mean value of the first process variable was added to the normal/setpoint value of the first process variable.

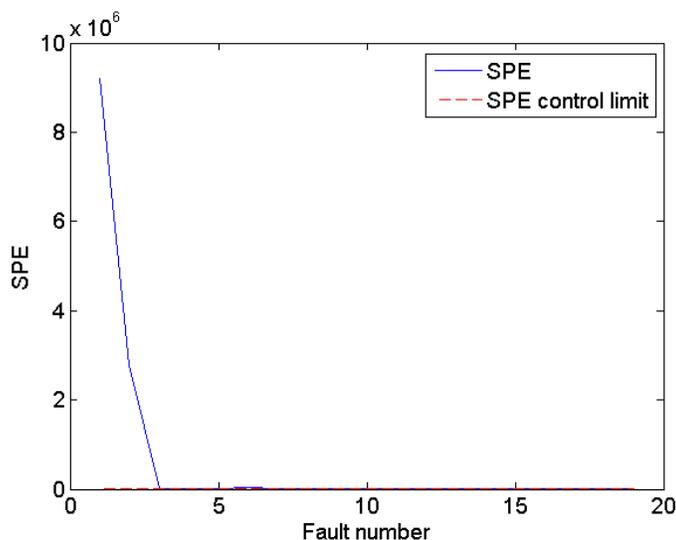


Figure 5.8: Fault detection using SPA. All 19 faults present in the mean values of the process variables are detected.

Figures 5.8-5.10 show the fault detection performance of SPA using the SPE index. It can be seen that all the faults are detected successfully. He and Wang [47] were also able to detect all the faults present in the original dataset (different from the faults induced in this work) using SPA. The faults considered in their study are described in Section 4.3. SPA provides better fault detection results than MPCA-based methods because: 1) SPA utilizes

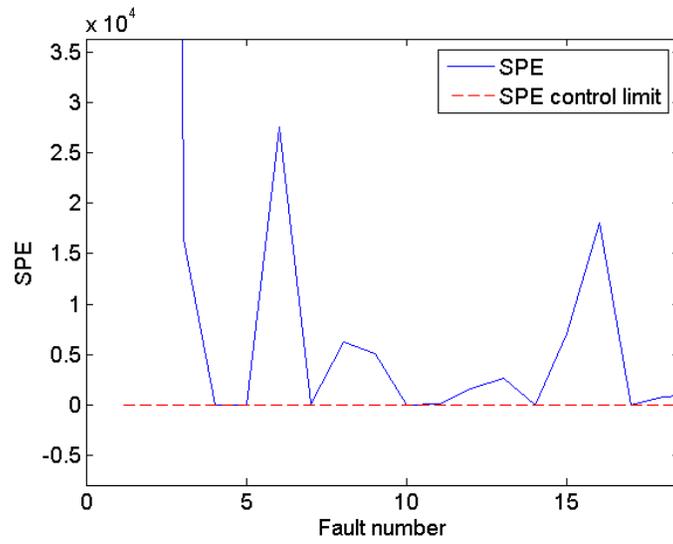


Figure 5.9: Fault detection using SPA. Zoomed-in view of Figure 5.8.

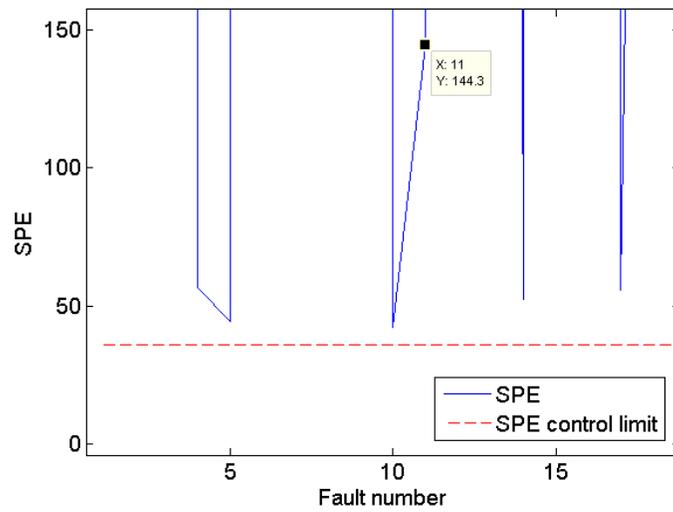


Figure 5.10: Fault detection using SPA. Zoomed-in view of Figure 5.9.

batch statistics, which exhibit better Gaussian characteristics than the process variables, to form the PCA model; 2) SPA incorporates higher-order statistics (skewness and kurtosis) in the model to understand the non-Gaussian characteristics of the data.

Table 5.2: Comparison of three fault detection methods studied in this work: MPCA, MPCA with EWMA filtering of residuals, and SPA in terms of number of different types of faults detected. A total of 74 faults comprising of 19 mean faults (one for each process variable), 19 variance faults, 18 skewness faults, and 18 kurtosis faults were introduced.

Fault detection method	MPCA	MPCA with EWMA filtering of residuals	SPA
Mean faults detected	14	12	19
Variance faults detected	12	7	17
Skewness faults detected	3	2	13
Kurtosis faults detected	2	2	10
Total faults detected	31	23	59

Similar fault detection studies were done by introducing faults in other batch statistics (variance, skewness, and kurtosis) and the results using MPCA, MPCA with EWMA filtering of residuals, and SPA are summarized in Table 5.2. For each batch statistic, 19 faults of size equal to 20% of the corresponding batch statistic were introduced in the data for normal wafers. It can be observed that both the MPCA-based methods are able to detect more mean and variance faults than the skewness and kurtosis faults. This is due to the fact that MPCA-based methods are second-order methods which only consider mean and variance of the data. The numbers of detected mean and variance faults by using both MPCA-based methods are less than those detected by SPA. This is due to the non-Gaussian characteristics of the data collected

from semiconductor manufacturing processes. MPCA-based methods wrongly assume the data to be Gaussian and calculate erroneous control limits for the fault detection indices. It can be observed that MPCA with EWMA filtering of residuals detects less faults than MPCA for all the fault types. Due to filtering of the residuals, the effect of the faults appears slowly in the filtered residuals. MPCA with EWMA filtering of residuals provides better detection when a small value of the EWMA forgetting factor ( $\Gamma$ ) is used, which favors the detection of mean faults. So, it detects more mean faults than the variance faults. The only advantage of using EWMA filtering is that it leads to fewer false alarms than MPCA.

The above discussion concludes that SPA provides better fault detection performance for different types of faults as compared to MPCA-based methods. Not only it detects more faults, SPA also significantly reduces the number of false alarms. Therefore, this study recommends that SPA should be employed as a fault detection method to detect faults in the Virtual Metrology sensors. After detecting the faults, fault identification is performed to identify the sensors that caused the faults. The next section focuses on fault identification and compares three well-known identification methods: contribution plot approach, sensor validity index (SVI) approach, and reconstruction-based contribution (RBC) approach.

### 5.3 Fault Identification and Reconstruction

While much work has been reported in fault detection, only a few methods are available for fault identification/diagnosis. As an early and popular method, contribution plots are used to identify the cause of a fault by determining the contribution of each variable to the fault detection indices [80, 86, 129]. The assumption behind the contribution plot method is that the faulty variables have high contributions to the fault detection indices. Several approaches have been proposed for defining variable contributions [17, 103, 129, 133]. The variable contributions to the SPE index are defined exactly, but the variable contributions to the  $T^2$  index and the combined  $\phi$  index involve several approximations.

Other fault identification methods present in the relevant literature include: (a) sensor validity index (SVI) approach [24, 25, 145]; (b) reconstruction-based contribution (RBC) method [2]; (c) discrimination by angles [106, 143]; (d) pattern matching methods by calculating similarity and dissimilarity factors between normal data and an extended period of fault data [60, 114, 115]; and (e) isolation enhanced techniques from model-based methods [34, 35, 101, 102]. Among these methods, rigorous diagnosability analyses are available for the contribution plot method [2], sensor validity index method [24], and RBC method [2].

In this section, we will review and implement three fault identification methods: contribution plot method, sensor validity index method, and reconstruction-based contribution (RBC) method to identify the sensors that

cause the artificial faults presented in Section 4.4.1. Our goal is to compare the three methods and figure out which of these methods are best suited for identifying/diagnosing the faults present in virtual metrology sensors. Accurate fault identification is required to remove the effect of fault from the faulty sensor signal and consequently, predict process outputs with high precision by feeding the fault-free sensor signals to the virtual metrology model.

### 5.3.1 Contribution plot approach

A fault is detected after one or more fault detection indices exceed their control limits. Contribution plots are based on the idea that the variables with the largest contributions to the fault detection index are most likely the faulty variables. The contribution plots are constructed by determining the contribution of each variable to the fault detection index. In order to calculate these contributions, first we have to notice that the expressions of the fault detection indices,  $SPE$ ,  $T^2$ , and  $\phi$  given by Equations 4.6, 4.10, and 4.15, respectively, have the general quadratic form:

$$Index(x) = x^T M x = \|x\|_M^2 \quad (5.7)$$

where  $M = \tilde{C}$  for SPE index,  $M = D = P\Lambda^{-1}P^T$  for  $T^2$  index, and  $M = \Phi$  for the combined  $\phi$  index.  $Index(x)$  can be rewritten as:

$$Index(x) = x^T M x = \|M^{\frac{1}{2}}x\|^2 = \sum_{i=1}^n (\xi_i^T M^{\frac{1}{2}}x)^2 = \sum_{i=1}^n c_i^{Index} \quad (5.8)$$

where

$$c_i^{Index} = (\xi_i^T M^{\frac{1}{2}}x)^2 \quad (5.9)$$

is the contribution of the variable  $x_i$  to the  $Index(x)$ . Here,  $\xi_i$  is the  $i^{th}$  column of the identity matrix and the direction of  $x_i$ . For example, for a process with three sensors, the direction of sensor  $x_1$  is

$$\xi_1 = [1 \ 0 \ 0]^T \quad (5.10)$$

$$c_i^{SPE} = (\xi_i^T \tilde{C}^{\frac{1}{2}} x)^2 = (\xi_i^T \tilde{C} x)^2 \quad (5.11)$$

The variable contributions for the SPE index are obtained by substituting M by  $\tilde{C}$  in Equation 5.9 as  $\tilde{C}^{\frac{1}{2}} = \tilde{C}$ . Equation 5.11 shows the contribution of the variable  $x_i$  to the SPE index as defined by Miller et al. [80].

$$c_i^{T^2} = (\xi_i^T D^{\frac{1}{2}} x)^2 \quad (5.12)$$

The variable contributions for the  $T^2$  index are obtained by substituting M by D in Equation 5.9. Equation 5.12 defines the contribution of the variable  $x_i$  to the  $T^2$  index as proposed by Wise et al. [133].

$$c_i^{\phi} = (\xi_i^T \Phi^{\frac{1}{2}} x)^2 \quad (5.13)$$

The variable contributions for the  $\phi$  index are obtained by substituting M by  $\Phi$  in Equation 5.9. Equation 5.13 defines the contribution of the variable  $x_i$  to the  $\phi$  index. Although there are other definitions for the variable contributions of the  $T^2$  and  $\phi$  indices, such definitions involve several forms through approximations [17, 103].

### 5.3.2 Reconstruction-based contribution (RBC) method

The reconstruction of a fault detection index along a variable direction minimizes the effect of this variable on the detection index [24]. Alcala and Qin

[2] used the amount of reconstruction along a variable direction as an amount of contribution of the variable to the fault detection index that is reconstructed. This amount of reconstruction was designated as the reconstruction-based contribution (RBC) of this variable to the fault detection index.

In a system with  $n$  sensors, when a fault happens in sensor  $x_i$ , the faulty measurement  $x$  is a vector of length  $n$  and the direction of the fault is  $\xi_i$ . The reconstructed vector along direction  $\xi_i$  is

$$z_i = x - \xi_i f_i \quad (5.14)$$

Dunia and Qin [24] provide reconstructions along an arbitrary direction for SPE index and Yue and Qin [145] provide reconstructions for  $T^2$  and  $\phi$  indices. In a general form, the fault detection index of the reconstructed measurement is

$$Index(z_i) = z_i^T M z_i = \|z_i\|_M^2 = \|x - \xi_i f_i\|_M^2 \quad (5.15)$$

The task of reconstruction is to find a value of  $f_i$  such that  $Index(z_i)$  is minimized. This minimization is done by taking the first derivative of  $Index(z_i)$  with respect to  $f_i$  and equating it to 0. This first step yields,

$$\frac{d(Index(z_i))}{df_i} = -2(x - \xi_i f_i)^T M \xi_i \quad (5.16)$$

Setting Equation 5.16 equal to 0 and solving for  $f_i$  yields

$$f_i = (\xi_i^T M \xi_i)^{-1} \xi_i^T M x \quad (5.17)$$

The reconstruction-based contribution of variable  $x_i$  to a fault detection index,  $RBC_i^{Index}$ , is the amount of reconstruction along the direction  $\xi_i$  which can be expressed from Equation 5.15 as

$$RBC_i^{Index} = \|\xi_i f_i\|_M^2 \quad (5.18)$$

On substituting  $f_i$  in the last equation, we get

$$RBC_i^{Index} = \|\xi_i(\xi_i^T M \xi_i)^{-1} \xi_i^T M x\|_M^2 = x^T M \xi_i (\xi_i^T M \xi_i)^{-1} \xi_i^T M x \quad (5.19)$$

Equation 5.19 represents the reconstruction-based contribution,  $RBC_i^{Index}$ , of the variable  $x_i$  to the fault detection index of interest. The reconstructed index,  $Index(z_i)$ , is obtained by substituting the value of  $f_i$  in Equation 5.15 and is found to be

$$Index(z_i) = x^T M [I - \xi_i (\xi_i^T M \xi_i)^{-1} \xi_i^T M] x = x^T M x - x^T M \xi_i (\xi_i^T M \xi_i)^{-1} \xi_i^T M x \quad (5.20)$$

$$Index(z_i) = Index(x) - RBC_i^{Index} \quad (5.21)$$

$$Index(x) = Index(z_i) + RBC_i^{Index} \quad (5.22)$$

Qin et al. [104] refer to the ratio of  $Index(z_i)$  to  $Index(x)$  as sensor validity index (SVI) and used it for fault identification. It should be noted that  $\xi_i$  direction in the above derivation does not have to be a sensor direction as in Equation 5.10; it can be an arbitrary process fault direction. Furthermore,  $\xi_i$  does not have to be a vector; it can be a column-like matrix representing

a multidimensional fault or multiple sensor faults. Therefore, RBC is a more general approach than the conventional contribution plots.

The reconstruction-based contribution of variable  $x_i$  to  $SPE(x)$ ,  $RBC_i^{SPE}$ , is obtained by substituting M by  $\tilde{C}$  in Equation 5.19.

$$RBC_i^{SPE} = x^T \tilde{C} \xi_i (\xi_i^T \tilde{C} \xi_i)^{-1} \xi_i^T \tilde{C} x = \frac{(\xi_i^T \tilde{C} x)^2}{\tilde{c}_{ii}} \quad (5.23)$$

where  $\tilde{c}_{ii} = \xi_i^T \tilde{C} \xi_i$  is the  $i^{th}$  diagonal element of  $\tilde{C}$ . Since  $\tilde{x}_i = \xi_i^T \tilde{C} x$  from Equation 5.11,  $RBC_i^{SPE}$  can be expressed as

$$RBC_i^{SPE} = \frac{\tilde{x}_i^2}{\tilde{c}_{ii}} = \frac{c_i^{SPE}}{\tilde{c}_{ii}} \quad (5.24)$$

Thus,  $RBC_i^{SPE}$  and  $c_i^{SPE}$  differ only by a scaling coefficient  $\tilde{c}_{ii}$ . However, since  $\tilde{c}_{ii}$  varies with  $i$ ,  $RBC_i^{SPE}$  and  $c_i^{SPE}$  are fundamentally different. An advantage of the RBC approach in determining the contributions of the  $T^2$  and  $\phi$  indices is that no approximations are involved and it is consistent for all indices.

The reconstruction-based contribution of variable  $x_i$  to the  $T^2$  index,  $RBC_i^{T^2}$ , is obtained by substituting M by D in Equation 5.19.

$$RBC_i^{T^2} = x^T D \xi_i (d_{ii})^{-1} \xi_i^T D x = \frac{(\xi_i^T D x)^2}{d_{ii}} \quad (5.25)$$

where  $d_{ii}$  is the  $i^{th}$  diagonal element of D. Unlike  $RBC_i^{SPE}$  and  $c_i^{SPE}$ , there is no obvious connection between  $RBC_i^{T^2}$  and  $c_i^{T^2}$ .

The reconstruction-based contribution of variable  $x_i$  to the  $\phi$  index,  $RBC_i^\phi$ , is obtained by substituting M by  $\Phi$  in Equation 5.19.

$$RBC_i^\phi = x^T \Phi \xi_i (\phi_{ii})^{-1} \xi_i^T \Phi x = \frac{(\xi_i^T \Phi x)^2}{\phi_{ii}} \quad (5.26)$$

where  $\phi_{ii}$  is the  $i^{th}$  diagonal element of  $\Phi$ .

### 5.3.3 Sensor validity index (SVI) method

When a sensor fault occurs, the measurement vector can be represented as follows:

$$x = x^* + \xi_i f_i \quad (5.27)$$

where  $x^*$  denotes the normal (fault-free) part of the measurement vector,  $\xi_i$  is the direction vector of unit length for the faulty sensor, and  $f_i$  is the magnitude of the fault which can be positive, negative, or zero. All possible  $m$  sensor fault directions are represented by the set  $\{\xi_i, i=1, 2, \dots, m\}$ .

To identify the true fault among all possible faults, Dunia et al. [25] proposed an identification approach by reconstructing  $x^*$  from  $x$  for all possible fault directions,  $\xi_i$ . For each assumed fault, the fault magnitude is estimated by performing reconstruction using other sensors as shown in Equation 5.17. This equation calculates the fault magnitude by minimizing a general fault detection index,  $x^T M x$ , for the reconstructed values. The occurrence of a fault significantly increases the values of the fault detection indices. If the true sensor is assumed, the largest reduction in the fault detection indices is expected, as all the reconstruction methods try to minimize the fault detection indices. The ratio of the fault detection index after reconstruction to the original fault detection index is sensitive to the fault and is termed as the sensor validity index (SVI). SVI can be defined for all three fault detection

indices,  $SPE$ ,  $T^2$ , and  $\phi$ , studied in this chapter. In general,

$$SVI_i = \frac{Index_{recons}(x_i)}{Index(x)} \quad (5.28)$$

In Equation 5.28,  $Index_{recons}(x_i)$  is the value of the fault detection index obtained after reconstruction of an assumed fault in the  $i^{th}$  sensor. It should be noted that  $0 \leq SVI_i \leq 1$  since  $Index_{recons}(x_i)$  is the minimized value of  $Index(x)$ . A SVI value close to one indicates that the sensor variations follow the variations experienced by the remaining sensors, while a SVI value close to zero indicates that the sensor is faulty. A detailed discussion of SVI can be found in Dunia et al. [25]. The SVI approach can guarantee correct fault identification when the fault direction is known and is in the candidate set of directions, but it cannot guarantee correct identification results for faults with unknown directions.

It should be noted that  $Index(z_i)$  in Equation 5.22 and  $Index_{recons}(x_i)$  in Equation 5.28 represent the same quantity, the general fault detection index after reconstructing the fault in the  $i^{th}$  sensor. This allows us to rewrite Equation 5.28 as

$$SVI_i = \frac{Index(z_i)}{Index(x)} \quad (5.29)$$

Substituting the value of  $Index(z_i)$  from Equation 5.22 into Equation 5.29 yields

$$SVI_i = \frac{Index(x) - RBC_i^{Index}}{Index(x)} = 1 - \frac{RBC_i^{Index}}{Index(x)} \quad (5.30)$$

Equation 5.30 shows the relationship between the SVI and the RBC. It can be recalled that both these methods were based on the reconstruction of

faulty sensors. Arriving at this relationship is not a surprising result as these two methods implement the same idea in different ways.

#### 5.3.4 Fault identification results

In order to compare the identification performance of the three fault identification methods discussed above, we utilize the same artificial faults presented in Section 4.4.1. Section 5.2.3 concluded that SPA provides better fault detection results than MPCA and MPCA with EWMA filtering of residuals. The main benefit of performing identification on the faults detected by SPA is that not only the faulty sensor can be identified, but the statistic (e.g., mean, variance) in which the fault occurred is also identified. This information is crucial for predicting accurate outputs using virtual metrology models, which mostly use the statistics of the sensor signals as the inputs. Therefore, we will perform identification on the faults detected by SPA in this section.

Figures 5.11-5.14 show the fault identification results obtained using RBC approach for the faults present in four different statistics of the process variables. Particularly, Figure 5.11 shows the reconstruction-based contributions (RBCs) of 245 statistics present in the statistics pattern (SP) to the SPE index. This SP corresponds to a wafer in which a fault was induced in the mean of process variable RF power. The magnitude of a vertical bar in Figure 5.11 represents the contribution of the corresponding statistic to the observed SPE fault detection index of the wafer. SP statistic number 11, which represents the mean value of RF power, has the largest RBC value of 8.4 (the value

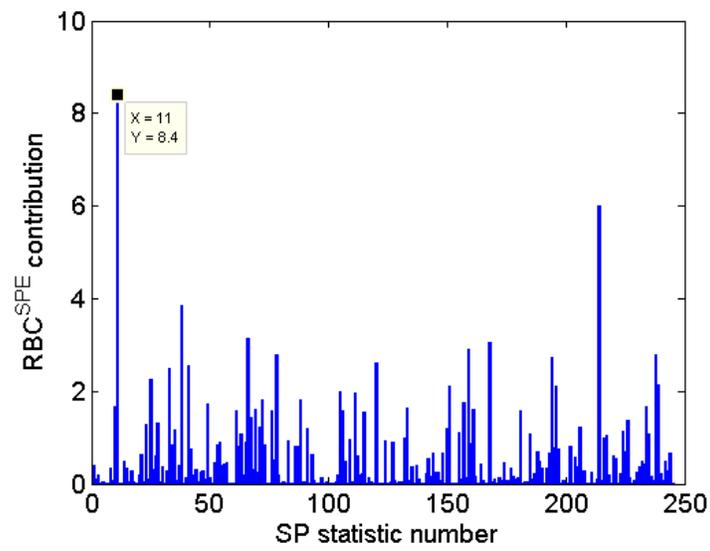


Figure 5.11: Fault identification for a fault in the mean of process variable RF power. SP statistic number 11, which corresponds to the mean of RF power, shows the largest contribution indicating correct identification.

of SPE for this fault was 144.3 as shown in Figure 5.10). Hence, the mean fault in RF power is correctly identified using RBC approach.

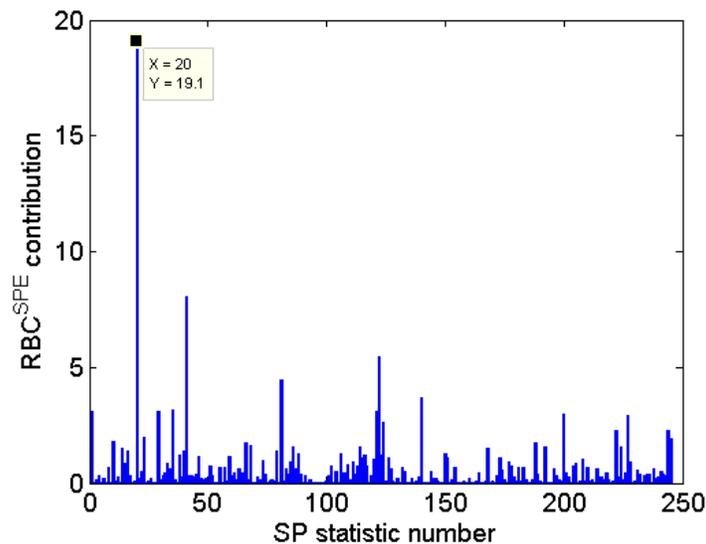


Figure 5.12: Fault identification for a fault in the variance of process variable  $BCl_3$  flow rate. SP statistic number 20, which corresponds to the variance of process variable  $BCl_3$  flow rate, shows the largest contribution indicating correct identification.

Figure 5.12 shows the reconstruction-based contributions (RBCs) of 245 statistics present in the statistics pattern (SP) to the SPE index. This SP corresponds to a wafer in which a fault was induced in the variance of process variable  $BCl_3$  flow rate. The magnitude of a vertical bar in Figure 5.12 represents the contribution of the corresponding statistic to the observed SPE fault detection index of the wafer. SP statistic number 20, which represents the variance of  $BCl_3$  flow rate, has the largest RBC value. Hence, the variance fault in  $BCl_3$  flow rate is correctly identified using RBC approach.

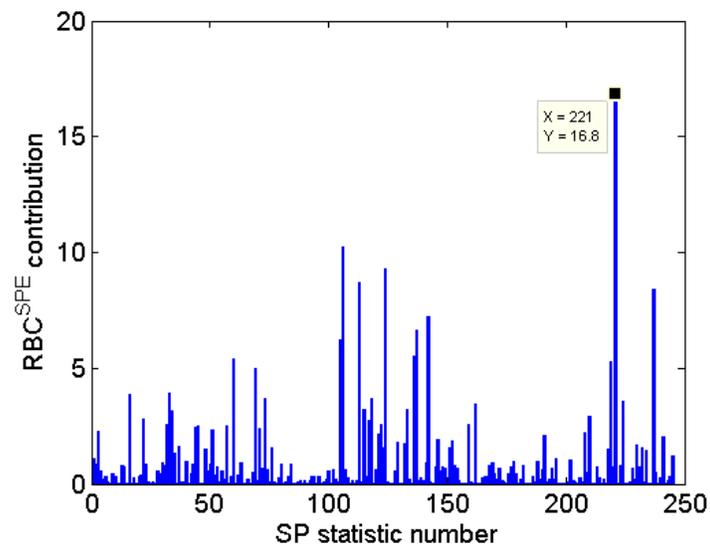


Figure 5.13: Fault identification for a fault in the skewness of process variable TCP tuner. SP statistic number 221, which corresponds to the skewness of process variable TCP tuner, shows the largest contribution indicating correct identification.

Figure 5.13 shows the reconstruction-based contributions (RBCs) of 245 statistics present in the statistics pattern (SP) to the SPE index. This SP corresponds to a wafer in which a fault was induced in the skewness of process variable TCP tuner. The magnitude of a vertical bar in Figure 5.13 represents the contribution of the corresponding statistic to the observed SPE fault detection index of the wafer. SP statistic number 221, which represents the skewness of TCP tuner, has the largest RBC value. Hence, the skewness fault in TCP tuner is correctly identified using RBC approach.

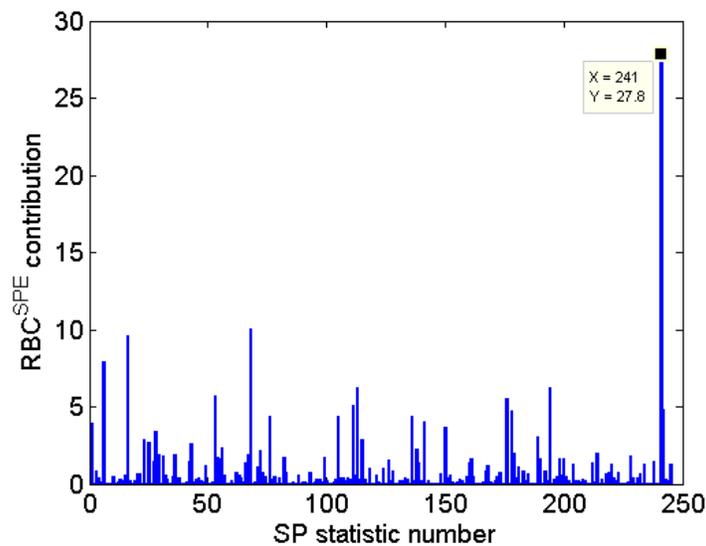


Figure 5.14: Fault identification for a fault in the kurtosis of process variable TCP impedance. SP statistic number 241, which corresponds to the kurtosis of process variable TCP impedance, shows the largest contribution indicating correct identification.

Figure 5.14 shows the reconstruction-based contributions (RBCs) of 245 statistics present in the statistics pattern (SP) to the SPE index. This SP

corresponds to a wafer in which a fault was induced in the kurtosis of process variable TCP impedance. The magnitude of a vertical bar in Figure 5.14 represents the contribution of the corresponding statistic to the observed SPE fault detection index of the wafer. SP statistic number 241, which represents the kurtosis of TCP impedance, has the largest RBC value. Hence, the kurtosis fault in TCP impedance is correctly identified using RBC approach.

In this study, a total of 74 faults comprising of 19 mean faults (one for each process variable), 19 variance faults, 18 skewness faults, and 18 kurtosis faults were introduced. The number of skewness and kurtosis faults is one less than that of mean and variance faults because one of the process variables, RF bottom reflected power, had zero variance and led to indefinite values of skewness and kurtosis.

The fault identification performance of the three methods mentioned in this chapter: contribution plot method, RBC method, and SVI method is compared in Table 5.3. The table shows the number of correctly identified faults using the three identification methods for different kinds of faults.

It can be observed from Table 5.3 that the RBC method and the SVI method exhibit similar identification results. This is due to the fact that both of these methods are based on the same idea of reconstruction of faults and are related by Equation 5.30. The contribution plot method identifies a smaller number of faults correctly as compared to the RBC and the SVI methods because of the limitations listed below: a) In contribution plots, the contributions are spread unevenly across variables when there is no fault. In

Table 5.3: Comparison of three fault identification methods studied in this work: contribution plots, RBC, and SVI in terms of number of correctly identified faults for different fault types. A total of 74 faults comprising of 19 mean faults (one for each process variable), 19 variance faults, 18 skewness faults, and 18 kurtosis faults were introduced.

Fault identification method	Contribution plots	RBC	SVI
Correctly identified mean faults	12	17	17
Correctly identified variance faults	8	13	13
Correctly identified skewness faults	3	7	7
Correctly identified kurtosis faults	4	6	6
Total correctly identified faults	27	43	43

other words, some variables have large contributions while others have relatively smaller contributions for the normal (fault-free) data. Therefore, a fault in a normally small-contribution variable may not make the contribution of that variable the largest unless the fault magnitude is very large. This is a common cause of misidentification while using contribution plots; b) Westerhuis [129] showed that a fault in one variable smears into the contributions of other variables in the contribution plot approach and degrades the fault identification performance. This fault smearing also exists in the RBC approach. Alcalá and Qin [2] have shown that the RBC method guarantees correct fault identification, while contribution plots cannot guarantee correct identification even if the fault is only in a single sensor; c) In the contribution plot approach, faulty data are used to calculate the contributions of all the sensors to the fault detection index. Therefore, a fault present in a sensor propagates to other sensors causing an increase in their contributions [25]. As a consequence, the chances of erroneous identification increase. In the RBC and the SVI methods, the faulty sensor data are not used for reconstruction. So, the

effect of a fault in a sensor does not propagate to other sensors and results in better identification; d) Qin [100] showed that the identification performance of the contribution plot approach is dependent on the scaling of variables. In other words, improper scaling might degrade the identification performance of contribution plots.

Despite the superior identification performance of the RBC and the SVI methods as compared to the contribution plot approach, they suffer from the limitations listed below: a) The RBC and the SVI methods are dependent on the accurate reconstructions of faulty data. Their performance might degrade if faulty data cannot be reconstructed accurately; b) The SVI method requires the calculation of a threshold value of SVI, which might need a large amount of historical data [25]; c) SVI experiences oscillations when no sensor fault is present in the system. A filter is required to eliminate the oscillations and reduce the possibility of erroneous identification; d) The RBC and the SVI methods require prior knowledge of the fault directions [100]. This is not an issue for sensor faults, whose fault directions can be easily obtained by utilizing the columns of the identity matrix. These methods might not be able to identify a process fault correctly if no prior knowledge of the corresponding fault direction is available.

The above discussion shows that all three identification methods researched here exhibit a few shortcomings. However, it was observed that the effects of the limitations of the contribution plot approach are more pronounced than the limitations of the RBC and SVI methods, which led to the

superior identification performance of the RBC and the SVI methods as compared to the contribution plot approach, as shown in this work. Hence, we recommend the use of the RBC and the SVI methods to perform the identification/diagnosis of faults in virtual metrology sensors.

### **5.3.5 Fault reconstruction**

In the last subsection, we studied the identification performance of three fault identification methods: the contribution plot method, the RBC method, and the SVI method. After identifying the sensor which caused the fault, the next step is to estimate the size of the fault. The estimation of fault magnitude is very important in order to make accurate predictions using a virtual metrology model. To obtain the fault-free sensor signals that are used as inputs to the virtual metrology model, the effect of the fault needs to be removed from the faulty sensor signal with high precision. Equation 5.14 shows that the fault-free sensor signals are obtained by subtracting the magnitude of the fault from the faulty sensor signal.

The concept of reconstruction was introduced in Sections 5.3.2 and 5.3.3, which presented the RBC method and the SVI method, respectively. Both of these methods were based on the reconstruction of faulty sensor data. Fault reconstruction is revisited in this subsection to explain how the faulty sensor signals are reconstructed after identifying the faults by contribution plot method. The fault reconstruction results for 19 mean faults are also provided in this subsection.

Fault reconstruction is mainly done in three ways: reconstruction via iteration, the missing value approach, and reconstruction via optimization. An interested reader can refer to Qin et al. [104] for details. Qin et al. showed that all three ways of doing reconstruction mentioned above lead to the same results. The magnitude of faults estimated using these methods is same as the magnitude provided in Equation 5.17. Recall that the fault identification methods provide the true fault directions  $\xi_i$ . Knowing the fault directions and the fault magnitudes, we can calculate the reconstructed/fault-free values of the sensor signals using Equation 5.14.

Table 5.4: The estimates of the fault magnitudes for 19 mean faults. Fairly accurate estimates are obtained when faults are correctly identified.

Fault number	Correct identification?	Actual fault magnitude	Estimated fault magnitude
1	Yes	150.33	150.29
2	Yes	150.63	150.59
3	Yes	26.72	26.77
4	Yes	0.008	0.018
5	No	127.67	-13.33
6	Yes	20.11	19.92
7	Yes	246.43	253.86
8	Yes	1886.14	1901.96
9	Yes	1796.00	1796.14
10	No	-125.03	-385.37
11	Yes	5.53	5.51
12	Yes	3335.29	3359.24
13	Yes	3865.10	3803.54
14	Yes	7.83	6.78
15	Yes	3303.44	3263.25
16	Yes	69.79	69.25
17	Yes	0.027	0.041
18	Yes	5639.01	5690.96
19	Yes	9.99	10.11

Table 5.4 provides the faults magnitudes estimated using Equation 5.17 for 19 mean faults. Faults of size equal to 20% of the mean values of the process variables were introduced. We observed that fairly good estimates of the fault

magnitude are obtained by using Equation 5.17 when the faults are identified correctly. In the case of incorrect identification, the fault direction  $\xi_i$  is not known correctly and leads to an erroneous estimation of the fault magnitude.

## 5.4 Conclusions

For a successful implementation of virtual metrology (VM), we need to make sure that the data entering the VM model are free from faults. Sensor faults are the most relevant faults in the context of VM as VM relies on the sensor data to predict the process outputs. When a sensor fault occurs, the corresponding sensor data are erroneous and do not represent the true behavior of the process. The quality of sensor data, which serve as inputs for VM models, has a direct effect on the quality of predicted values. In the presence of faulty input data, an accurate VM model will provide erroneous predictions of the outputs. This situation is known as Garbage-In-Garbage-Out in the process modeling terms. The objective of this chapter was to remove the effect of sensor faults from the sensor data and feed the corrected (reconstructed) sensor data to the VM model.

To achieve the objective stated above, three steps: fault detection, fault identification, and fault reconstruction, were performed. In order to compare the performance of various fault detection and identification methods, a benchmark dataset (see Section 4.3) was utilized. The dataset consisted of processing data of 108 normal wafers and 21 faulty wafers. However, the faults in this dataset were introduced with a limited intention of comparing several

methods in terms of their fault detection performance. These intentionally induced faults serve the purpose of comparing different fault detection methods successfully, but are not suitable for performing fault identification and reconstruction. Fault identification aims at finding the process variable/process variables which caused a fault in a wafer/batch and serves as the basis for obtaining reconstructed (fault-free) data.

In order to perform fault identification and reconstruction effectively, artificial faults were introduced in the data corresponding to 107 normal wafers in this work. For example, a sensor fault in the mean value of a process variable was simulated by adding a constant bias to the setpoint value of the process variable. These artificial faults appear more promising than the ones in the original dataset for performing fault identification and reconstruction as the values of the altered process variables are not reset to their setpoint values. Using the artificial faults, the chance of the altered process variable getting identified as the one causing the fault is much more than its chance while using the faults present in the original dataset.

The first step to achieve the goal of removal of the effect of fault from the faulty sensor data is fault detection. In this chapter, we presented a statistics pattern analysis (SPA) based method that performs PCA on the statistics of the process variables unlike the traditional PCA and MPCA methods that perform PCA on the temporal values of the process variables. The number of detected mean and variance faults by using both the MPCA-based methods were less than those detected by SPA. This is due to the non-Gaussian

characteristics of the data collected from semiconductor manufacturing processes. MPCA-based methods wrongly assume the data to be Gaussian and calculate erroneous control limits for the fault detection indices. The fault detection study concluded that SPA provides better fault detection performance for different types of faults as compared to MPCA-based methods. Not only it detects more faults, SPA also significantly reduces the number of false alarms. Therefore, this study recommends that SPA should be employed as a fault detection method to detect faults in the VM sensors.

The second step to achieve our goal is to perform fault identification, which aims at finding the process variable/process variables which caused a fault in a wafer/batch. Apart from the fact that SPA detected more faults than the two MPCA-based methods, the main benefit of performing identification on the faults detected by SPA is that not only the faulty sensor can be identified, but the statistic (e.g. mean, variance) in which the fault occurred is also identified. This information is crucial for predicting accurate outputs using virtual metrology models, which mostly use the statistics of the sensor signals as the inputs. Therefore, fault identification was performed on the faults detected by SPA in this study.

We presented and implemented three well-known fault identification methods present in literature. Specifically, these included contribution plot approach, reconstruction-based contribution (RBC) approach, and sensor validity index (SVI) approach (see Sections 5.3.1 - 5.3.3 for details). An equation that relates the RBC with the SVI was derived. The RBC method and the SVI

method exhibited similar identification results. This is due to the fact that both these methods are based on the same idea of reconstruction of faults and are related by Equation 5.30. The contribution plot method identified a smaller number of faults correctly as compared to the RBC and the SVI methods because of the limitations listed in Section 5.3.4. The shortcomings of all three identification methods implemented in this study were presented. The effects of the limitations of the contribution plot approach are more pronounced than those of the limitations of the RBC and the SVI methods, which led to the superior identification performance of the RBC and the SVI methods as compared to the contribution plot approach as shown in this work. Hence, we recommend the use of the RBC and the SVI methods to perform the identification/diagnosis of faults in virtual metrology sensors.

After identifying the sensor that caused the fault, the third step to achieve our goal is to perform fault reconstruction, which includes the estimation of the size of the fault. The estimation of fault magnitude is very important in order to make accurate predictions using a virtual metrology model. To obtain the fault-free sensor signals that are fed as inputs to the virtual metrology model, the effect of the fault needs to be removed from the faulty sensor signal with high precision.

Fault reconstruction is mainly done in three ways: reconstruction via iteration, the missing value approach, and reconstruction via optimization. Qin et al. [104] have shown that all three ways of doing reconstruction mentioned above lead to the same results. The magnitude of the fault was estimated by

minimizing the fault detection indices, SPE,  $T^2$ , or  $\phi$  (see Section 5.3.5 for details). Fairly good estimates of the fault magnitude were obtained when the faults were identified correctly. In the case of incorrect identification, the fault direction  $\xi_i$  was not known correctly and led to an erroneous estimation of the fault magnitude.

## Chapter 6

# Improvements in Run-to-Run Process Control Using Virtual Metrology

The previous chapters of the dissertation suggested methods to deal with the noise and faults present in sensor data, which are used as inputs for Virtual Metrology (VM) models. A comparison of various modeling techniques based on both the industrial data and simulated data was also provided. Chapters 2 and 3 of the dissertation focused on the development of mathematical models for VM.

The quality of sensor data, which serve as inputs for the VM model, has a significant effect on the quality of predicted values. In the presence of faulty input data, an accurate VM model might provide erroneous predictions for the outputs. This situation is known as Garbage-In-Garbage-Out in the process modeling terms. Chapters 4 and 5 focused on the detection of sensor faults in the process inputs and suggested methods to isolate the faults and estimate the fault magnitudes. This isolation and estimation allowed us to reconstruct the faulty process inputs and obtain fault-free process inputs. The improvement in the prediction quality of a VM model integrated with an accurate fault detection and identification system was demonstrated.

## 6.1 Introduction

The idea of using estimates made by VM as a substitute for physical metrology seems very alluring at the first sight. VM may significantly reduce the measurement costs as it does not require any actual physical measurements. Substituting the physical measurements completely by VM might be the most economical solution to the problem of high measurement costs, but it might fail in the presence of process disturbance and shifts. If any undesired process change happens, VM model might not be able to compensate for the unknown change in the process and might not be able to predict the outputs accurately. The process operator will be under the false impression that the process is running normally, while the actual processed products will not be on the target.

On the other hand, a combination of physical measurements and VM might be a more robust approach. Instead of blindly relying on the estimates made by VM, the combined approach aims at monitoring the quality of VM estimates and performs a physical measurement whenever the quality of VM estimates falls below a threshold value. More metrology events increase the measurement costs and decrease the product throughput (by increasing cycle time), whereas too few metrology events might hamper the product quality. Therefore, the frequency of metrology events needs to be optimized. Thus, the implementation of the combined approach requires the development of optimal sampling plans that will tell the semiconductor manufacturers when to perform a physical measurement to supplement VM predictions.

In the context of deciding which wafers or products should be physi-

cally measured, the terms sampling and scheduling represent the same concept. Scheduling the metrology events is equivalent to sampling the wafers to be measured. Whenever the VM prediction accuracy falls below a certain threshold, an actual measurement should be done by the metrology tool and the VM model should be updated. An intuitive solution is to do more frequent physical measurements when VM predictions are quite different from the metrology values and update the prediction model. Some work [88, 131] on optimal sampling is present in semiconductor manufacturing literature, but not in the context of virtual metrology.

Scheduling of semiconductor processes can be done in various ways leading to different objectives. So, each schedule is tailored in order to reach a certain goal. Pasadyn et al. [93, 94] worked on identifying the wafer samples that should be measured in order to extract more information about the system from the measurements. The optimal samples were found by minimizing the trace of state error covariance matrix. A good literature review on optimal sampling is available in Lee et al. [70]. A comparison of uniform sampling, dynamic (multi-rate) sampling, random sampling, and hybrid sampling based on an industrial dataset for a thin film deposition process was reported. Dynamic sampling method yielded good results and was also found to be applicable to real manufacturing.

In this chapter, first we will simulate a Single-Input-Single-Output (SISO) process with process drift and noise. Run-to-Run (R2R) control will be employed to adjust the recipe settings (inputs) to ensure that the output

stays on the target in the presence of process drift and noise. After discussing the working of R2R control for the SISO in detail to obtain a good understanding, we will implement R2R control on a Multiple-Input-Multiple-Output (MIMO) model present in the VM literature (see Section 6.4.2). In general, the implementation of R2R control includes the estimation of process gain matrix, process drift, or both. In semiconductor manufacturing, process drift is a major issue of concern as process gain matrix remains almost constant owing to the physics and chemistry behind the process. So, in this work the process drift will be estimated using the measurements done according to the sampling plan. Whenever a measurement is made, the value of process drift is estimated by Exponentially-Weighted-Moving-Average (EWMA), a weighted average of the previous estimate of the process drift and the process drift value suggested by the current measurement.

Devising an optimal sampling plan is critical in order to ensure that the process outputs are on target, while not spending a large amount of money by measuring many products. We will implement three commonly known sampling methods, uniform sampling, random sampling, and dynamic sampling, in order to demonstrate the superior performance of a novel reliance index based sampling method that utilizes VM estimates. The most common sampling strategy is uniform sampling, which measures a product after a fixed interval of time or products. Random sampling does not have a fixed measurement interval, but measures the products at random intervals that have specified lower and upper limits. Both of these methods do not take advantage of the

known past and current behavior of the process. Dynamic sampling is based on the intuitive idea of measuring more products when the process seems to drift away from the target and measuring fewer products when the process outputs are fairly close to the target. Bayesian detection approach adopted by Lee et al. [70] is employed to implement dynamic sampling in this work. The Bayesian detection approach calculates a posterior probability distribution using a prior probability distribution and the observed data. When the probability that the currently observed data is coming from a drifting process exceeds a threshold value, the sampling frequency is increased. An improved dynamic sampling approach, which updates the value of EWMA forgetting factor  $\lambda$  using Bayesian detection, is proposed in this work.

In the three sampling methods mentioned above, uniform sampling, random sampling, and dynamic sampling, the estimates of the process drift and the recipe settings (inputs) of the process are only updated when a physical measurement was made as dictated by the sampling plan. Using the predictions made by VM model, it is possible to make these updates even when a physical measurement is not done. VM enables us to update the estimate of process drift and the recipe settings of the process after processing each product wafer irrespective of the fact whether the wafer was physically measured or not. An accurate VM model will ensure reduced measurement costs and better controller performance. After processing each wafer, a decision whether the most recent wafer should be measured or not needs to be made. This can be decided by calculating a reliance index that quantifies how much a manu-

facturer can rely on the VM estimate. If the value of calculated reliance index is below a certain threshold, a physical measurement needs to be made as the manufacturer cannot rely on the VM estimate. Some work on a reliance index is present in VM literature [16] but it suffers from a few shortcomings (see Section 6.5.5 for details). A new reliance index, which is more attractive from a mathematical and practical point of view, will be proposed in this work.

## 6.2 Sampling Methods

In this section, three commonly-used sampling methods will be presented. These sampling methods will determine which wafers should be measured in order to update the R2R controller and ensure that the process outputs are on the target. After simulating a SISO and a MIMO process, a detailed discussion about the pros and cons of these methods will be provided in Sections 6.5.1 and 6.5.3.

### 6.2.1 Uniform sampling

Uniform sampling means the sampling of wafers at fixed intervals. In other words, a wafer is measured after processing a certain number of wafers. Suppose a sequence of wafers is being processed in a fab and the  $k^{th}$  wafer is measured. If  $s$  is the sampling interval, the  $(k + s)^{th}$  wafer will be the one to be measured next. For example, the sampling interval is 3, the wafers that would be measured are wafer numbers  $k+3$ ,  $k+6$ ,  $k+9$ , and so on. Sampling interval effects the performance of a R2R controller based on uniform sampling.

A smaller sampling interval would mean more frequent sampling. The R2R controller will be updated frequently leading to good control performance and the outputs will be close to the target. A larger sampling interval would lead to less frequent sampling, and less frequent update of R2R controller, leading to sluggish adaptation of the controller to process drift.

Figure 6.1 shows the variation of Mean Squared Error (MSE) with sampling interval for uniform sampling. Mean squared errors are the means of squared errors between the output controlled by the R2R controller and the target value. We observe that as the sampling interval increases, the MSE value also increases because of the infrequent updates of the R2R controller. This suggests that the sampling interval value should be chosen to be 1 as it provides the least value of MSE. However, too frequent measurements will lead to high measurement costs. Practically, the optimum sampling interval is decided by the trade-off between the desired product quality and the measurements costs.

### 6.2.2 Random sampling

Random sampling does not have a fixed measurement interval, but measures the products at random intervals that have specified lower and upper limits. Suppose a sequence of wafers is being processed in a fab and the  $k^{th}$  wafer is measured. If  $a$  and  $b$  are the lower and upper limits of the intervals, the next wafer to be measured could be any wafer between  $(k + a)^{th}$  wafer and  $(k + b)^{th}$  wafer. The probability that which wafers get measured will

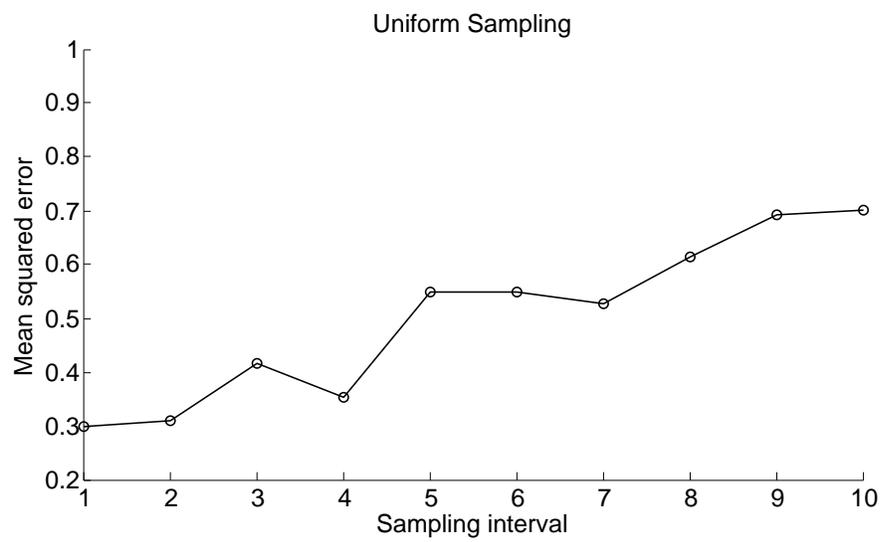


Figure 6.1: Variation of mean squared error (MSE) with sampling interval for uniform sampling. As the sampling interval increases, measurements are done less frequently causing sluggish adaptation of R2R controller to the process drift.

be determined by a probability density function. In general, the wafers are given equal probability to be picked for measurement, which corresponds to a uniform probability distribution function.

For implementation, we run the random number generator based on uniform distribution to determine the next wafer to be measured. It should be noted that uniform probability distribution function is used in random sampling for generating a random number between specified limits and should not be confused with uniform sampling, which measures wafers after a fixed interval. For practical purposes, it has been found that random sampling is rarely used in the industry due to its random nature. The next subsection presents dynamic sampling, which has a variable sampling rate that changes according to the process behavior.

### **6.2.3 Dynamic sampling**

Both the sampling methods discussed earlier, uniform sampling and random sampling, do not take advantage of the known past and current behavior of the process. Dynamic sampling is based on the intuitive idea of measuring more products when the process seems to drift away from the target and measuring fewer products when the process outputs are fairly close to the target. In the context of state estimation, Bayesian detection was first proposed by Wang and He [127, 128] to improve the state estimation performance. Bayesian detection approach adopted by Lee et al. [70] is employed to implement dynamic sampling in this work. The Bayesian detection ap-

proach calculates a posterior probability distribution using a prior probability distribution and the observed data. When the probability that the currently observed data is contaminated with process shift or drift exceeds a threshold value, the sampling frequency is increased.

The basic principle behind Bayesian detection approach is Bayes' theorem [27, 39, 51]. Suppose our objective is to estimate the values of a set of parameters  $\Theta$  for some data set  $D$  generated from an underlying model. For any given model, one can write an expression for the likelihood function  $P(D|\Theta)$  of obtaining the data vector  $D$  given a particular set of values for the parameters  $\Theta$ . In addition to the likelihood function, one may impose a prior distribution  $P(\Theta)$  on the parameters, which represents our state of knowledge regarding the values of the parameters before analyzing the data  $D$ . Bayes' theorem calculates the posterior probability as:

$$P(\Theta|D) = \frac{P(D|\Theta)P(\Theta)}{P(D)} \quad (6.1)$$

which gives the posterior distribution  $P(\Theta|D)$  in terms of the likelihood, the prior, and the evidence  $P(D)$ [9].

If a step change occurs in a process with output vectors  $X_k$ , the posterior probability is generated by computing the joint posterior probability for each subset of the post-change window,  $X_k$ , where  $X_k \equiv \{x_1, x_2, \dots, x_k\}$ . Assuming that the mean of samples in the pre-change window is zero, the step magnitude  $\mu_D$  is calculated as the mean of  $X_k$  by the following equation.

$$\mu_D = \frac{\sum_{i=1}^k x_i}{k} \quad (6.2)$$

The probability density functions for normal and shifted process states are denoted by  $N(0, \sigma^2)$  and  $N(\mu_D, \sigma^2)$ , respectively (assuming Gaussian distribution), where  $\sigma$  is the process standard deviation. The likelihood function of a step disturbance for a single observation  $x_i$  is given by:

$$p(x_i|\Theta_D) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_i - \mu_D)^2}{2\sigma^2}\right] \quad (6.3)$$

If all the vectors in  $X_k(x_1, x_2, \dots, x_k)$  are independent and identically distributed, the likelihood function of  $X_k$  is given by the product of individual likelihood functions of the vectors.

$$P(X_i|\Theta_D) = \prod_{i=1}^k p(x_i|\Theta_D) \quad (6.4)$$

Using Equation 6.3, Equation 6.4 can be rewritten as:

$$P(X_i|\Theta_D) = \frac{1}{(\sqrt{2\pi\sigma^2})^k} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^k (x_i - \mu_D)^2\right] \quad (6.5)$$

An analogous function for the normal data (no shift) can be obtained by setting  $\mu_D$  equal to zero.

$$P(X_i|\Theta_N) = \frac{1}{(\sqrt{2\pi\sigma^2})^k} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^k x_i^2\right] \quad (6.6)$$

Figure 6.2 shows the normal data and the shifted data. The onset location is the point where the shift occurs.

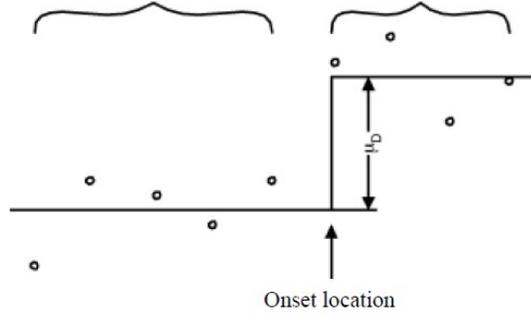


Figure 6.2: Normal data and the shifted data

Setting  $X_k$  as the data vector  $D$  in Equation 6.1,

$$P(\Theta_D|X_k) = \frac{P(X_k|\Theta_D)P(\Theta_D)}{P(D)} \quad (6.7)$$

The evidence  $P(D)$  mentioned in Equation 6.7 is a summation of two terms. The first term is the probability of observing data vectors  $X_k$  given that the shift has occurred multiplied by the prior probability of the shift. The second term is the probability of observing data vectors  $X_k$  given that no shift has occurred (normal state) multiplied by the prior probability of the normal state.

$$P(D) = P(X_k|\Theta_D)P(\Theta_D) + P(X_k|\Theta_N)(1 - P(\Theta_D)) \quad (6.8)$$

Substituting Equation 6.8 in Equation 6.7 gives:

$$P(\Theta_D|X_k) = \frac{P(X_k|\Theta_D)P(\Theta_D)}{P(X_k|\Theta_D)P(\Theta_D) + P(X_k|\Theta_N)(1 - P(\Theta_D))} \quad (6.9)$$

Plugging the conditional probabilities given by Equation 6.5 and Equation 6.6 in Equation 6.9 and setting the prior probability  $P(\Theta_D) = P_o$  yields:

$$P(\Theta_D|X_k) = \frac{P_o}{P_o + (1 - P_o)\exp\left[-\frac{(\sum_{i=1}^k x_i)^2}{2k\sigma^2}\right]} \quad (6.10)$$

After the step disturbance, the posterior probability of the first point is given by:

$$P(\Theta_D|x_1) = \frac{P_o}{P_o + (1 - P_o)\exp\left[-\frac{x_1^2}{2\sigma^2}\right]} \quad (6.11)$$

$P(\Theta_D|x_1)$  represents the probability that data point  $x_1$  is coming from a shifted process. If this probability is close to one, we are almost certain that a shift occurred in the process. A probability close to zero would indicate that  $x_1$  originated from a normal process. Recalling the idea behind dynamic sampling, our goal is to sample more frequently when a process shift occurs. Hence, the sampling frequency is increased (sampling interval is shortened) as  $P(\Theta_D|x_1)$  increases. The values of  $P(\Theta_D|x_1)$  at which the sampling frequency is changed and the magnitude of the change are the tuning parameters of this approach. A user can adjust these parameters according to the process knowledge and the economical constraints.

Employing a Bayesian detection approach for dynamic sampling offers the following two advantages as compared to an approach that simply monitors the observations  $X$  to determine the sampling interval. This approach will be referred to as  $3\sigma$  approach. Firstly, the calculated posterior probabilities  $P(\Theta_D|X_k)$  of Bayesian detection approach are bounded with a minimum value equal to the prior probability  $P_o$  and a maximum probability approaching one. No such bounded decision parameters exist for the  $3\sigma$  approach. Secondly, the Bayesian detection approach calculates the posterior probability by examining the size of observed data  $x$  relative to the process standard deviation  $\sigma$ . This ensures the correct calculation of the posterior probabilities of any process with a given standard deviation  $\sigma$ .

Figure 6.3 provides an insight into the implementation of dynamic sampling using Bayes' theorem. The Gaussian curve in blue color represents normal process behavior. Observed data  $x$  is plotted on  $x$  axis and the probability of observing data  $x$  is plotted on  $y$  axis. In this figure, normal data has a mean value of zero and a standard deviation of 2. The Gaussian curve in red color represents the process data after a shift has occurred in the process. The shifted data has a mean value of 5 and a standard deviation of 2. So, the magnitude of shift in the process is 5.

Recalling Equation 6.9, the posterior probability given by Bayes' theorem is calculated as:

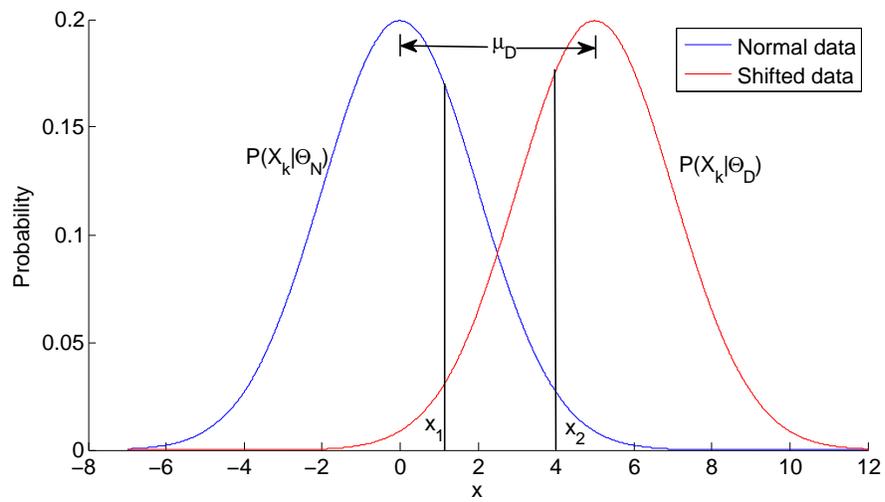


Figure 6.3: Normal data and shifted data to demonstrate how Bayes' theorem is used in dynamic sampling

$$P(\Theta_D|X_k) = \frac{P(X_k|\Theta_D)P(\Theta_D)}{P(X_k|\Theta_D)P(\Theta_D) + P(X_k|\Theta_N)(1 - P(\Theta_D))} \quad (6.12)$$

In Figure 6.3, suppose  $x_1$  is observed as a data point. We want to know what is the probability that it originated from a shifted process. It is clear from Figure 6.3 that  $P(X_k|\Theta_N)$  is much larger than  $P(X_k|\Theta_D)$  at  $x_1$ . Equation 6.12 will provide a small value for  $P(\Theta_D|X_k)$ , which means that the probability that  $x_1$  comes from a shifted process is low. This is in accordance with our knowledge that  $x_1$  is closer to the normal mean value of zero than the shifted mean value of 5. So, dynamic sampling suggests to reduce or maintain the sampling frequency after observing data point  $x_1$ .

On the contrary, suppose  $x_2$  is observed as a data point. It is clear from Figure 6.3 that  $P(X_k|\Theta_D)$  is much larger than  $P(X_k|\Theta_N)$  at  $x_2$ . Equation 6.12 will provide a large value for  $P(\Theta_D|X_k)$ , which means that the probability that  $x_2$  comes from a shifted process is high. This is in accordance with our knowledge that  $x_2$  is closer to the shifted mean value of 5 than the normal mean value of zero. As a result, dynamic sampling will recommend an increase in the sampling frequency so that prompt control action can be taken to bring the process back in the normal operation region.

The next section presents the implementation of run-to-run control that will ensure the process outputs to be on target. Three sampling methods discussed in this section, uniform sampling, random sampling, and dynamic sampling, will be used to decide which products should be measured. When-

ever a product is measured, the parameters of the controller are updated and the recipe settings (inputs) for the next product are calculated. Hence, an appropriate choice of the products to be measured is critical to achieve good control performance while maintaining low measurement costs.

### **6.3 Run-to-Run Control**

During the past two decades, the semiconductor industry has been continuously progressing towards smaller feature sizes and larger wafer dimensions. Although a number of solutions, including improved equipment design and process innovation, will continue to aid in making these transitions cost effective, it has become clear that they are no longer sufficient. Specifically, it has become generally accepted that process and wafer quality sensing and subsequent process tuning will be required to complement these equipment and process improvements. The main form of process tuning that is being implemented as a standard process and equipment control solution in the industry is run-to-run (R2R) control. R2R control is now a proven and available technology, and has become a critical component of the success of existing and next-generation fabrication facilities also known as fabs [82]. Applications of R2R control by Bode [6], Campbell [11], and Edgar et al. [26] have shown that multivariable control with constraint-handling capability offers definite benefits over conventional control strategies for semiconductor manufacturing.

Run-to-run control is a form of discrete process and machine control in which the product recipe with respect to a particular machine process is

modified *ex situ*, i.e., between machine runs to minimize process drift, shift, and variability. This type of control is event-driven, where the events include the determination and reporting of pre- and/or postprocess *ex situ* metrology data.

There are many ways one can design a R2R controller, and indeed many different types of R2R control algorithms have been developed and implemented both in industry and academia. Offset drift cancellation approaches are useful to counteract the undesired shift/drift that happens in the process. The idea behind these approaches is to estimate the current offset term and select an input setting to compensate for the offset. In these approaches, the term offset means a fixed or variable departure away from the expected output value. If this departure is constant, it is called shift and if it is variable, it is referred to as drift. The reader should not get confused with the usage of terms drift and shift in this dissertation as they imply the same idea, the only difference being that drifts are variable and a shift corresponds to a fixed value.

One important offset drift cancellation approach is that of Exponentially-Weighted-Moving-Average (EWMA) R2R control [83]. This will be discussed in detail in the next sub-section.

### **6.3.1 Exponentially-Weighted-Moving-Average R2R control**

In EWMA-R2R control, the offset or shift term is estimated using a EWMA equation. The idea is to estimate the current shift term as a weighted

average of the previous estimate of the shift term and the value of shift term suggested by the most recent measurement. Equation 6.13 represents a linear process model with outputs  $y$ , inputs  $u$ , process gain matrix  $A$ , offset/shift term  $b$ , and Gaussian noise  $e$  with mean zero and standard deviation  $\sigma$ . The subscript  $k$  can be referred to processing time or the wafer being processed. If the process model has  $p$  inputs and  $q$  outputs,  $u_k$  will be a row vector of size  $1 \times p$ .  $y_k$ ,  $b_k$ , and  $e_k$  will be row vectors of size  $1 \times q$  and  $A$  will be a matrix of size  $p \times q$ .

$$y_k = u_k A + b_k + e_k \quad (6.13)$$

Equation 6.13 shows that the offset term  $b_k$  directly affects the outputs  $y_k$ . Any undesired change in the offset is detrimental to the quality of process control. But if this offset can be estimated fairly accurately, it is possible to minimize or nearly eliminate the effect of the undesired change in the offset on the outputs. Equation 6.14 estimates the offset term using an EWMA equation.  $\lambda$  is known as forgetting factor of the EWMA equation because it weights the previous value of the offset term.

$$\hat{b}_{k+1} = (1 - \lambda)\hat{b}_k + \lambda(y_k - u_k A) \quad (6.14)$$

The value of  $\lambda$  is chosen to be a fraction between 0 and 1. The choice of  $\lambda$  is the trade-off between the amount of noise filtering and the speed of adaptation to the process shift. A value of  $\lambda$  near zero provides a small weighting

of the recent information and updates the offset estimate very conservatively. If a process shift happens, the adaptation of the offset terms to this new value would be very sluggish and lead to a poor controller performance. On the positive side, updating the offset term conservatively shields it from the noise present in the measured data.

On the contrary, a value of  $\lambda$  near one provides a large weightage to the recent information and updates the offset estimate very aggressively. As a result, the offset term quickly adapts to the new value if a process shift happens. The only disadvantage in choosing a large value of  $\lambda$  is that it is more sensitive to the effects of noise present in the measured data. By providing a larger weight to the recently measured values, the estimate of the offset term might show random fluctuations driven by noise.

Once the value of offset term is estimated fairly accurately, this information can be used to adjust the recipe settings (inputs) of the process to compensate for the offset to ensure the products are on target. The assumption here is that the offset term for the next wafer is not much different from the one estimated using Equation 6.14. As the desired value of outputs  $y$  is the target value, the inputs for the next wafer can be calculated from Equation 6.13 as:

$$u_{k+1} = (Target - \hat{b}_{k+1})A^{-1} \quad (6.15)$$

The expected value of outputs by feeding the inputs given by Equation

6.15 to the process will be:

$$\hat{y}_{k+1} = Target - \hat{b}_{k+1} + b_{k+1} + e_{k+1} \quad (6.16)$$

It can be seen from Equation 6.16 that if the offset term is perfectly estimated, i.e.,  $\hat{b}_{k+1} = b_{k+1}$ , the outputs will differ from the target only due to noise, which is of much smaller magnitude than that of the process shifts. This is a case when the effect of offset has been completely canceled by the EWMA-R2R controller.

The implementation of R2R control on a linear multivariable process given by Equation 6.13 is summarized in Figure 6.4. The figure represents a general case when the drift might be present in the parameters of process gain matrix  $A$  or offset term  $b$ , or both.

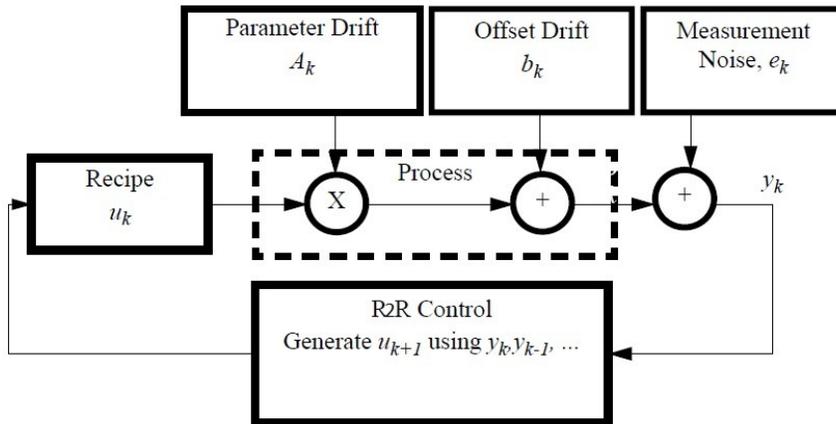


Figure 6.4: Implementation of R2R control on a linear process

Equation 6.13 represents a generic model for linear processes. First, the

implementation of EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling will be demonstrated on a Single-Input-Single-Output (SISO) model in Section 6.5. SISO model can be obtained by setting the process gain matrix  $A$  in Equation 6.13 to 1. Next, EWMA-R2R control will be implemented on a Multi-Input-Multi-Output (MIMO) model present in VM literature [41, 63] that represents a generic semiconductor process. The next section introduces these two models.

## 6.4 Models

R2R control requires a model of how the outputs of a process are related to the inputs, which can include process settings and incoming wafer characteristics (outputs of the previous processes). Often it is not necessary to have an extremely accurate or detailed model. Control strategies involve making modest adjustments to input settings to counteract drifts in the process behavior. Consequently, the first-order sensitivities are all that is required for control. The majority of the process models used in semiconductor industry are linear in nature as they are simple, easy to analyze, and explain the process behavior fairly well. Two linear process models that will be employed in this chapter are presented in the following sub-sections.

### 6.4.1 SISO model

For easier understanding of EWMA-R2R control based on different sampling methods, the simulations using a SISO model will be presented first.

The SISO model shown in Equation 6.17 can be obtained by setting the process gain matrix  $A$  in Equation 6.13 to 1. This model has been widely used for lithography overlay control [7, 74], which aims at the proper alignment of the lithography layers. Lithography overlay control is performed by modifying adjustable exposure system controls to align each successive pattern in a device. In most cases, the differences between nominal stage positions in different tools cause overlay errors.

$$y_k = u_k + b_k + e_k \quad (6.17)$$

Equation 6.17 represents a linear SISO model with output  $y$ , input  $u$ , offset/shift term  $b$ , and Gaussian noise  $e$  with mean zero and standard deviation  $\sigma$ . The subscript  $k$  can be referred to processing time or the wafer being processed. If  $n$  process runs are simulated using this model, all the vectors ( $y$ ,  $u$ ,  $b$ , and  $e$ ) will be of size  $nx1$ .

#### 6.4.2 MIMO model

To compare the performance of EWMA-R2R control using different sampling methods for semiconductor manufacturing, a process model that imitates a typical semiconductor process must be employed. One such model has been proposed by Moyne et al. and Han et al. [41, 63]. The underlying structure is a linear model given by Equation 6.13.

$$y_k = u_k A + b_k + e_k \quad (6.18)$$

Let us assume that the process represented by Equation 6.18 can be described by two input variables ( $u_1$  and  $u_2$ ), six process variables ( $v_1, v_2, v_3, v_4, v_5$ , and  $v_6$ ), and two output variables ( $y_1$  and  $y_2$ ). The process variables and the process gain matrix are chosen such that the simulation study closely mimics the true behavior of a semiconductor manufacturing process; some process variables are correlated with each other, not all the process variables depend on the inputs, there is no 1-1 relationship between the inputs and the outputs, and the output variables experience different amounts of drifts and noise. Taking these features into consideration, the process variables and the process gain matrix can be represented by Equations 6.19 - 6.25.

$$v_{1k} = 0.3u_{1k} + 0.4u_{2k} + 0.7 \quad (6.19)$$

$$v_{2k} = 0.2v_{1k} \quad (6.20)$$

$$v_{3k} = 0.2u_{1k} + 0.2b_{1k} + 0.1 \quad (6.21)$$

$$v_{4k} = 0.7u_{1k} + 0.5u_{2k} + 0.3b_{1k} + 0.8b_{2k} + 0.4 \quad (6.22)$$

$$v_{5k} = 0.2v_{1k} - 0.1v_{4k} \quad (6.23)$$

$$v_{6k} = 1.5 \quad (6.24)$$

$$A = \begin{bmatrix} 0.50 & -0.20 \\ 0.25 & 0.15 \end{bmatrix} \quad (6.25)$$

EWMA-R2R control based on uniform sampling, random sampling, and dynamic sampling only utilizes the inputs  $u$  and the outputs  $y$  to update the estimates of the offset term. In the above equations, the process variables

$v$  are provided for VM assisted EWMA-R2R control, which will be explained in the next section. These process variables will serve as inputs for the VM model to provide estimates of the outputs, which will act as substitutes for the physical measurements to bring down the metrology costs. The next section presents the simulation results of EWMA-R2R control on two models, SISO and MIMO, discussed in this section.

## 6.5 Results and Discussion

In this section, the simulation results for the SISO model are presented first for the better understanding of the reader. EWMA-R2R control results using uniform sampling, random sampling, and dynamic sampling for the SISO model are shown in Figure 6.5. For details, see Section 6.2 for sampling methods, Section 6.3 for EWMA-R2R control, and Section 6.4 for the SISO and MIMO models.

### 6.5.1 EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling for SISO model

Figure 6.5 uses data simulated for 250 wafers using Equation 6.17. Process shifts were introduced into the process output during processing of wafer numbers 76, 126, and 201. These shifts were of magnitude equal to 3, -1, and 1 away from the target value of zero. For uniform sampling, a sampling interval of 3 was chosen. For random sampling, the next wafer to be measured was determined by randomly picking an integer from the closed interval, [1 5].

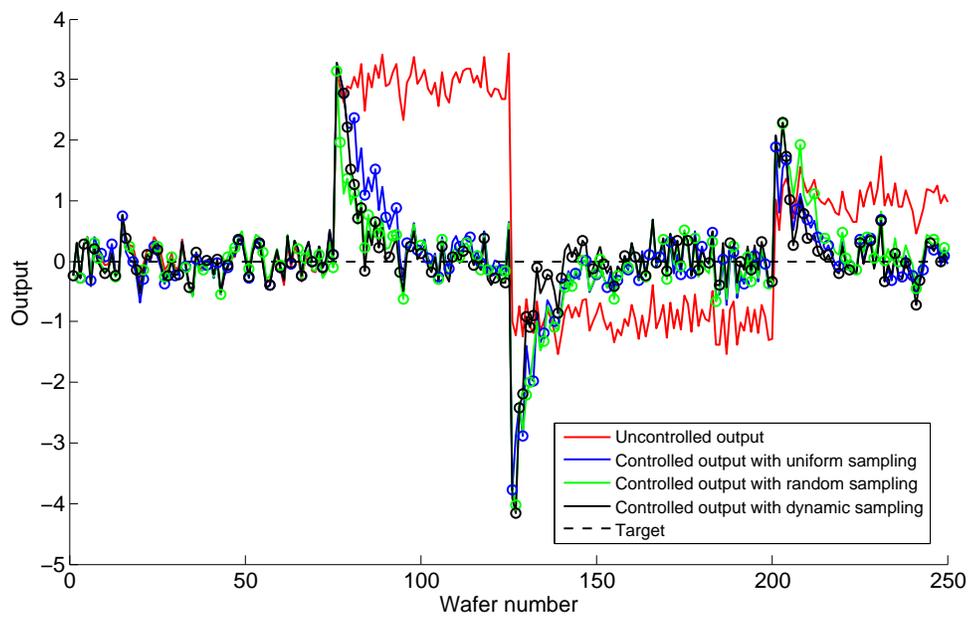


Figure 6.5: EWMA-R2R Control Using Different Sampling Methods for SISO Model

For dynamic sampling, the prior probability that the observed data is coming from a shifted process,  $P(\Theta)$  was set to 0.1 and the process standard deviation,  $\sigma$  was set to 0.25. If the posterior probability was found to be more than 0.6, the sampling interval was shortened to 1, i.e., the next processed wafer will be measured. This interval was relaxed to 2 if the posterior probability value was between 0.3 and 0.6. The sampling interval was set to 3 if the calculated value of posterior probability was less than 0.3, indicating that the data arose from normal process.

The circles in Figure 6.5 correspond to the wafers that are measured for a particular sampling method denoted by the color of the circle. For example, black circles stand for the wafers that are measured when dynamic sampling is employed.

Figure 6.5 shows that the process output can be controlled fairly well using EWMA-R2R control in presence of process shifts as blue, green, and black lines are much closer to the target than the red line. A closer look reveals that dynamic sampling provides better controller performance than uniform sampling and random sampling. The adjustments made in the sampling rate according to the observed output values is the reason behind the superior performance of dynamic sampling as compared to uniform sampling and random sampling. When a shift occurs, consecutive wafers are measured until the posterior probability value is less than a certain threshold value, which is set to 0.6 in this case. The overall effect of dynamic sampling is that we can achieve much better control performance by measuring a few more

wafers.

Another interesting result to be noted is the superior controller performance achieved by using uniform sampling than random sampling. The reason is the random nature of choosing the next wafer to be measured in random sampling. If a shift occurs, it is possible that the next wafer that is measured is the one at the maximum allowed interval from the current one (5 in this case). This will lead to a delayed update of the offset term and cause poor control performance. On the other hand, uniform sampling measures every 3<sup>rd</sup> wafer in this case and updates the offset term more consistently. The controller performance for different sampling methods is quantified by calculating the difference between the controlled value of the output and the target value. Specifically, mean squared error (MSE) can be calculated for each sampling method using Equation 6.26.

$$MSE = \frac{\sum_{i=1}^n (y_i - Target)^2}{n} \quad (6.26)$$

In Equation 6.26,  $y_i$  corresponds to the controlled value of the output for  $i^{th}$  wafer and  $n$  is the total number of wafers, which is equal to 250 in this study.

The calculated MSE values are provided in Table 6.1 along with the number of wafers measured by each sampling method. It should be noted that for random sampling, the numbers provided in the table are the average of 100 simulations of random sampling in order to reflect the true behavior of

random sampling. If these results were provided based on only one simulation, the randomness might provide misleading conclusions.

Table 6.1: EWMA-R2R controller performance for SISO model using different sampling methods

Control scheme	MSE	Number of wafers measured
Uncontrolled output	2.3055	0
Uniform sampling	0.5768	83
Random sampling	0.6650	81
Dynamic sampling	0.4767	97

Figure 6.6 plots the output  $y$ , estimated offset term  $\hat{b}$ , and the input calculated by the EWMA-R2R controller against the wafer number. It can be seen that estimated values of the offset term are fairly close to the actual values, which were preset and used to simulate the model. EWMA-R2R controller calculates the values of inputs for the next wafers using Equation 6.15. For SISO model, process gain matrix  $A = 1$  and target is zero. So the value of input for the next wafer to be processed is set to negative of the estimated value of offset term. The bottom plot in Figure 6.6 shows the output controlled by EWMA-R2R controller using dynamic sampling. For most of the wafers, the output is close to the target value of zero due to good estimation of the offset term  $b$ . The small fluctuations in the output arise from measurement noise. We can observe that whenever a process shift occurs, the controlled output values deviate from the target. This deviation takes place because the EWMA equation takes some time to adapt to the new value of offset term. The value of EWMA forgetting factor  $\lambda$  was set to 0.3 for these simulations.

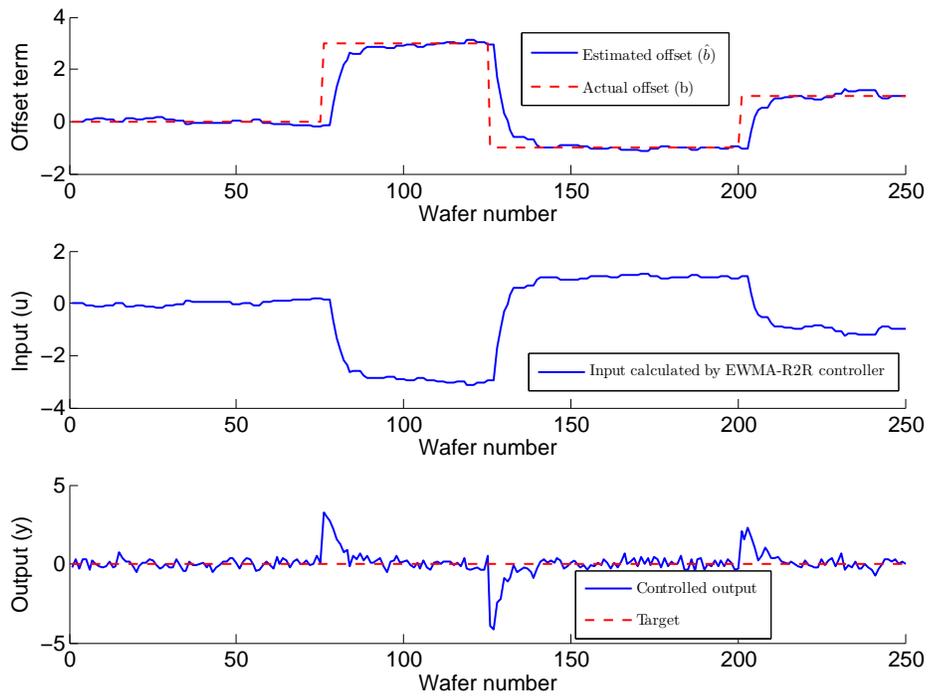


Figure 6.6: Dynamic sampling results for the SISO process with  $\lambda = 0.3$

### 6.5.2 Effect of EWMA forgetting factor

Figures 6.7 and 6.8 show the simulation results for the SISO model for  $\lambda = 0.1$  and  $\lambda = 0.7$ , respectively. A small value of  $\lambda$  updates the estimate of the offset term sluggishly on the occurrence of a process shift. This can be observed in Figure 6.7, where the estimates adapt to the shifted value of the offset term slowly. This sluggish behavior further propagates into the inputs calculated by the controller and the controlled outputs. On the other hand, a large value of  $\lambda$  provides quick adaptation to the shifted value of the offset term, but is not able to filter out the measurement noise appropriately. As a result, the estimates have more fluctuations as compared to those obtained using smaller values of  $\lambda$ . These fluctuations propagate into the inputs calculated by the controller and the controlled outputs. Due to these limitations, most industrial applications use a value in the range 0.2 - 0.4.

### 6.5.3 EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling for MIMO model

Section 6.5.1 provided simulation results for different sampling methods for the SISO model. This section provides similar analysis for the MIMO model presented in Section 6.4.2. Figure 6.9 shows the performance of EWMA-R2R controller using different sampling methods for the two outputs of the MIMO model. The nominal values of the inputs  $u_1$  and  $u_2$  are set to 0.4 and 0.6, respectively. The targets calculated as  $u^*A$  are 0.35 for  $y_1$  and 0.01 for  $y_2$ . As we have two process outputs now, the conditions that determine the change

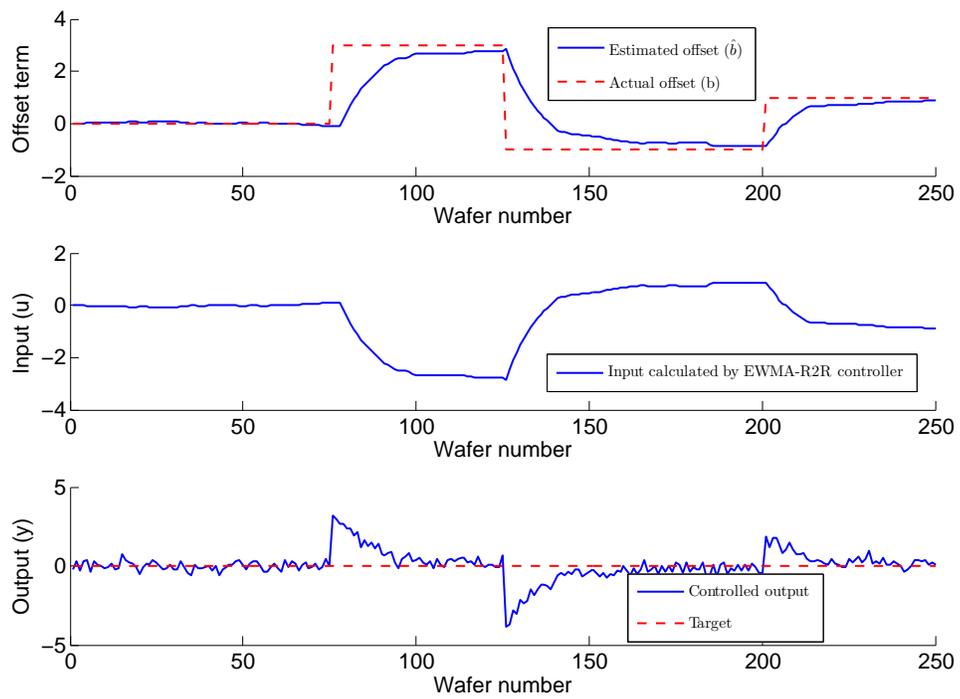


Figure 6.7: Dynamic sampling results for the SISO process with  $\lambda = 0.1$

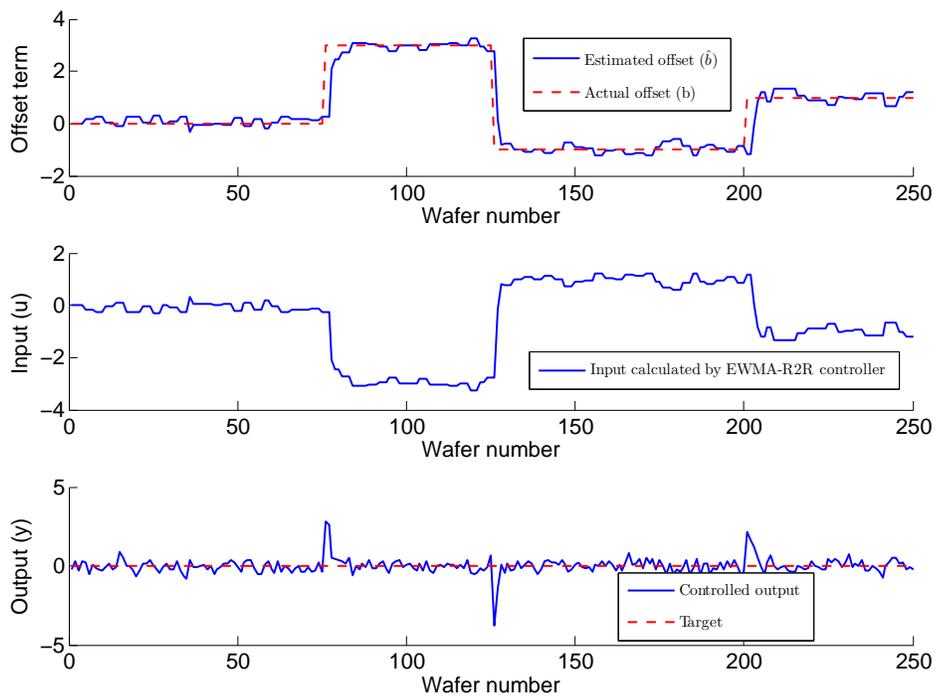


Figure 6.8: Dynamic sampling results for the SISO process with  $\lambda = 0.7$

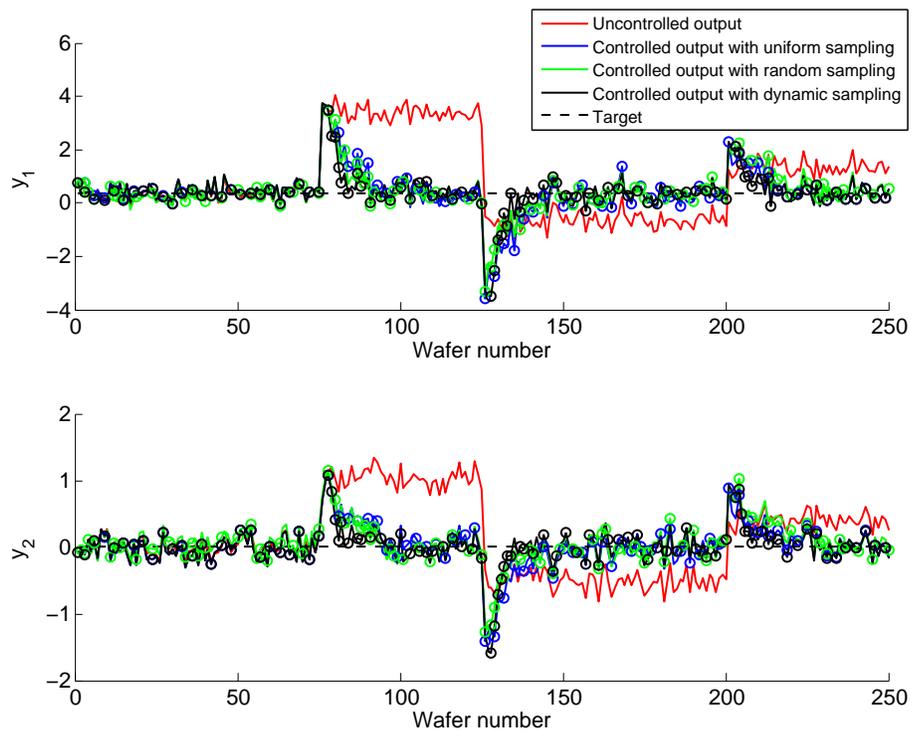


Figure 6.9: EWMA-R2R Control Using Different Sampling Methods for MIMO Model

of sampling interval need to be redefined. The posterior probabilities of both the outputs need to be monitored to change the sampling interval in a timely fashion. If one of the outputs has a value of posterior probability greater than 0.6, the sampling interval is reduced from 3 to 1. If both of the outputs have their posterior probabilities less than 0.3 indicating normal operation, the sampling interval is set to the normal value of 3. For all other cases not covered by the preceding two statements, the sampling interval is set to 2. The rest of the parameters were kept same as in the case of SISO model.

Table 6.2 provides the MSE values using different sampling methods for the two outputs,  $y_1$  and  $y_2$  of the MIMO model. It is evident that EWMA-R2R control using dynamic sampling outperforms the control performance obtained using uniform sampling and random sampling for both the outputs. Also, uniform sampling provides superior results than random sampling because of the reason outlined in Section 6.5.1.

Table 6.2: EWMA-R2R controller performance for MIMO model using different sampling methods

Control scheme	MSE ( $y_1$ )	MSE ( $y_2$ )	Number of wafers measured
Uncontrolled output	2.4347	0.3377	0
Uniform sampling	0.6494	0.0954	83
Random sampling	0.7264	0.1152	84
Dynamic sampling	0.5433	0.0843	99

Figures 6.10 and 6.11 show the timeline of offset terms, inputs calculated by the R2R controller and the controlled outputs. These results are based on a  $\lambda$  value of 0.3. We can see that the estimates of the offset term

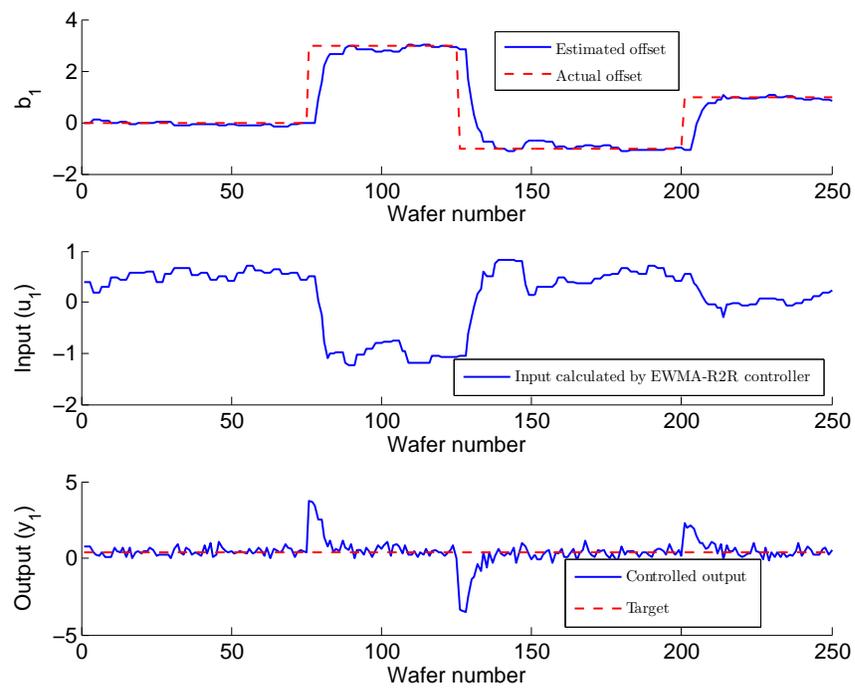


Figure 6.10: Dynamic sampling results for  $b_1$ ,  $u_1$ , and  $y_1$  of MIMO process with  $\lambda = 0.3$

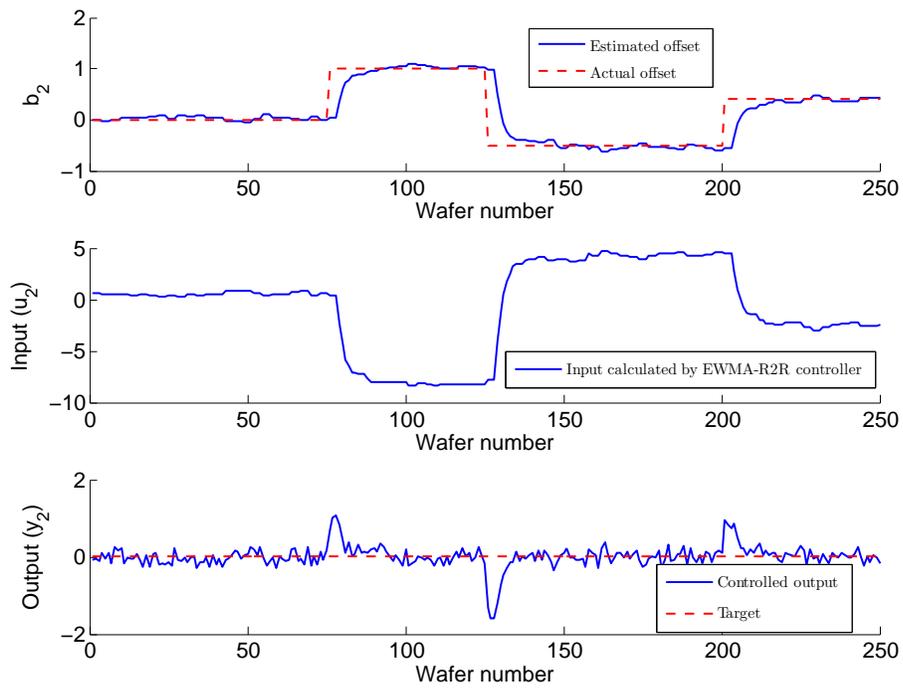


Figure 6.11: Dynamic sampling results for  $b_2$ ,  $u_2$ , and  $y_2$  of MIMO process with  $\lambda = 0.3$

adapt to the shifted value fairly quickly and ensure that the outputs stay at the target. This subsection concludes that EWMA-R2R control using dynamic sampling provides superior results as compared to EWMA-R2R control based on uniform sampling and random sampling for a typical semiconductor process represented by the MIMO model simulated in this study. EWMA-R2R control using dynamic sampling is explored in more detail in the subsequent subsections.

Dynamic sampling with Bayesian detection approach has been studied extensively by Lee [70] with successful implementation on both simulated and industrial datasets. However, not much focus was put on the following two aspects of dynamic sampling in the study. First, the effect of  $\lambda$  on the performance of EWMA-R2R controller was not studied. We saw in Section 6.5.2 that EWMA forgetting factor  $\lambda$  has a significant effect on the performance of EWMA-R2R controller. In the next subsection, this effect will be exploited to arrive at a novel way of implementing EWMA-R2R control using dynamic sampling. Second, Lee suggested that when the process is in control, the sampling is done according to a baseline sampling rate that is determined by the economics of the process. We will show in Sections 6.5.5 and 6.5.6 that during the normal operation, the sampling rate can be further reduced to a value lower than the baseline sampling rate without compromising the control performance. When the process is in control, we can easily rely on the estimates made by the VM model instead of making physical measurements according to the baseline sampling rate.

#### 6.5.4 Improvement in EWMA-R2R control performance using Bayesian update of EWMA forgetting factor

We discussed the effect of EWMA forgetting factor  $\lambda$  on the performance of EWMA-R2R controller in Section 6.5.2. A small value of  $\lambda$  updates the estimate of the offset term sluggishly on the occurrence of a process shift. This can be observed in Figure 6.7, where the estimates adapt to the shifted value of the offset term slowly. This sluggish behavior further propagates into the inputs calculated by the controller and the controlled outputs. On the other hand, a large value of  $\lambda$  provides quick adaptation to the shifted value of the offset term, but is not able to filter out the measurement noise appropriately. As a result, the estimates have more fluctuations as compared to those obtained using smaller values of  $\lambda$ . Thus, a small value of  $\lambda$  is desired when the process is operating normally and infected with measurement noise only. The small value of  $\lambda$  will ensure that noise is filtered out and the estimates do not show much fluctuation. Also, a large value of  $\lambda$  is desired when the process undergoes a shift in order to quickly adapt to the shifted value of the offset term.

These desired characteristics of EWMA forgetting factor  $\lambda$  match those of posterior probability  $P(\Theta_D|X_k)$  very well.  $P(\Theta_D|X_k)$  represents the probability that the current observation is coming from a shifted process. When the process is operated normally infected with measurement noise only, this probability is low, and when a process shift occurs, this probability increases and suggests a decrease in the size of sampling interval. Another similarity is

that the values of both  $\lambda$  and  $P(\Theta_D|X_k)$  are bounded by the closed interval  $[0,1]$ . In previous sections, Bayesian detection approach assisted dynamic sampling by letting the user know when to change the sampling frequency. In this section, we will show that even better control performance can be achieved by updating  $\lambda$  using Bayesian detection approach. EWMA with  $\lambda$  updated using Bayesian detection approach will be referred to as B-EWMA in this dissertation. Equations 6.27 and 6.28 show how EWMA forgetting factor  $\lambda$  can be related to the posterior probabilities calculated from Bayesian detection approach. In Equation 6.28,  $P(\Theta_N|X_k)$  represents the probability that the current observation is coming from a normal process (no shift).

$$\lambda = P(\Theta_D|X_k) \tag{6.27}$$

$$1 - \lambda = 1 - P(\Theta_D|X_k) = P(\Theta_N|X_k) \tag{6.28}$$

Owing to the reasons discussed above, B-EWMA-R2R control using dynamic sampling results in better control performance as compared to EWMA-R2R control using uniform sampling, random sampling, and dynamic sampling for both SISO and MIMO model. The control results for the SISO and MIMO processes studied in this work are provided in Tables 6.3 and 6.4, respectively. The significant improvement by employing B-EWMA-R2R control is evident as MSE values are roughly 30 percent less than those obtained by employing EWMA-R2R control. Moreover, better control results lead to reduced number of measured wafers. Figures 6.12 and 6.13 show the plots for the controlled outputs using different control schemes for the SISO and MIMO model,

respectively. It is clear from the plots that B-EWMA-R2R control using dynamic sampling is the quickest to adapt to the process shift as compared to other control schemes and provides the outputs that are closest to the target. In essence, better control performance with reduced metrology costs can be achieved by using B-EWMA-R2R control.

Table 6.3: Controller performance for SISO model using different sampling methods

Control scheme	MSE	Number of wafers measured
Uncontrolled output	2.3055	0
EWMA-R2R with uniform sampling	0.5768	83
EWMA-R2R with random sampling	0.6650	81
EWMA-R2R with dynamic sampling	0.4767	97
B-EWMA-R2R with dynamic sampling	0.2664	91

Table 6.4: Controller performance for MIMO model using different sampling methods

Control scheme	MSE ( $y_1$ )	MSE ( $y_2$ )	Number of wafers measured
Uncontrolled output	2.4347	0.3377	0
EWMA-R2R with uniform sampling	0.6494	0.0954	83
EWMA-R2R with random sampling	0.7264	0.1152	84
EWMA-R2R with dynamic sampling	0.5433	0.0843	99
B-EWMA-R2R with dynamic sampling	0.3838	0.0570	91

### 6.5.5 VM-assisted EWMA-R2R control

The sampling methods discussed so far in this chapter, uniform sampling, random sampling, and dynamic sampling, update the estimates of the offset term and the recipe settings (inputs) of the process only when a physical measurement was made as dictated by the sampling plan. Using the predictions made by VM model, it is possible to make these updates even when a

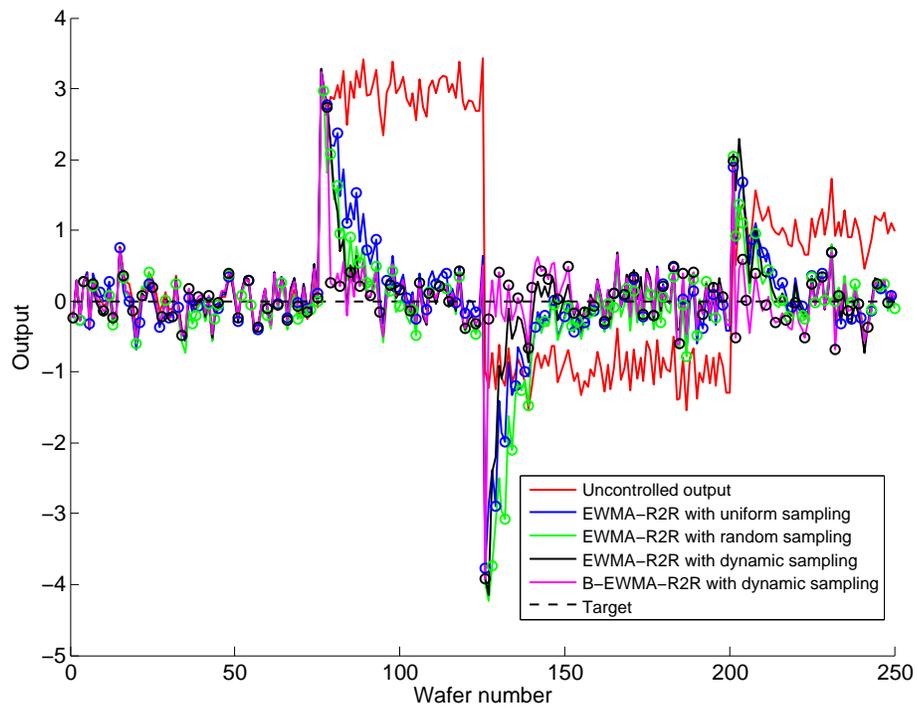


Figure 6.12: Comparison of different control schemes for SISO model. B-EWMA-R2R control using dynamic sampling provides the best control performance

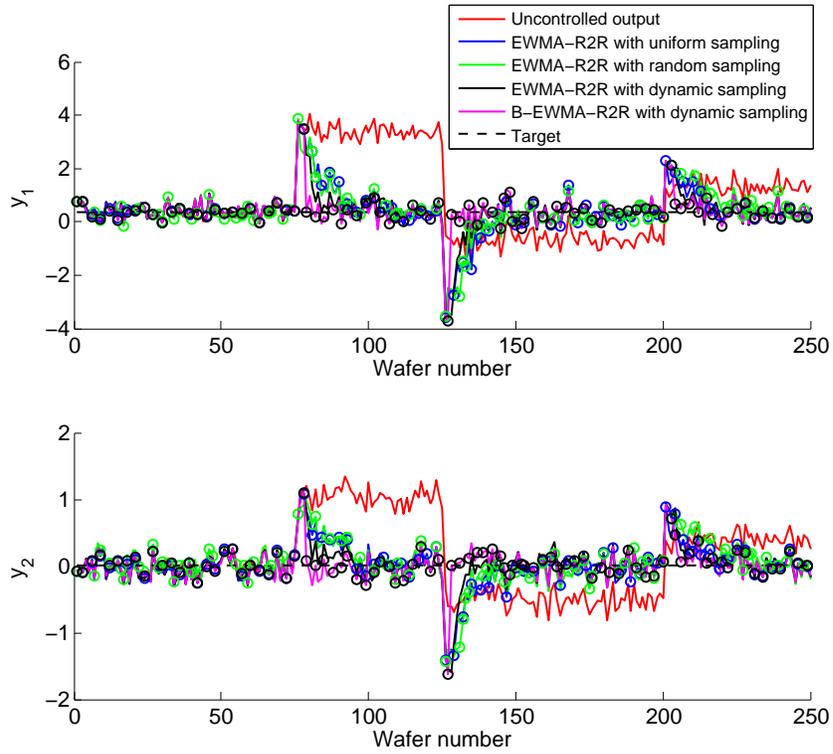


Figure 6.13: Comparison of different control schemes for MIMO model. B-EWMA-R2R control using dynamic sampling provides the best control performance

physical measurement is not done. VM enables us to update the estimate of offset term and the recipe settings of the process after processing each product wafer irrespective of the fact whether the wafer was physically measured or not. When the process is operating normally, the sampling rate can be reduced to a value below the baseline sampling rate determined by process economics. An accurate VM model will ensure reduced measurement costs and better controller performance. EWMA-R2R control that updates the estimates of the offset term and the recipe settings of the process using the outputs predicted by VM model will be referred to as VM-assisted EWMA-R2R control. The working of VM-assisted EWMA-R2R control is shown in Figure 6.14.

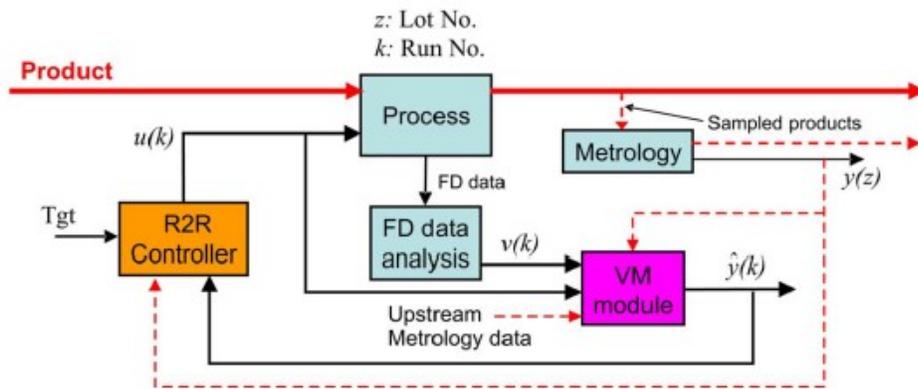


Figure 6.14: The working of VM-assisted EWMA-R2R control

Khan et al. [62, 63] have shown that VM has the potential to improve the performance of R2R controllers. The results obtained using VM to predict the outputs were much closer to target than those obtained without using VM. However, their approach suffers from the two limitations mentioned be-

low. First, a PLS model was built to predict the process outputs using process variables and process drift. The PLS model explicitly included the process drift as one of the input parameters, which makes it easier to quantify the relationship between the process drift and the outputs. In this work, this will be addressed more appropriately by building a PLS model that predicts process outputs using process variables only. Using process variables for predicting process outputs is a more practical approach as the values of process variables are always known for a given process whereas we might not have much knowledge about the unexpected process drifts that might happen during processing. The effect of process drift is usually apparent in the process variables. In the presence of process drift, a good VM model should be capable of predicting process outputs fairly accurately using process variables only without considering the process drift explicitly as an input parameter.

Second, Khan et al. assumed that the physical measurements were made after a fixed number of process runs. VM was employed to predict the outputs for the process runs between two consecutive physical measurements. Therefore, all the simulations in their study were based on uniform sampling. Also, no evaluation of the accuracy of the VM estimate was provided. If the VM model can predict the outputs with good accuracy, the physical measurements need not be made. Making measurements according to uniform sampling might lead to wasting of resources as the VM model can provide very similar estimates at no cost. In this work, we will be calculating a reliance index for the estimates made by VM. Whenever the value of the calculated

reliance index falls below a certain threshold value, a physical measurement will be made.

Some confidence-interval based methods have been proposed in the related literature in the past. Chryssolouris et al. [18] presented a method that finds confidence-intervals for neural-network prediction models. Later, Rivals and Personnaz [107] presented an approach to construct confidence intervals for neural networks based on least squares estimation. These confidence-interval based approaches provide a range in which the predicted outputs might fall based on the historical data, which is conceptually different from calculating a reliance index of the predicted values. Therefore, confidence-interval based approaches cannot be used to determine the degree to which we can rely on the VM estimates.

More recently, the concept of performance confidence value (CV) for assessing performance degradation using a watchdog prognostics agent was presented by some researchers [21, 141]. The proposed assessment only calculates a numerical performance CV but does not set up a proper threshold value to determine whether the performance CV is reliable or not. To our best knowledge, the only significant study done on defining a reliance index in the context of VM has been done by Cheng et al. [16]. They assumed that the actual process output values can be obtained by feeding process input information into a reference prediction model. The predictions made by VM model are then compared with the predictions made by the reference model to calculate the reliance index. This method cannot be applied practically as

the actual output values are not known unless we measure them. Also, if a model that can predict the outputs very accurately exists, setting the outputs predicted by that model as the reference to calculate the reliance level of VM estimates does not seem reasonable. One may just use that reference model to obtain better VM estimates instead. The authors assumed Multiple Regression (MR) model as the reference model, and Neural Network (NN) model as the VM model. In other words, they assumed that MR model (a linear model) can provide more accurate results than NN model (nonlinear model). This assumption contradicts the fact that a nonlinear model allows for more degrees of freedom and provides better modeling results than a linear model.

Due to the limitations stated above, we can conclude that no established and reliable approach exists for quantifying the reliance level of VM estimates. In the next section, we will present a new approach to calculate the reliance level of VM estimates that addresses the shortcomings of the previous approaches.

#### **6.5.6 New approach for calculating reliance index of VM estimates**

After processing each wafer, a decision whether the most recent wafer should be measured or not needs to be made. This can be decided by calculating a reliance index that quantifies how much a manufacturer can rely on the VM estimate. If the value of calculated reliance index is below a certain threshold, a physical measurement needs to be made as the manufacturer cannot rely on the VM estimate. Some of the previous work done on reliance index

is summarized in Section 6.5.5. Due to the limitations of these approaches, an opportunity exists to devise more practical and reliable approaches. This section proposes one such approach that calculates reliance index of VM estimates in a practically applicable way.

Suppose we have some output measurements available from the historical data. Assuming normal distribution with mean and standard deviation calculated from the historical data, we can calculate the probability of observing a particular value of output measurement. In this work, this distribution based on the historical measurements will be employed as reference distribution. A moving-window approach will be adopted to update the historical dataset. Whenever a new measurement is made, the oldest entry in the historical dataset will be replaced by the new measurement. The idea behind this replacement is to maintain a historical dataset that represents the current behavior of the process.

The predictions made by the VM model are evaluated against the reference distribution to assess their quality. In order to consider the error that might be present in the VM prediction, we propose to predict a distribution of outputs instead of a fixed output value for a given set of process variables. The value predicted by the VM model will serve as mean for the distribution of outputs. The standard deviation is calculated as the standard deviation of the predicted values for the historical dataset. Using these mean and standard deviation values, a normal distribution of outputs predicted by the VM model can be built. This distribution will be referred to as VM distribution in the

subsequent sections.

Whenever new process data are available, the VM model will provide an output value with a reliance index calculated using the reference distribution and the VM distribution. Specifically, the reliance index is calculated as the overlapping area between the reference distribution and the VM distribution as shown in Figure 6.15. If these two distributions are very similar, the reliance index would lie near one; if the two distributions are very different from each other, the value of reliance index would be near zero. These two distributions will have only one intersection point if they have different mean values but same standard deviation, where as two intersection points if standard deviations are different. Mathematically, the overlapping area can be calculated using an analytical expression provided in Equation 6.29 or a numerical expression provided in Equation 6.30. In these equations,  $f_{min}(s)$  corresponds to the smaller one of the reference distribution function and VM distribution function given by Equations 6.31 and 6.32, respectively. In Equation 6.30,  $\Delta x$  is the interval for numerical integration and be set as a small number by the user.  $n$  is the number of such intervals that engulf all the overlapping area.

$$RI = \int_{-\infty}^{\infty} f_{min}(x) dx \quad (6.29)$$

$$RI = \sum_{i=1}^n f_{min}(x_i) \Delta x \quad (6.30)$$

$$f_{ref}(x) = \frac{1}{\sqrt{2\pi}\sigma_{ref}} e^{-\frac{(x-\mu_{ref})^2}{2\sigma_{ref}^2}} \quad (6.31)$$

$$f_{vm}(x) = \frac{1}{\sqrt{2\pi}\sigma_{vm}} e^{-\frac{(x-\mu_{vm})^2}{2\sigma_{vm}^2}} \quad (6.32)$$

$$f_{min}(x) = \min(f_{ref}(x), f_{vm}(x)) \quad (6.33)$$

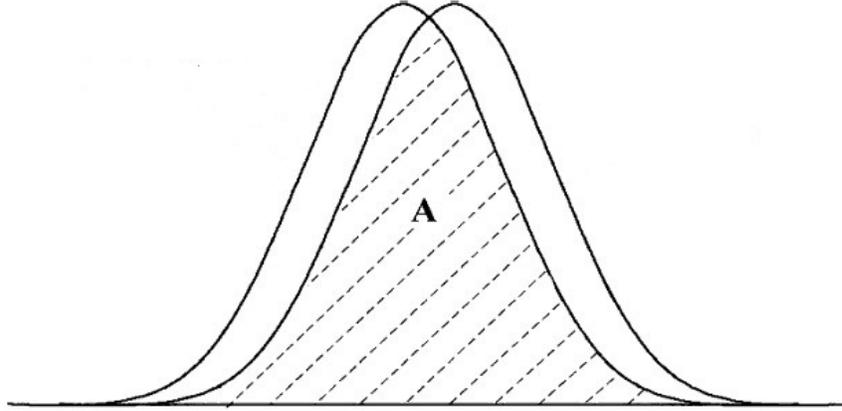


Figure 6.15: Reliance index is calculated as the overlapping area between the reference distribution and the VM distribution

In this study, a recursive PLS model is adopted as VM model to predict the outputs. As the SISO model discussed in Section 6.4.1 has only one input and one output, building a recursive PLS model for it to show the superior performance of VM-assisted EWMA-R2R control might not seem appealing. Instead, a recursive PLS model based on process variables presented in Section 6.4.2 is employed as VM model. After predicting the value of two outputs,  $y_1$  and  $y_2$ , reliance indices were calculated for each of them. If one or both of them had a value less than the threshold value of 0.7, the estimates were declared to be unreliable and actual measurements were made. These actual measurements were sent to the R2R controller to update the estimates of the offset term and the process inputs. Otherwise, the estimates were considered

reliable and passed on to the R2R controller to update the estimates of the offset term and process inputs. Therefore, the estimates of the offset term and process inputs are updated after processing each wafer, either using actual measurements or the VM estimates. It can be recalled that for the control schemes discussed earlier in this chapter, the updates were only done when actual measurements were performed. This is the fundamental reason behind the better control performance of VM-assisted EWMA-R2R control as compared to regular EWMA-R2R control.

Figure 6.16 shows the control performance of VM-assisted EWMA-R2R control along with other control schemes discussed earlier in this chapter. It should be noted that the results of EWMA-R2R control using uniform sampling and random sampling are not shown in this figure so that the controlled output trajectories for different control schemes are clearly visible. However, the MSEs of VM-assisted EWMA-R2R control are provided along with those of all the control schemes discussed earlier in Table 6.5. We can see in Figure 6.16 that when a process shift occurs, VM-assisted EWMA-R2R control is the first method to recognize the process shift and quickly adapts to the shift to bring the outputs back to target. This is because of the fact that VM makes it possible to update the EWMA-R2R controller at the end of processing of each wafer. For this particular study, the first process shift occurs after processing wafer number 75. Both EWMA-R2R with dynamic sampling and B-EWMA-R2R with dynamic sampling make measurements according to baseline uniform sampling rate of 3 when the process is operating normally.

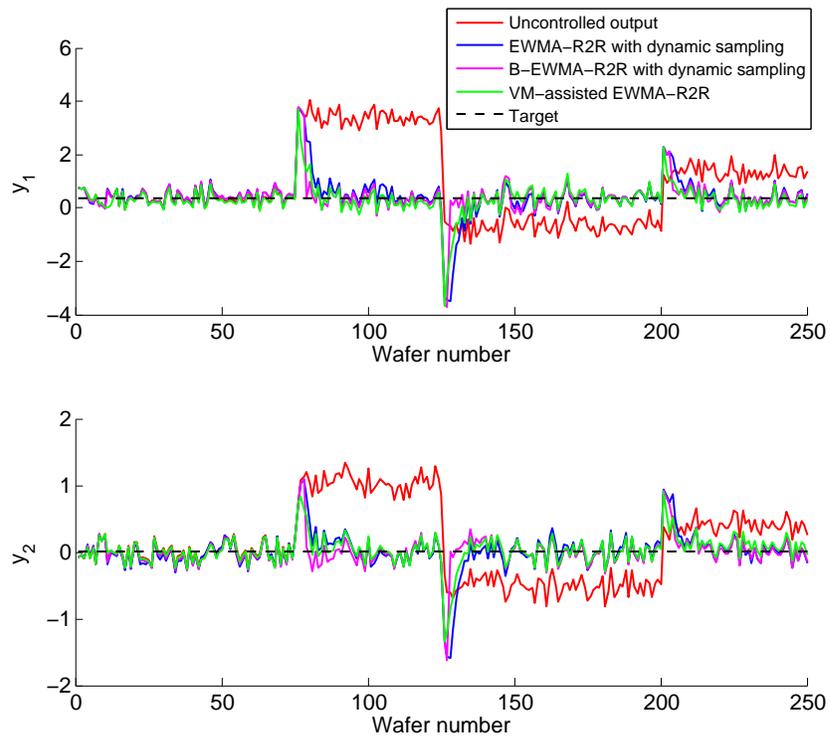


Figure 6.16: Simulation results showing the superior performance of VM-assisted EWMA-R2R control as compared to other control schemes.

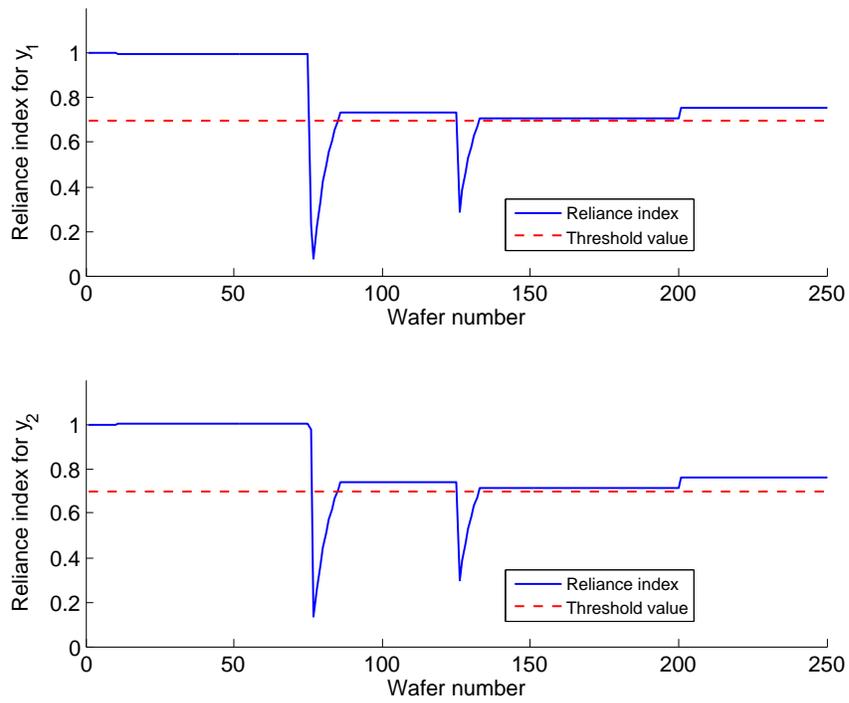


Figure 6.17: Reliance indices for the VM estimates of two process outputs. A threshold value of 0.7 was used in this simulation.

So, when a process shift occurs after processing of wafer number 75, it is quite possible that the next wafer that is measured according to baseline sampling rate is wafer number 77 or wafer number 78. This late detection leads to the sluggish adaptation of the EWMA-R2R controller to the process shift and gives rise to high MSE values. If there is a metrology delay, the situation becomes worse and can lead to even more sluggish adaptation of the controller. On the other hand, VM-assisted EWMA-R2R control will provide greater benefits in the case of a metrology delay by updating the controller after processing each wafer.

Another advantage of using VM-assisted EWMA-R2R control is the significant reduction in the number of physical measurements of wafers. During normal operation, VM predicts the process outputs fairly accurately. The reliance index of the VM estimates is monitored continuously during the processing of wafers. Whenever the value of reliance index falls below a set threshold value, actual measurements are done until the value of reliance index becomes greater than the threshold value as shown in Figure 6.17. So, most of the measurements are only done after the process shift and very few measurements are done during the normal operation. Table 6.5 shows that VM-assisted EWMA-R2R control leads to a drastic reduction in the number of required physical measurements. Similar improvements in the controller performance and the reduction in the number of measured wafers were observed by using different sets of parameters for the process model (see Section 6.4.2) simulated in this work. In essence, this study clearly demonstrates that VM has the potential

to reduce the measurement costs significantly while promising better process control.

Table 6.5: Controller performance for MIMO model using different sampling methods

Control scheme	MSE ( $y_1$ )	MSE ( $y_2$ )	Number of wafers measured
Uncontrolled output	2.4347	0.3377	0
EWMA-R2R with uniform sampling	0.6494	0.0954	83
EWMA-R2R with random sampling	0.7264	0.1152	84
EWMA-R2R with dynamic sampling	0.5433	0.0843	99
B-EWMA-R2R with dynamic sampling	0.3838	0.0570	91
VM-assisted EWMA-R2R	0.3167	0.0504	17

## 6.6 Conclusions

This chapter showed how to combine physical measurements with the VM estimates to develop a more robust approach than using VM alone. Instead of blindly relying on the estimates made by VM, the combined approach aims at monitoring the quality of VM estimates and performs a physical measurement whenever the quality of VM estimates falls below a threshold value. More metrology events increase the measurement costs and decrease the product throughput (by increasing cycle time), whereas too few metrology events might hamper the product quality. Therefore, the frequency of metrology events needs to be optimized. Thus, the implementation of the combined approach requires the development of optimal sampling plans that will tell the semiconductor manufacturers when to perform a physical measurement to supplement VM predictions.

In this chapter, first we simulated a Single-Input-Single-Output (SISO)

process with process drift and noise. Run-to-Run (R2R) control was employed to adjust the recipe settings (inputs) to ensure that the output stays on the target in the presence of process drift and noise. In general, the implementation of R2R control includes the estimation of process gain matrix, process drift, or both. In semiconductor manufacturing, process drift is a major issue of concern as process gain matrix remains almost constant owing to the physics and chemistry behind the process. So, in this work the process drift was estimated using the measurements done according to the sampling plan. Whenever a measurement was made, the value of process drift was estimated by Exponentially-Weighted-Moving-Average (EWMA), a weighted average of the previous estimate of the process drift and the process drift value suggested by the current measurement.

Devising an optimal sampling plan is critical in order to ensure that the process outputs are on target, while not spending a large amount of money by measuring too many products. We implemented some well-known sampling methods in order to demonstrate the superior performance of reliance index based sampling method that utilizes VM estimates. The most common sampling strategy is uniform sampling, which measures a product after a fixed interval of time or products. Random sampling does not have a fixed measurement interval, but measures the products at random intervals that have specified lower and upper limits. Neither of these methods take advantage of the known past and current behavior of the process. Dynamic sampling is based on the intuitive idea of measuring more products when the process drifts

away from the target and measuring fewer products when the process outputs are fairly close to the target. In this work, Bayesian detection approach was employed to implement dynamic sampling. The Bayesian detection approach calculates a posterior probability distribution using a prior probability distribution and the observed data. When the probability that the currently observed data is coming from a drifting process exceeds a threshold value, the sampling frequency is increased. Better control results were observed for the SISO process when the sampling plan was driven by dynamic sampling as compared to uniform sampling and random sampling (see Section 6.5.1 for details). For a Multiple-Input-Multiple-Output (MIMO) model present in the VM literature (see Section 6.4.2), the R2R results were closer to the target when dynamic sampling was employed as compared to the R2R results when uniform sampling and random sampling were employed (see Section 6.5.3 for details).

The R2R control results were found to be sensitive to the EWMA forgetting factor,  $\lambda$ , which determines the weighting of the previous estimate of the process drift vs. the weighting of the value of process drift suggested by the recent measurement. A smaller value of  $\lambda$  filters the noise to a higher degree, but causes the outputs to adapt to the process drift sluggishly. On the other hand, a larger value of  $\lambda$  drives the outputs to adapt to the process drift quickly, but filters the noise to a smaller extent. This trade-off can be exploited to meet the needs of the R2R controller implemented in this work. Most of the literature about R2R control for semiconductor manufacturing assumes a

constant value of  $\lambda$  [70, 98]. However, this work introduced a novel way to update  $\lambda$  in order to achieve an improved controller performance. For the best performance, the controller must be able to reject most of the noise present in the measurements and adapt to the process drift as quickly as possible. The R2R controller with Bayesian update of  $\lambda$  showed better control performance than the conventional R2R controller with a fixed value of  $\lambda$  (see Section 6.5.4).

In the three sampling methods discussed above, uniform sampling, random sampling, and dynamic sampling, the estimates of the process drift and the recipe settings (inputs) of the process were only updated when a physical measurement was made as dictated by the sampling plan. Using the predictions made by VM model, it is possible to make these updates even when a physical measurement is not done. VM enables us to update the estimate of process drift and the recipe settings of the process after processing each product wafer irrespective of the fact whether the wafer was physically measured or not. An accurate VM model will ensure reduced measurement costs and better controller performance. After processing each wafer, a decision whether the most recent wafer should be measured or not needs to be made. This can be decided by calculating a reliance index that quantifies how much a manufacturer can rely on the VM estimate. If the value of calculated reliance index is below a certain threshold, a physical measurement needs to be made as the manufacturer cannot rely on the VM estimate. Some work on a reliance index is present in VM literature [16] but it suffers from a few shortcomings (see Section 6.5.5 for details). A new reliance index, which is more attractive from a

mathematical and practical point of view, is proposed in this work. The R2R controller results obtained by utilizing predictions made by VM were found to be better than those obtained by employing the three sampling methods mentioned above. Moreover, the number of physical measurements was also drastically reduced by implementing VM-assisted R2R control. The simulation results clearly demonstrate that VM has the potential to reduce measurement costs significantly while promising better process control.

# Chapter 7

## Summary and Future Work

### 7.1 Summary of Contributions

In Chapter 2, various VM methods were introduced and compared in terms of prediction accuracy using four industrial datasets collected from a plasma etch system at Texas Instruments, Inc.. Specifically, multiple linear regression (MLR), principal component regression (PCR), partial least squares regression (PLSR), recursive partial least squares regression (R-PLSR), time series analysis, and Kalman filter estimation were implemented to predict process outputs such as etch rate, sheet resistance, and critical dimension (CD). Kalman filter estimation was employed in a novel way to serve as a VM model for predicting outputs of a static process.

First, lot-level predictions were made for etch rate using 18 optical emission spectroscopy (OES) signals for the first three datasets. Recursive PLS regression (R-PLSR) and Kalman filter showed the best prediction results as they update the model whenever new measurements are available. However, the correlation between the OES signals and etch rate was not found to be very strong because only one value of measured etch rate was available per lot. Next, to obtain better correlation between the input and the output

variables, a quality variable that was measured for each wafer was identified. Sheet resistance data were collected for 1121 wafers and correlated with OES data using various VM methods mentioned above. Recursive PLS regression and Kalman filter showed the best wafer-level predictions for sheet resistance using the OES data. It was observed that the OES data have much better correlation with the sheet resistance data as compared to the etch rate data. Sheet resistance was observed to be a strong function of the OES signals that represent the optical emissions from the gases present in the etch recipe. In other words, the modeling results were found to be in agreement with the process chemistry. Last, Dataset 4 was collected from a gate etch process to figure out the reason behind the non-uniformity in the etch CDs of wafers. The model predictions were found to be fairly good with a MAPE value of 1.5159 and a  $R^2$  value of 0.4324. Nine process variables that had the most significant effect on the CD were identified. Most likely, these process variables were responsible for causing the undesired CD values for the first two wafers in the lots under consideration.

In Chapter 3, two PLS variants (PLS with EWMA mean update and recursive PLS) were proposed as robust VM algorithms that can predict process outputs fairly well in the presence of unexpected process drifts and noise. Three types of process drifts were simulated and it was found that recursive PLS and PLS with EWMA mean update provided better predictions than traditional PLS algorithm for all drift types; recursive PLS being the best prediction method. However, in the presence of large measurement noise, PLS

with EWMA mean update provided the best predictions as it is more conservative than recursive PLS in adapting to new measurements. These general guidelines reinforce VM technology by suggesting appropriate prediction methods when unexpected process changes occur. Other modeling features such as the selection of model inputs, tuning of the EWMA factor  $\lambda$ , and design of experiments were also discussed.

For a successful implementation of virtual metrology (VM), we need to make sure that the data entering the VM model are free from faults. Sensor faults are the most relevant faults in the context of VM as VM relies on the sensor data to predict the process outputs. The objective of Chapters 4 and 5 was to remove the effect of sensor faults from the sensor data and feed the corrected (reconstructed) sensor data to the VM model. To achieve this objective, three steps (fault detection, fault identification, and fault reconstruction) were performed. In order to compare the performance of various fault detection and identification methods, a benchmark dataset was utilized.

The first step to achieve the goal of removal of the effect of fault from the faulty sensor data is fault detection. First, we presented fault detection using principal component analysis (PCA). PCA is nominally able to detect faults for a two-dimensional data matrix only, the two dimensions being time and process variables in most cases. However, the data collected from a semiconductor manufacturing process are three-dimensional, with an additional dimension for different wafers. Instead, multiway principal component analysis (MPCA) is employed to address this limitation of the standard PCA implementation.

MPCA was implemented to detect the artificial faults induced in the benchmark dataset. The effect of the fault magnitude and the confidence level ( $\alpha$ ) on the fault detection performance of MPCA was also studied. It was found that MPCA raised several false alarms (i.e., MPCA indicated the presence of a fault for the fault-free data). Next, we presented a variation of MPCA that reduces false alarms by EWMA filtering of the residuals, and a statistics pattern analysis (SPA) based method, which performs PCA on the statistics of the process variables vs. the temporal values of the process variables. It was observed that the MPCA-based methods were able to detect more mean and variance faults than the skewness and kurtosis faults. This is due to the fact that MPCA-based methods are second-order methods which only consider mean and variance of the data. The number of detected mean and variance faults by using both the MPCA-based methods were less than those detected by SPA. This is due to the non-Gaussian characteristics of the data collected from semiconductor manufacturing processes. MPCA-based methods wrongly assume the data to be Gaussian and calculate erroneous control limits for the fault detection indices. MPCA with EWMA filtering of residuals detected fewer faults than MPCA for all the fault types. Due to the filtering of the residuals, the effect of the faults appears slowly in the filtered residuals. MPCA with EWMA filtering of residuals provided better detection when a small value of the EWMA forgetting factor ( $\Gamma$ ) was used, which favors the detection of mean faults. So, it detected more mean faults than the variance faults. The only advantage of using EWMA filtering is that

it leads to fewer false alarms than MPCA.

The fault detection study concluded that SPA provides better fault detection performance for different types of faults as compared to MPCA-based methods. Not only it detects more faults, SPA also significantly reduces the number of false alarms. Therefore, this study recommends that SPA should be employed as a fault detection method to detect faults in the VM sensors.

The second step to achieve our goal is to perform fault identification, which aims at finding the process variable/process variables which caused a fault in a wafer/batch. Apart from the fact that SPA detected more faults than the two MPCA-based methods, the main benefit of performing identification on the faults detected by SPA is that not only the faulty sensor can be identified, but the statistic (e.g., mean, variance) in which the fault occurred is also identified. This information is crucial for predicting accurate outputs using virtual metrology models, which mostly use the statistics of the sensor signals as the inputs. Therefore, fault identification was performed on the faults detected by SPA in this study.

We presented and implemented three well-known fault identification methods present in literature. Specifically, these included contribution plot approach, reconstruction-based contribution (RBC) approach, and sensor validity index (SVI) approach. An equation that relates the RBC with the SVI was derived. The RBC method and the SVI method exhibited similar identification results. This is due to the fact that both these methods are based on the same idea of reconstruction of faults. The contribution plot method identified

a smaller number of faults correctly as compared to the RBC and the SVI methods because of its limitations. Superior identification performance of the RBC and the SVI methods as compared to the contribution plot approach was shown in this work. Hence, we recommend the use of the RBC and the SVI methods to perform the identification/diagnosis of faults in virtual metrology sensors.

After identifying the sensor that caused the fault, the third step to achieve our goal is to perform fault reconstruction, which includes the estimation of the size of the fault. The estimation of fault magnitude is very important in order to make accurate predictions using a virtual metrology model. To obtain the fault-free sensor signals that are fed as inputs to the virtual metrology model, the effect of the fault needs to be removed from the faulty sensor signal with high precision. Fault reconstruction is mainly done in three ways: reconstruction via iteration, the missing value approach, and reconstruction via optimization. Qin et al. [104] have shown that all three ways of doing reconstruction mentioned above lead to essentially the same results. The magnitude of the fault was estimated by minimizing the fault detection indices, SPE,  $T^2$ , or  $\phi$ . Fairly good estimates of the fault magnitude were obtained when the faults were identified correctly. In the case of incorrect identification, the fault direction  $\xi_i$  was not known correctly and led to an erroneous estimation of the fault magnitude.

Chapter 6 showed how to combine physical measurements with the VM estimates to develop a more robust approach than using VM alone. Instead of

blindly relying on the estimates made by VM, the combined approach aims at monitoring the quality of VM estimates and performs a physical measurement whenever the quality of VM estimates falls below a threshold value. More metrology events increase the measurement costs and decrease the product throughput (by increasing cycle time), whereas too few metrology events might hamper the product quality. Therefore, the frequency of metrology events needs to be optimized. The implementation of the combined approach requires the development of optimal sampling plans that will tell the semiconductor manufacturers when to perform a physical measurement to supplement VM predictions.

In Chapter 6, first we simulated a Single-Input-Single-Output (SISO) process with process drift and noise. Run-to-Run (R2R) control was employed to adjust the recipe settings (inputs) to ensure that the output stays on the target in the presence of process drift and noise. In general, the implementation of R2R control includes the estimation of process gain matrix, process drift, or both. In semiconductor manufacturing, process drift is a major issue of concern. Because the process gain matrix remains almost constant owing to the physics and chemistry behind the process, the process drift can be estimated using the measurements done according to the sampling plan. Whenever a measurement was made, the value of process drift was estimated by Exponentially-Weighted-Moving-Average (EWMA), a weighted average of the previous estimate of the process drift and the process drift value suggested by the current measurement.

Devising an optimal sampling plan is critical in order to ensure that the process outputs are on target, while not spending a large amount of money by measuring too many products. We implemented some well-known sampling methods in order to demonstrate the superior performance of reliance index based sampling method that utilizes VM estimates. The most common sampling strategy is uniform sampling, which measures a product after a fixed interval of time or products. Random sampling does not have a fixed measurement interval, but measures the products at random intervals that have specified lower and upper limits. Neither of these methods take advantage of the known past and current behavior of the process. Dynamic sampling is based on the intuitive idea of measuring more products when the process drifts away from the target and measuring fewer products when the process outputs are fairly close to the target. In this work, Bayesian detection approach was employed to implement dynamic sampling. Better control results were observed for the SISO process when the sampling plan was driven by dynamic sampling as compared to uniform sampling and random sampling. For a Multiple-Input-Multiple-Output (MIMO) model present in the VM literature, the R2R results were closer to the target when dynamic sampling was employed as compared to the R2R results when uniform sampling and random sampling were employed.

The R2R control results were found to be sensitive to the EWMA forgetting factor,  $\lambda$ , which determines the weighting of the previous estimate of the process drift vs. the weighting of the value of process drift suggested

by the recent measurement. A smaller value of  $\lambda$  filters the noise to a higher degree, but causes the outputs to adapt to the process drift sluggishly. On the other hand, a larger value of  $\lambda$  drives the outputs to adapt to the process drift quickly, but filters the noise to a smaller extent. This trade-off can be exploited to meet the needs of the R2R controller implemented in this work. Most of the literature about R2R control for semiconductor manufacturing assumes a constant value of  $\lambda$  [70, 98]. However, this work introduced a novel way to update  $\lambda$  in order to achieve an improved controller performance. For the best performance, the controller must be able to reject most of the noise present in the measurements and adapt to the process drift as quickly as possible. The R2R controller with Bayesian update of  $\lambda$  showed better control performance than the conventional R2R controller with a fixed value of  $\lambda$ .

In the three sampling methods discussed above, uniform sampling, random sampling, and dynamic sampling, the estimates of the process drift and the recipe settings (inputs) of the process were only updated when a physical measurement was made as dictated by the sampling plan. Using the predictions made by VM model, it is possible to make these updates even when a physical measurement is not done. VM enables updating the estimate of process drift and the recipe settings of the process after processing each product wafer irrespective of the fact whether the wafer was physically measured or not. An accurate VM model will ensure reduced measurement costs and better controller performance. After processing each wafer, a decision whether the most recent wafer should be measured or not needs to be made. This

can be decided by calculating a reliance index that quantifies how much a manufacturer can rely on the VM estimate. If the value of calculated reliance index is below a certain threshold, a physical measurement needs to be made as the manufacturer cannot rely on the VM estimate. Some work on a reliance index is present in VM literature [16] but it suffers from a few shortcomings. A new reliance index, which is more attractive from a mathematical and practical point of view, is proposed in this work. The R2R controller results obtained by utilizing predictions made by VM were found to be better than those obtained by employing the three sampling methods mentioned above. Moreover, the number of physical measurements was also drastically reduced by implementing VM-assisted R2R control. The simulation results clearly demonstrate that VM has the potential to reduce measurement costs significantly while promising better process control.

## 7.2 Recommendations for Future Work

Although several regression and estimation methods were employed to build the VM model in this work, an opportunity exists to explore more complex nonlinear techniques like neural networks. A more comprehensive comparison of the modeling methods is required before implementing VM on industrial scale. While estimating the states using a Kalman filter, it is assumed that the system noise and measurement noise are Gaussian white noise terms. In other words, they have Gaussian (Normal) distribution with mean zero and standard deviation  $\sigma$ . There is a possibility that the system is infected with

a non-Gaussian noise (having a distribution other than Gaussian). Kalman filtering of colored (non-white) noise has proven useful in image restoration [139] and speech enhancement [36]. Also, Independent Component Analysis (ICA) [49, 50, 72, 113, 138, 140] can be applied for estimation in the presence of non-Gaussian noise.

Deciding the threshold value for the acceptable coefficient of determination ( $R^2$ ) is also an very important issue and needs to be addressed for the context of VM [111]. In Chapter 3, two PLS variants (PLS with EWMA mean update and recursive PLS) were proposed as robust VM algorithms that can predict process outputs fairly accurately in the presence of unexpected process drifts and noise. However, the data were generated from a model present in VM literature. The results obtained in Chapter 3 can be validated by comparing the PLS variants for industrial data with process drifts and noise.

While implementing statistics pattern analysis (SPA) for fault detection in Chapter 5, the statistics pattern was made up of four batch statistics: mean, variance, skewness, and kurtosis. More research is required to choose these statistics so that all the critical features of the process are captured. Specifically, an index that will quantify the degree of information captured by batch statistics for a process needs to be defined. For similarity quantification, both distance-based and angle-based similarities need to be compared to gain better understanding on their capabilities. New similarity indices can be developed to obtain better fault detection and identification performance.

In the contribution plot approach used for fault identification, the con-

tributions are spread unevenly across the variables when there is no fault. In other words, some variables have large contributions while others have relatively smaller contributions for the normal (fault-free) data. Therefore, a fault in a normally small-contribution variable may not make the contribution of that variable the largest unless the fault magnitude is very large. This is a common cause of misidentification while using contribution plots. An effort can be made to improve the contribution plot approach by calculating separate control limits for each variable. While identifying a fault, the contribution of a variable will be compared with its control limit and not with the contributions of other variables as done in current contribution plot approach. This modification might allow correct identification when a fault occurs in a normally small-contribution variable.

Three identification methods including contribution plot approach, SVI approach, and RBC approach were compared in Chapter 5. Other identification methods such as (a) discrimination by angles [106, 143]; (b) pattern matching methods by calculating similarity and dissimilarity factors between normal data and an extended period of fault data [60, 114, 115]; and (c) isolation enhanced techniques from model-based methods [34, 35, 101, 102] can be simulated to assess their performance for faults in semiconductor manufacturing processes.

Chapters 4 and 5 focused on the removal of the effect of sensor faults from the sensor data and feeding the corrected (reconstructed) sensor data to the VM model. The benefit of correct detection, identification, and reconstruc-

tion of faults needs to be demonstrated by building a VM model to predict process outputs from sensor signals. Correct reconstruction should lead to an increased accuracy of the predicted values of the process outputs.

Fault reconstruction is mainly done in three ways: reconstruction via iteration, the missing value approach, and reconstruction via optimization. Qin et al. [104] showed that all three ways of doing reconstruction mentioned above lead to the same results. Research can be carried out to arrive at better reconstruction methods in the future.

The results presented in Chapter 6 were based on simulated data. In order to validate the practical advantages of using B-EWMA R2R control and VM-assisted EWMA-R2R control over EWMA-R2R control, the control schemes should be compared using industrial data. In this work, the baseline uniform sampling rate was set to 3. In other words, every 3<sup>rd</sup> wafer is measured when the process is in control. For practical purposes, a rigorous economic evaluation should be carried out to determine the baseline uniform sampling rate.

## Bibliography

- [1] T. Adamson, G. Moore, M. Passow, J. Wong, and Y. Xu. Strategies for successfully implementing fab-wide FDC methodologies in semiconductor manufacturing. In *Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XVIII*, Westminster, Colorado, 2006.
- [2] C. Alcala and S. J. Qin. Reconstruction-based contribution for process monitoring. *Automatica*, 45(7):1593–1600, July 2009.
- [3] C. Angeli and A. Chatzinikolaou. On-line fault detection techniques for technical systems: A survey. *International Journal of Computer Science & Applications*, 1(1):12–30, 2004.
- [4] R. Bettocchi, M. Pinelli, P. R. Spina, and M. Venturini. Artificial intelligence for the diagnostics of gas turbines - Part 1: Neural network approach. *Journal of Engineering for Gas Turbines and Power*, 129(3):711–719, 2007.
- [5] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2003.
- [6] C. A. Bode. *Run-to-run Control of Overlay and Linewidth in Semiconductor Manufacturing*. Ph.D. dissertation, The University of Texas at Austin, 2001.

- [7] 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*, 12(7):893 – 900, 2004.
- [8] J. Bokor and Z. Szabo. Fault detection and isolation in nonlinear systems. *Annual Reviews in Control*, 33(2):113–123, 2009.
- [9] W. M. Bolstad. *Introduction to Bayesian Statistics*. Wiley-Interscience, 2004.
- [10] Box and Jenkins. *Time Series Analysis: Forecasting and Control*, volume 2<sup>nd</sup> edition. San Francisco: Holden-Day, 1976.
- [11] W. J. Campbell. *Model Predictive Run-to-run Control of Chemical Mechanical Planarization*. Ph.D. dissertation, The University of Texas at Austin, 1999.
- [12] I. Castillo. *Nonlinear Model-based Fault Detection and Isolation: Improvements in the Case of Single/Multiple Faults and Uncertainties in the Model Parameters*. Ph.D. dissertation, The University of Texas at Austin, 2011.
- [13] K. Chamness and T. F. Edgar. Local-kNN fault detection algorithm description and examples. In *Texas-Wisconsin Modeling and Control Consortium (TWMCC)*, Austin, Texas, 2006.

- [14] K. A. Chamness. *Multivariate Fault Detection and Visualization in the Semiconductor Industry*. Ph.D. dissertation, The University of Texas at Austin, 2006.
- [15] F. T. Cheng, J. Chang, H. Huang, C. Kao, Y. Chen, and J. Peng. Benefit model of virtual metrology and integrating AVM into MES. *IEEE Transactions on Semiconductor Manufacturing*, 24(2):261 – 272, 2011.
- [16] F. T. Cheng, Y. T. Chen, Y. C. Su, and D. L. Zeng. Evaluating reliance level of a virtual metrology system. *IEEE Transactions on Semiconductor Manufacturing*, 21(1):92 – 103, 2008.
- [17] G. Cherry and S. J. Qin. Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *IEEE Transactions on Semiconductor Manufacturing*, 19(2):159–172, 2006.
- [18] G. Chryssolouris, M. Lee, and A. Ramsey. Confidence interval prediction for neural network models. *IEEE Transactions on Neural Networks*, 7(1):229 – 232, 1996.
- [19] H. Cramer. *Mathematical Methods of Statistics*. Princeton University Press, 1946.
- [20] J. Dedecker and E. Rio. On mean central limit theorems for stationary sequences. *Annales de l'Institut Henri Poincaré Probabilités et Statistiques*, 44:693–726, 2008.

- [21] D. Djurdjanovic, J. Lee, and J. Ni. Watchdog agent—an infotronics-based prognostics approach for product performance degradation assessment and prediction. *Advanced Engineering Informatics, Intelligent Maintenance Systems*, 17(3-4):109 – 125, 2003.
- [22] D. Dochain. State and parameter estimation in chemical and biochemical processes: a tutorial. *Journal of Process Control*, 13:801–818, 2003.
- [23] H. H. Doh, P. Wang, N. Xu, P. Yadav, E. Magni, and B. McMillin. Application of PCA/PLS to optical emission spectra for plasma etch process monitoring. In *Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XV*, Colorado, 2003.
- [24] R. Dunia and S. J. Qin. Subspace approach to multidimensional fault identification and reconstruction. *AIChE Journal*, 44:1813–1831, 1998.
- [25] R. Dunia, S. J. Qin, T. F. Edgar, and T. J. McAvoy. Identification of faulty sensors using principal component analysis. *AIChE Journal*, 42:2797–2812, 1996.
- [26] T. F. Edgar, W. J. Campbell, and C. A. Bode. Model-based control in microelectronics manufacturing. In *Proceedings of the 38<sup>th</sup> Conference on Decision and Control, IEEE Control Systems Society*, volume 4, pages 4185–4192, Arizona, USA, 1999.
- [27] E. A. Feinberg and A. N. Shiryaev. Quickest detection of drift change for

- Brownian motion in generalized Bayesian and minimax settings. *Statistics and Decisions*, 24(4):445–470, 2007.
- [28] A. Ferreira and C. Kernaflen. Virtual metrology models for predicting overlay of photolithography process. In *Proceedings of the 10<sup>th</sup> European Advanced Equipment Control/Advanced Process Control (AEC/APC) Conference*, Sicily, Italy, 2010.
- [29] T. Floquet, J. P. Barbot, W. Perruquetti, and M. Djemai. On the robust fault detection via a sliding mode disturbance observer. *International Journal of Control*, 77(7):622–629, 2004.
- [30] P. M. Frank, S. X. Ding, and B. Koppen-Seliger. Current developments in the theory of FDI. In *IFAC Fault Detection, Supervision and Safety of Technical Processes*, Budapest, June 14-16, 2000.
- [31] P. Garimella and B. Yao. Robust model-based fault detection using adaptive robust observers. In *44<sup>th</sup> IEEE Conference on Decision Control/European Control Conference (CDC-ECC)*, pages 3073–3078, Seville, Spain, December 12-15, 2005.
- [32] P. Geladi and B. R. Kowalski. Partial least-squares regression - a tutorial. *Analytica Chimica Acta*, 185:1–17, 1986.
- [33] J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, 1998.

- [34] J. Gertler and J. Cao. PCA-based fault diagnosis in the presence of control and dynamics. *AIChE Journal*, 50(2):388–402, 2004.
- [35] J. Gertler, W. Li, Y. Huang, and T. J. Mcavoy. Isolation-enhanced principal component analysis. *AIChE Journal*, 45(2):323–334, 1999.
- [36] J. D. Gibson, B. R. Koo, and S. D. Gray. Filtering of colored noise for speech enhancement and coding. *IEEE Transactions on Signal Processing*, 39(8):1732–1742, 1991.
- [37] B. S. Gill, T. F. Edgar, and J. D. Stuber. A novel approach to virtual metrology using Kalman filtering. *Future Fab International*, 35:86–91, 2010.
- [38] D. Gleispach and J. Besnard. Metrology models for predicting cvd oxide thickness of a pecvd process. In *Proceedings of the 10<sup>th</sup> European Advanced Equipment Control/Advanced Process Control (AEC/APC) Conference*, Sicily, Italy, 2010.
- [39] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEEE Proceedings On Radar and Signal Processing*, 140(2):107–113, 1993.
- [40] M. S. Grewal and A. P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*, volume 2<sup>nd</sup> edition. John Wiley & Sons, Inc., 2001.

- [41] K. P. Han, T. F. Edgar, and J. R. Moyne. Implementation of virtual metrology by selection of optimal adaptation method. In *Proceedings of Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium*, Ann Arbor, Michigan, 2009.
- [42] Q. P. He. Novel multivariate fault detection methods using Mahalanobis distance. In *Proceedings of Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XVII*, Indian Wells, California, 2005.
- [43] Q. P. He and J. Wang. A multivariate fault detection method using k-nearest-neighbor rule. In *Proceedings of Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XVIII*, Westminster, Colorado, 2006.
- [44] Q. P. He and J. Wang. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):345–354, 2007.
- [45] Q. P. He and J. Wang. Principal component based k-nearest-neighbor rule for semiconductor process fault detection. In *Proceedings of American Control Conference*, pages 1606–1611, Seattle, Washington, 2008.
- [46] Q. P. He and J. Wang. Large-scale semiconductor process monitoring using a fast pattern recognition based method. *IEEE Transactions on Semiconductor Manufacturing*, 23(2):194–200, 2010.

- [47] Q. P. He and J. Wang. Statistics pattern analysis: A new process monitoring framework and its application to semiconductor batch processes. *AIChE Journal*, 57(1):107–121, 2011.
- [48] D. M. Himes, R. H. Storer, and C. Georgakis. Determination of the number of principal components for disturbance detection and isolation. In *American Control Conference*, pages 1279–1283, Baltimore, Maryland, 1994.
- [49] C. C. Hsu and L. S. Chen. Integrate independent component analysis and support vector machine for monitoring non-Gaussian multivariate process. In *Proceedings of 4<sup>th</sup> International Conference on Wireless Communications, Networking and Mobile Computing*, pages 8063–8066, 2008.
- [50] C. C. Hsu, L. S. Chen, and C. H. Liu. A process monitoring scheme based on independent component analysis and adjusted outliers. *International Journal of Production Research*, 48(6):1727–1743, 2010.
- [51] D. A. Hsu. A Bayesian robust detection of shift in the risk structure of stock market returns. *Journal of the American Statistical Association*, 77(377):29–39, 1982.
- [52] M. H. Hung, T. H. Lin, F. T. Cheng, and R. C. Lin. A novel virtual metrology scheme for predicting CVD thickness in semiconductor manufacturing. *IEEE-ASME Transactions on Mechatronics*, 12(3):308–316, 2007.

- [53] R. Isermann. Model-based fault-detection and diagnosis - status and applications. *Annual Reviews in Control*, 29(1):71–85, 2005.
- [54] R. Isermann. On fuzzy logic applications for automatic control, supervision, and fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Part A-Systems and Humans*, 28(2):221–235, March 1998.
- [55] R. Isermann and P. Ball. Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5):709–719, 1997.
- [56] J. E. Jackson. *A Users Guide to Principal Components*. Wiley-Interscience: New York, 1991.
- [57] J. E. Jackson and G. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21:341–349, 1979.
- [58] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82:35–45, 1960.
- [59] P. Kang, H. J. Lee, S. Cho, D. Kim, J. Park, C. K. Park, and S. Doh. A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, 36(10):12554–12561, 2009.

- [60] M. Kano, K. Nagao, S. Hasebe, I. Hashimoto, and H. Ohno. Statistical process monitoring based on dissimilarity of process data. *AIChE Journal*, 48(6):1231–1240, 2002.
- [61] M. Kermit and O. Tomic. Independent component analysis applied on gas sensor array measurement data. *IEEE Sensors Journal*, 3:218–228, 2003.
- [62] A. A. Khan, J. R. Moyne, and D. M. Tilbury. An approach for factory-wide control utilizing virtual metrology. *IEEE Transactions on Semiconductor Manufacturing*, 20(4):364–375, 2007.
- [63] A. A. Khan, J. R. Moyne, and D. M. Tilbury. Virtual metrology and feedback control for semiconductor manufacturing processes using recursive partial least squares. *Journal of Process Control*, 18(10):961–974, 2008.
- [64] B. Kim and S. Kim. Prediction of plasma etching using a classification-based neural network. *Journal of the Electrochemical Society*, 151(9):C585–C589, 2004.
- [65] T. Kourti. Multivariate dynamic data modeling for analysis and statistical process control of batch processes, start-ups and grade transitions. *Journal of Chemometrics*, 17:93–109, 2003.
- [66] T. Kourti. Application of latent variable methods to process control and multivariate statistical process control in industry. *International*

- Journal of Adaptive Control and Signal Processing*, 19(4):213–246, 2005.
- [67] T. Kourti and J. F. MacGregor. Multivariate SPC methods for process and product monitoring. *Journal of Quality Technology*, 28:409–428, 1996.
- [68] T. Kourti, P. Nomikos, and J. F. MacGregor. Analysis, monitoring, and fault diagnosis of batch processes using multi-block and multi-way PLS. *Journal of Process Control*, 5:277–284, 1995.
- [69] J. Lacaille and M. Zagrebnov. An unsupervised diagnosis for process tool fault detection: the flexible golden pattern. *IEEE Transactions on Semiconductor Manufacturing*, 20:355–363, 2007.
- [70] H. Lee. *Advanced Process Control and Optimal Sampling in Semiconductor Manufacturing*. Ph.D. dissertation, The University of Texas at Austin, 2008.
- [71] J. Lee, S. J. Qin, and I. B. Lee. Fault detection and diagnosis based on modified independent component analysis. *AIChE Journal*, 52:3501–3514, 2006.
- [72] J. M. Lee, C. K. Yoo, and I. B. Lee. Statistical process monitoring with independent component analysis. *Journal of Process Control*, 14(5):467–485, 2004.

- [73] S. F. Lee and C. J. Spanos. Prediction of wafer state after plasma processing using real-time tool data. *IEEE Transactions on Semiconductor Manufacturing*, 8(3):252–261, 1995.
- [74] H. J. Levinson. *Lithography Process Control*. SPIE Optical Engineering Press, Bellingham, WA, 1999.
- [75] T. H. Lin, F. T. Cheng, W. M. Wu, C. A. Kao, A. J. Ye, and F. C. Chang. Nn-based key-variable selection method for enhancing virtual metrology accuracy. *IEEE Transactions on Semiconductor Manufacturing*, 22(1):204–211, 2009.
- [76] J. F. MacGregor and T. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414, 1995.
- [77] K. V. Mardia. *Multivariate Analysis-IV*, chapter :Mahalanobis distances and angles, pages 495–511. North-Holland: Amsterdam, 1977.
- [78] D. W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- [79] L. F. Mendonca, J. M. C. Sousa, and J. M. G. Sa da Costa. An architecture for fault detection and isolation based on fuzzy methods. *Expert Systems With Applications*, 36(2, Part 1):1092–1104, March 2009.

- [80] P. Miller, R. E. Swanson, and C. F. Heckler. Contribution plots: The missing link in multivariate quality control. In *Fall Conference of the ASQC and ASA*, Milwaukee, Wisconsin, 1993.
- [81] T. Moore, B. Harner, G. Kestner, C. Baab, and J. Stanchfield. Intel's FDC proliferation in 300 mm HVM: progress and lessons learned. In *Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XVIII*, Westminster, Colorado, 2006.
- [82] J. Moyne, E. D. Castillo, and A. M. Hurwitz. *Run-to-run Control in Semiconductor Manufacturing*. CRC Press; 1<sup>st</sup> edition, 2004.
- [83] J. Musacchio. *Run-to-run Control in Semiconductor Manufacturing*. Master's thesis, University of California, Berkeley, 1997.
- [84] P. Nomikos and J. F. MacGregor. Monitoring of batch processes using multiway principal component analysis. *AIChE Journal*, 40:1361–1375, 1994.
- [85] P. Nomikos and J. F. MacGregor. Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30:97–108, 1995.
- [86] P. Nomikos and J. F. MacGregor. Multivariate SPC charts for monitoring batch processes. *Technometrics*, 37:41–59, 1995.
- [87] A. Norvilas, E. Tatara, A. Negiz, J. DeCicco, and A. Cinar. Monitoring and fault diagnosis of a polymerization reactor by interfacing knowledge-

- based and multivariate SPM tools. In *American Control Conference*, volume 6, pages 3773–3777, June 1998.
- [88] R. K. Nurani, R. Akella, A. J. Strojwas, R. Wallace, M. G. McIntyre, J. Shields, and I. Emami. Development of an optimal sampling strategy for wafer inspection. In *Extended Abstracts of International Symposium on Semiconductor Manufacturing (ISSM)*, pages 143–146, 1994.
- [89] R. Pachter, R. B. Altman, and O. Jardetzky. The dependence of a protein solution structure on the quality of the input NMR data - Application of the double-iterated Kalman filter technique to Oxytocin. *Journal of Magnetic Resonance*, 89(3):578–584, 1990.
- [90] V. Palade, D. C. Bocaniala, and L. C. Jain. *Computational Intelligence in Fault Diagnosis*. Springer-Verlag, 2006.
- [91] J. C. H. Pan and D. H. E. Tai. Implementing virtual metrology for in-line quality control in semiconductor manufacturing. *International Journal of Systems Science*, 40(5):461–470, 2009.
- [92] Pankratz. *Forecasting With Univariate Box-Jenkins Models: Concepts and Cases*. New York: Wiley, 1983.
- [93] A. J. Pasadyn. *Simultaneous Run-to-Run Control and Identification for Multiple Products and Process Environments*. Ph.D. dissertation, The University of Texas at Austin, 2001.

- [94] A. J. Pasadyn, H. Lee, and T. F. Edgar. Scheduling semiconductor manufacturing processes to enhance system identification. *Journal of Process Control*, 18(10):946 – 953, 2008.
- [95] R. J. Patton and C. J. Lopez-toribio. Soft computing approaches to fault diagnosis for dynamic systems: a survey. In *Proceedings of the 4<sup>th</sup> IFAC Symposium: Safeprocess*, pages 298–311, 2000.
- [96] R. J. Patton and C. J. Lopez-Toribio. Artificial intelligence approaches to fault diagnosis. In *Update on Developments in Intelligent Control*, pages 1–12, October 1998.
- [97] F. Pene. Rate of convergence in the multidimensional central limit theorem for stationary processes: Application to the Knudsen gas and to the Sinai billiard. *Annals of Applied Probability*, 15:2331–2392, 2005.
- [98] A. Prabhu. *Performance Monitoring of Run-to-run Control Systems Used in Semiconductor Manufacturing*. Ph.D. dissertation, The University of Texas at Austin, 2008.
- [99] S. J. Qin. Recursive pls algorithms for adaptive data modeling. *Computers & Chemical Engineering*, 22(4-5):503–514, 1998.
- [100] S. J. Qin. Statistical process monitoring: basics and beyond. *Journal of Chemometrics*, 17(8-9):480–502, Aug-Sep 2003.

- [101] S. J. Qin and W. Li. Detection, identification and reconstruction of faulty sensors with maximized sensitivity. *AIChE Journal*, 45:1963–1976, 1999.
- [102] S. J. Qin and W. Li. Detection and identification of faulty sensors in dynamic processes. *AIChE Journal*, 47(7):1581–1593, 2001.
- [103] S. J. Qin, S. Valle-Cervantes, and M. Piovoso. On unifying multi-block analysis with applications to decentralized process monitoring. *Journal of Chemometrics*, 15:715–742, 2001.
- [104] S. J. Qin, H. Yue, and R. Dunia. Self-validating inferential sensors with application to air emission monitoring. *Industrial & Engineering Chemistry Research*, 36:1675–1685, 1997.
- [105] E. Ragnoli and S. McLoone. A multiple modelling approach to a virtual metrology case study using oes. In *Proceedings of the 10<sup>th</sup> European Advanced Equipment Control/Advanced Process Control (AEC/APC) Conference*, Sicily, Italy, 2010.
- [106] A. Raich and A. Cinar. Statistical process monitoring and disturbance diagnosis in multivariate continuous processes. *AIChE Journal*, 42:995–1009, 1996.
- [107] I. Rivals and L. Personnaz. Construction of confidence intervals for neural networks based on least squares estimation. *Neural Networks*, 13(4-5):463 – 484, 2000.

- [108] A. S. Ross G. H. Bodammer S. Smith, A. J. Walton and J. M. Stevenson. Evaluation of the issues involved with test structures for the measurement of sheet resistance and linewidth of copper damascene interconnect. In *Proceedings of IEEE International Conference on Microelectronic Test Structures*, pages 195–200, 2001.
- [109] S. Samata, Y. Ushiku, K. Ishii, M. Tanaka, T. Furuhashi, T. Nakao, and A. Yamamoto. Fault detection method for dry vacuum pump of lpcvd system. In *Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XIV*, Snowbird, Utah, 2002.
- [110] E. N. Sanchez, D. A. Suarez, and J. A. Ruz. Fault detection in fossil electric power plant via neural networks. In *Proceedings of the Automation Congress*, volume 17, pages 213–218, June 28 - July 1, 2004.
- [111] C. Schoene. *Electrical parameter control for semiconductor manufacturing*. Ph.D. dissertation, The University of Texas at Austin, 2007.
- [112] C. Shan, P. Tianhong, and J. ShiShang. Development of a virtual metrology for high-mix TFT-LCD manufacturing processes. *Journal of Semiconductors*, 31(11):116006/1 – 116006/5, 2010.
- [113] J. Singh and S. Sapatnekar. Statistical timing analysis with correlated non-Gaussian parameters using independent component analysis. In *Proceedings of 43<sup>rd</sup> Design Automation Conference*, pages 155–160, 2006.

- [114] A. Singhal and D. Seborg. Pattern matching in multivariate time series databases using a moving-window approach. *Industrial & Engineering Chemistry Research*, 41:3822–3838, 2002.
- [115] A. Singhal and D. Seborg. Evaluation of a pattern matching method for the tennessee eastman challenge process. *Journal of Process Control*, 16:601–613, 2006.
- [116] O. A. Z. Sotomayor and D. Odloak. Observer-based fault diagnosis in chemical plants. *Chemical Engineering Journal*, 112:93–108, 2005.
- [117] R. Sreedhar, B. Fernandez, and G. Y. Masada. A neural network based adaptive fault detection scheme. In *American Control Conference*, volume 5, pages 3259–3263, June 1995.
- [118] M. Staroswiecki and G. Comtet-Varga. Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, 37(5):687–699, 2001.
- [119] T. Stutts. Method for calculating variable influences on the Mahalanobis distance. In *Proceedings of Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XX*, Salt Lake City, Utah, 2008.
- [120] Y. C. Su, T. H. Lin, F. T. Cheng, and W. M. Wu. Accuracy and real-time considerations for implementing various virtual metrology algorithms. *IEEE Transactions on Semiconductor Manufacturing*, 21(3):426–434, 2008.

- [121] N. Tracy, J. Young, and R. Mason. Multivariate control charts for individual observations. *Journal of Quality Technology*, 24:88–95, 1992.
- [122] V. Uraikul, C. W. Chan, and P. Tontiwachwuthikul. Artificial intelligence for monitoring and supervisory control of process systems. *Engineering Applications of Artificial Intelligence*, 20(2):115–131, 2007.
- [123] S. Valle, W. Li, and S. J. Qin. Selection of the number of principal components: The variance of the reconstruction error criterion with a comparison to other methods. *Industrial and Engineering Chemistry Research*, 38:4389–4401, 1999.
- [124] V. Venkatasubramanian, R. Rengaswamy, and S.N. Kavuri. A review of process fault detection and diagnosis: Part 2: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313–326, 2003.
- [125] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part 3: Process history based methods. *Computers & Chemical Engineering*, 27(3):327–346, 2003.
- [126] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S.N. Kavuri. A review of process fault detection and diagnosis: Part 1: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293–311, 2003.

- [127] J. Wang and C. Bode. Bayesian statistics and its application in semiconductor manufacturing. In *Proceedings of Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XV*, Colorado, USA, 2003.
- [128] J. Wang and Q. P. He. A Bayesian approach for disturbance detection and classification and its application to state estimation in run-to-run control. *IEEE Transactions on Semiconductor Manufacturing*, 20(2):126–136, 2007.
- [129] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde. Generalized contribution plots in multivariate statistical process monitoring. *Chemometrics and Intelligent Laboratory Systems*, 51:95–114, 2000.
- [130] J. A. Westerhuis, T. Kourti, and J. F. MacGregor. Comparing alternative approaches for multivariate statistical analysis of batch process data. *Journal of Chemometrics*, 13:397–413, 1999.
- [131] R. Williams, D. Gudmundsson, K. Monahan, and J. G. Shanthikumar. Optimized sample planning for wafer defect inspection. In *Proceedings of IEEE International Symposium on Semiconductor Manufacturing*, pages 43–46, 1999.
- [132] B. M. Wise. Metal etch data for fault detection evaluation. Available at <http://software.eigenvector.com/Data/Etch/index.html>, 1999.

- [133] B. M. Wise, N. B. Gallagher, R. Bro, J. M. Shaver, W. Winding, and R. S. Koch. PLS toolbox user manual. Eigenvector Research Inc., 2006.
- [134] B. M. Wise, N. B. Gallagher, S. W. Butler, D. D. White, and G. G. Barna. A comparison of principal component analysis, multiway principal component analysis, trilinear decomposition and parallel factor analysis for fault detection in a semiconductor etch process. *Journal of Chemometrics*, 13(3-4):379–396, 1999.
- [135] B. M. Wise, N. B. Gallagher, and E. B. Martin. Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch. *Journal of Chemometrics*, 15:285–298, 2001.
- [136] S. Wold. Cross-validated estimation of the number of components in factor and principal components models. *Technometrics*, 20(4):397–406, 1978.
- [137] J. Wong. Batch PLS analysis and FDC process control of within lot SiON gate oxide thickness variation in sub-nanometer range. In *Advanced Equipment Control/Advanced Process Control (AEC/APC) Symposium XVIII*, Colorado, 2006.
- [138] Y. H. Wu, Y. H. Yang, S. K. Qin, and X. B. Chen. Process monitoring and fault detection method based on independent component analysis. In *Proceedings of the Sixth World Congress on Intelligent Control and Automation*, pages 5586–5589, 2006.

- [139] S. S. Xiong and Z. Y. Zhou. Neural filtering of colored noise based on Kalman filter structure. *IEEE Transactions on Instrumentation and Measurement*, 52(3):742–747, 2003.
- [140] Y. F. Xue, Y. J. Wang, and J. Yang. Independent component analysis based on gradient equation and kernel density estimation. *Neurocomputing*, 72(7-9):1597–1604, 2009.
- [141] J. Yan and J. Lee. Introduction of watchdog prognostics agent and its application to elevator hoistway performance assessment. *Journal of the Chinese Institute of Industrial Engineers*, 22(1):56 – 63, 2005.
- [142] X. Yan and C. Edwards. Robust sliding mode observer-based actuator fault detection and isolation for a class of nonlinear systems. *International Journal of Systems Science*, 39(4):349–359, 2008.
- [143] S. Yoon and J. F. MacGregor. Fault diagnosis with multivariate statistical models: Part 1: Using steady state fault signatures. *Journal of Process Control*, 11(4):387–400, August 2001.
- [144] H. Yue and S. J. Qin. Fault reconstruction and identification for industrial processes. In *AIChE Annual Meeting*, Miami, Florida, 1998.
- [145] H. Yue and S. J. Qin. Reconstruction based fault identification using a combined index. *Industrial & Engineering Chemistry Research*, 40:4403–4414, 2001.

- [146] H. Yue, S. J. Qin, R. Markle, C. Nauert, and M. Gatto. Fault detection of plasma etchers using optical emission spectra. *IEEE Transactions on Semiconductor Manufacturing*, 13:374–385, 2000.
- [147] H. Yue, S. J. Qin, J. Wiseman, and A. Toprac. Plasma etching endpoint detection using multiple wavelengths for small open area wafers. *Journal of Vacuum Science and Technology*, 19:66–75, 2001.
- [148] H. H. Yue and M. Tomoyasu. Weighted principal component analysis and its applications to improve FDC performance. In *43<sup>rd</sup> IEEE Conference on Decision and Control*, pages 4262–4267, Bahamas, 2004.
- [149] D. K. Zeng and C. J. Spanos. Virtual metrology modeling for plasma etch operations. *IEEE Transactions on Semiconductor Manufacturing*, 22(4):419–431, 2009.
- [150] J. Zhang, A. J. Morris, E. B. Martin, and C. Kiparissides. Estimation of impurity and fouling in batch polymerisation reactors through the application of neural networks. *Computers & Chemical Engineering*, 23(3):301–314, 1999.
- [151] Y. Zhang. *Improved Methods in Statistical and First Principles Modeling for Batch Process Control and Monitoring*. Ph.D. dissertation, The University of Texas at Austin, 2008.
- [152] Y. Zhang and T. F. Edgar. *New Directions in Bio-process Modeling and Control*, chapter 8: Multivariate statistical process control. ISA, 2006.

- [153] Y. Zhang and T. F. Edgar. A robust dynamic time warping algorithm for batch trajectory synchronization. In *Proceedings of the American Control Conference*, pages 2864–2869, Seattle, Washington, 2008.
- [154] Y. Zhou, J. Hahn, and M. S. Mannan. Fault detection and classification in chemical processes based on neural networks with feature extraction. *ISA Transactions*, 42(4):651–664, 2003.

## Vita

After graduating from Hindu College, Amritsar in 2003, Bhalinder Singh Gill entered Indian Institute of Technology (IIT), Guwahati. In Summer 2006, he was invited to work on a research project on a simulated moving bed process by the process control group at *Universität* Dortmund, Germany with financial support from German Academic Exchange Service, DAAD. Working with a team of Ph.D. students motivated him to pursue a doctoral degree in process systems engineering. In May 2007, he received his Bachelor of Technology degree in Chemical Engineering from Indian Institute of Technology (IIT), Guwahati. He entered graduate school at the University of Texas at Austin in August, 2007 and worked as a graduate research assistant under the supervision of Thomas F. Edgar. His research focuses on the modeling and control of semiconductor manufacturing processes. The research being supported by Texas Instruments Inc. enabled him to work on industrial data and gain in-depth knowledge of semiconductor manufacturing processes. He started working as a R2R process control engineer at Micron Technology, Inc. in Manassas, Virginia in July 2011.

Permanent address: bhalindersingh@gmail.com

This dissertation was typeset with L<sup>A</sup>T<sub>E</sub>X<sup>†</sup> by the author.

---

<sup>†</sup>L<sup>A</sup>T<sub>E</sub>X is a document preparation system developed by Leslie Lamport as a special

---

version of Donald Knuth's  $\text{\TeX}$  Program.