

Copyright
by
Alexander Q. Bleakie
2014

**The Dissertation Committee for Alexander Q. Bleakie certifies
that this is the approved version of the following dissertation:**

**Integrated Performance Prediction and Quality Control
in Manufacturing Systems**

SUPERVISING COMMITTEE:

Dragan Djurdjanovic, Supervisor

Andrew Bailey

Thomas Edgar

Eric Fahrenthold

Benito Fernandez

**Integrated Performance Prediction and Quality Control
in Manufacturing Systems**

by

Alexander Q. Bleakie, B.S.; M.S.E.

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

December 2014

This is dedicated to my father Robert Maxwell Bleakie II (1950-2010).

Acknowledgments

First and above all, I would like to thank my supervisor, Dr. Dragan Djurdjanovic, who always guided me with full support these past six years to make this dissertation possible even through the toughest times. Next, I thank my committee members: Dr. Fernandez for teaching me many concepts related to mechatronics, dynamics, control, and machine learning and being one of the best educators at this University; Dr. Fahrenthold another one of the best educators at this University for bringing his impeccable physics knowledge and for always supporting me, allowing me to teach for him many times throughout the years, and making education number one; Dr. Edgar for his invaluable help from his research and students; and finally Dr. Drew Bailey for your vast knowledge about the subject matter in my research, believing in my work, and for giving me a chance to begin an exciting career.

I would also like to acknowledge my colleagues that have been in the *Limes Lab* at any time in the past six years: Michael Cholette, Marcus Musselman, Merve Celen, Yibo Jiao, Jeff Krol, Yi Liu, Asad Ul Haq, Matt Graves, Derek Zhang, Kevin Wang, each individually have dedicated much of their time teaching me throughout our work together. Other colleagues that have helped me at various times along the way and need to be acknowledged are Donghwan Kim, Jonathan Lesage, Albert Espinoza, Tehmoor Dar, Jaewon Choi, Cameron Booth, Felipe Lopez, and Jack Hall. All of these people are doing good work.

On a personal note, I have to thank my grandmother Annie Vaught who just celebrated her 90th birthday. I feel blessed that you are around to see this accomplishment

and I hope to see you many more times in the future. My father Robert Bleakie and grandfather Joe Vaught, who I cannot share this accomplishment with, should be praised as defining the standard of hard work and showing me to never give up no matter how hard a situation gets. My older brother Beau and mom Joyce Bleakie, I thank you both for your full support from afar and everything that you have ever done for me. My accomplishments are the results of your unwavering confidence in me and I am grateful for having you around. I want to also thank my aunts Mary and Sue Bleakie, uncles Tim and Joe Bleakie, and cousins for supporting me.

I want to thank all of my friends and other mentors that I have had in the past. I thank my many long-time friends that I have back in Seabrook, TX for supporting me and for always letting me feel at home any time I come back even after all these years. Also, thank you to all of my friends in Austin that have allowed me to have some fun at times, specifically everyone in the Bartlett family for supporting me in this long endeavor. Finally, a few special professors that have propelled me to succeed in many vital times in my college life, namely Dr. Bruton, Dr. Downing, and Dr. Gruebel from Stephen F. Austin State University and Dr. Rasmussen at Texas A&M University.

Integrated Performance Prediction and Quality Control in Manufacturing Systems

Alexander Q. Bleakie, Ph.D.

The University of Texas at Austin, 2014

Supervisor: Dragan Djurdjanovic

Abstract

Predicting the condition of a degrading dynamic system is critical for implementing successful control and designing the optimal operation and maintenance strategies throughout the lifetime of the system. In many situations, especially in manufacturing, systems experience multiple degradation cycles, failures, and maintenance events throughout their lifetimes. In such cases, historical records of sensor readings observed during the lifecycle of a machine can yield vital information about degradation patterns of the monitored machine, which can be used to formulate dynamic models for predicting its future performance. Besides the ability to predict equipment failures, another major component of cost effective and high-throughput manufacturing is tight control of product quality. Quality control is assured by taking periodic measurements of the products at various stages of production. Nevertheless, quality measurements of the product require time and are often executed on costly measurement equipment, which increases the cost of manufacturing and slows down production. One

possible way to remedy this situation is to utilize the inherent link between the manufacturing equipment condition, mirrored in the readings of sensors mounted on that machine, and the quality of products coming out of it. The concept of Virtual Metrology (VM) addresses the quality control problem by using data-driven models that relate the product quality to the equipment sensors, enabling continuous estimation of the quality characteristics of the product, even when physical measurements of product quality are not available. VM can thus bring significant production benefits, including improved process control, reduced quality losses and higher productivity. In this dissertation, new methods are formulated that will combine long-term performance prediction of sensory signatures from a degrading manufacturing machine with VM quality estimation, which enables integration of predictive condition monitoring (prediction of sensory signatures) with predictive manufacturing process control (predictive VM model). The recently developed algorithm for prediction of sensory signatures is capable of predicting the system condition by comparing the similarity of the most recent performance signatures with the known degradation patterns available in the historical records. The method accomplishes the prediction of non-Gaussian and non-stationary time-series of relevant performance signatures with analytical tractability, which enables calculations of predicted signature distributions with significantly greater speeds than what can be found in literature. VM quality estimation is implemented using the recently introduced growing structure multiple model system paradigm (GSMMS), based on the use of local linear dynamic models. The concept of local models enables representation of complex, non-linear dependencies with non-Gaussian and non-stationary noise characteristics, using a locally tractable model representation. Localized modeling enables a VM that can

detect situations when the VM model is not adequate and needs to be improved, which is one of the main challenges in VM. Finally, uncertainty propagation with Monte Carlo simulation is pursued in order to propagate the predicted distributions of equipment signatures through the VM model to enable prediction of distributions of the quality variables using the readily available sensor readings streaming from the monitored manufacturing machine. The newly developed methods are applied to long-term production data coming from an industrial plasma-enhanced chemical vapor deposition (PECVD) tool operating in a major semiconductor manufacturing fab.

Table of Contents

Abstract.....	vii
List of Tables	xii
List of Figures.....	xiii
Chapter 1	1
Introduction.....	1
1.1 Motivation and Background.....	1
1.2 Research Objectives and Challenges.....	4
1.3 Outline of the Dissertation	8
Chapter 2	9
Review of Time-series Prediction, Quality Estimation, and Uncertainty Propagation in Dynamic Models	9
2.1 Time-series Prediction Methodology	10
2.2 Quality Estimation using Virtual Metrology.....	16
2.3 Uncertainty Propagation Methods for Dynamic Systems	26
Chapter 3	30
Analytical Approach to Similarity-Based Performance Prediction of Monitored Systems	30
3.1 Methodology Overview.....	30
3.2 Concept of Similarity Vectors.....	35
3.3 Weighted Likelihood Estimation of Mixture Gaussians with Similarity Vectors	37
3.4 Expectation-Maximization Algorithm Modifications for the Weighted Likelihood Estimation of Mixture Gaussians with Similarity Vectors	39
3.5 Results of Similarity-Based Performance Prediction	44
3.5.1 Experimental System Description	44
3.5.2 PECVD Tool Subsystems and Sensors	44
3.5.3 Tool Operation and Maintenance Schedule.....	46
3.5.4 Results from Experimental Study.....	48
3.5.5 Conclusions of Similarity Based Prediction Results	55

Chapter 4	57
Virtual Metrology (VM) Quality Estimation using Local Dynamic Model Paradigm.....	57
4.1 Overview and Motivation.....	57
4.2 Growing Structure Multiple Model System (GSMMS) Methodology	63
4.2.1 Structural Model Parameters	66
4.2.2 Local Model Parameters.....	69
4.2.3 Updating Methodology for VM Model	71
4.3 Results in VM Quality Estimation using Local Dynamic Model Paradigm	75
4.3.1 Description and Results of VM Comparison Between PLS, T-PLS and GSMMS	78
4.3.2 Conclusions for Comparison of PLS, T-PLS and GSMMS VM Models.....	96
Chapter 5	98
Integration of Performance Prediction and Quality Estimation of Monitored Systems using Uncertainty Propagation Techniques.....	98
5.1 Uncertainty Propagation through Virtual Metrology Model Methodology	98
5.2 Results on Applying Integrated Performance Prediction and Quality Estimation to Semiconductor Manufacturing	106
5.3.1 Conclusions for Integrated Performance Prediction and Quality Estimation in Manufacturing Systems	116
Chapter 6	117
Summary and Future Work	117
6.1 Scientific Contributions.....	120
6.2 Publications	121
Appendix.....	123
Bibliography	129
Vita	141

List of Tables

3.1. Features extracted from the data analyzed in this study.....	49
3.2. Results of the sensitivity analysis between pre and post In-situ clean features.....	51
4.1. Features extracted from the sensor readings to be used as inputs for the VM model.....	76
4.2. The features obtained using forward selection applied to the original set of 1274 features and used as inputs for the VM study.....	78

List of Figures

1.1. Three dimensional system experiencing long-term degrading performance shown in the state and residual spaces.....	2
1.2. Flowchart demonstrating integrated performance prediction and quality estimation methodology proposed in this work.....	6
2.1. Concept of replacing quality measurements with statistical models utilizing equipment data (Virtual Metrology) [20].....	18
3.1. Conceptual illustration of degradation dynamics for a system with different fault modes, monitored using two features (residuals).....	32
3.2. Steps for the similarity-based prediction algorithm described in this work.....	34
3.3. Pseudo-code for the modified EM algorithm utilized in this work for GMM fitting.....	43
3.4. Diagram representing the various components that make up the PECVD tool.....	45
3.5. Production cycle of the PECVD tool.....	48
3.6. Various degradation paths for three select runs of the PECVD tool using two sensitive dynamic features.....	52
3.7. Mean squared errors and computation time for the three prediction algorithms, starting at <i>cycle 11</i> in each of the 40 test runs and predicting 25 cycles (wafers) ahead. Results were averaged over 40 test runs.....	54
3.8. Mean squared errors and computation time for the three prediction algorithms, starting at <i>cycle 21</i> in each of the 40 test runs and predicting 15 cycles (wafers) ahead. Results were averaged over 40 test runs.....	54
4.1. Partition of input-output space using self-organizing network; <i>a)</i> Voronoi tessellation and <i>b)</i> self-organizing network.....	60
4.2. Illustration of localized model residuals for help in discerning model validity over the entire operating space.....	62

4.3. Divide and conquer approximation of non-linear behavior using GSMMS with two inputs, one output, and five local regression models.....	65
4.4. Illustration of the training procedure for the GSMMS and the breadth-first algorithm for calculating the shortest distance between the BMU and its neighboring nodes on the SOM.....	67
4.5. Illustration of updating GSMMS based on unusual input patterns. The original GSMMS model in plot (a) estimates the quality variable based on the equipment signatures available at that moment. When a number of equipment signatures appear outside any previously seen local region as shown in plot (b), the original map can grow following the growing method from [159] (plot (c-d)) to incorporate the new operating regime and refit the model for better quality estimation in the new regime (plot (e)).....	74
4.6. VM results of comparing PLS, T-PLS and GSMMS for prediction of mean film thickness using VM updating based on unusual inputs.....	81
4.7. VM results of comparing PLS, T-PLS and GSMMS for prediction of mean film thickness using VM updating based on fixed schedule measurements. Please note that “Model Updates” counts only the updates from the scheduled measurements (triggered from the 1% prediction error threshold).....	83
4.8. VM results of comparing PLS, T-PLS and GSMMS based VM for prediction of mean film thickness, using concurrent model updating based on fixed schedule measurements and based on unusual inputs. Please note that “Model Updates” counts only the updates from the scheduled measurements (triggered from the 1% prediction error threshold). Also, the unscheduled measurements bring their own model updates which should be added to this number to find the total model updates.....	85
4.9. Mean squared error for predicting mean wafer thickness of PLS, T-PLS and GSMMS models as the time between scheduled physical wafer measurements increases from 1/8 wafers to 1/200 wafers. GSMMS using both types of updating outperforms the other methods, providing equal accuracy with significantly fewer measurements.....	87

4.10. Mean squared error results similar to the previous figure using an optimal 100 wafer moving window for updating the PLS and T-PLS models. GSMMS using both types of updating outperforms both models.....89

4.11. Plot (a) shows percentages of wafers for which PLS, T-PLS, and GSMMS based VM had relative errors below 1%, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this research. Plot (b) gives analogous results to those from plot (a), except that it shows percentages of wafers with relative errors of VM below 3%. Plot (c) shows maximal relative errors corresponding to the PLS, T-PLS, and GSMMS based VM methods, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this thesis. Please note that a 1% difference in the number of wafers in this dataset is around 300 wafers.....92

4.12. Plot (a) shows percentages of wafers for which windowed PLS, windowed T-PLS, and GSMMS based VM had relative errors below 1%, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this research. Plot (b) gives analogous results to those from plot (a), except that it shows percentages of wafers with relative errors of VM below 3%. Plot (c) shows maximal relative errors corresponding to the windowed PLS, windowed T-PLS, and GSMMS based VM methods, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this thesis. Please note that a 1% difference in the number of wafers in this dataset is around 300 wafers.....94

4.13. VM results of comparing PLS, T-PLS and GSMMS based VM for prediction of mean film thickness, by demonstrating how much updating must be needed when simulating the entire dataset for a second time around. Updating here uses concurrent model updating based on fixed schedule measurements and based on unusual inputs exactly as Figure 4.8.....96

5.1 Demonstration of the predictive VM framework developed in this Chapter.....100

5.2	The process of using the GSMMS VM model directly to predict the quality measurements, while using uncertainty propagation through the GSMMS to obtain and update long-term prediction distributions of the quality measurements.....	104
5.3	Flowchart of the methodology realizing the newly derived predictive VM concept. It uses Monte Carlo random sampling from the predicted tool signature distributions which are then fed through the GSMMS based VM model from Chapter 4. To each VM sample, region-specific noise samples are added to produce samples for the predicted quality variables. These steps are performed consecutively each time a new set of equipment signatures arrives and the predicted tool signatures are updated following the methodology from Chapter 3.....	105
5.4	Illustration of the ideal linear behavior of the prediction error and range of the prediction distribution as the current run progresses. One can see, at a particular current cycle within the run, as it predicts further, the absolute error and range of the prediction become larger. Also, as the current run progresses from cycle 1 to 10 to 20, the absolute error and range both become smaller, demonstrating the capability of similarity based predictions along with an accurate VM model.....	108
5.5	Predictions with 95% confidence limits of the mean wafer thicknesses for an entire run based on the integrated predictive VM approach. On the left are the multiple step ahead predictions starting from cycle 1, 10 and 20 respectively. On the right are the one step ahead predictions for the entire run, which is useful in wafer-to-wafer control.....	109
5.6	Absolute prediction error and 95% confidence limits of prediction distributions as the run in Figure 5.5 progresses. The top two plots show multistep ahead errors and ranges at various snapshots corresponding to Figure 5.5 and comparable to the ideal situation in Figure 5.4. These errors should ideally grow as one predicts forward, but drop as one gathers more information about the current run. The bottom two plots show the predictions and confidence limits as a function of the current cycle, for predicting select cycles within the run. These ideally should drop for every cycle as the run progresses.....	110

5.7	Averaged predictions and 95% confidence limits for multi-step ahead predictions starting at current cycle 1, 10, 20, and 30. One can see that on average, the prediction are close to the actual and there is increasing confidence as the run progresses.....	112
5.8	Ranges of the 95% confidence limits of multi-step ahead predictions starting from cycle 1, 10, 20 and 30 respectively of the averaged run. One can see the range dropping, indicating a more confidence prediction as the current run progresses..	112
5.9	Averaged prediction errors of select cycles as a function of the current cycle. One can see the downward slopes, indicating better prediction of these cycles as the current cycle moves forward, on average.....	114
5.10	Slopes of the averaged prediction errors for all cycles being predicted within a run. The majority of the cycles being predicted have negative slopes, meaning most cycles being predicted within the run have increasing accuracy as the current run progresses on average.....	115
5.11	Percentage of actual metrology falling within the predicted metrology 95% confidence limits for each cycle being predicted within current runs. 83% of the wafers on average are predicted within the limits.....	115

Chapter 1

Introduction

1.1 Motivation and Background

Degradation and failure of machines and products occur in many settings every day, leading to significant cost, wasted material, and hazardous environments to workplaces and consumers. In 1981, maintenance costs in the United States economy were estimated at \$600 billion, a figure that doubled in the subsequent 20 years, with an estimated 30-50% of these costs wasted through ineffective maintenance and unexpected failures [1], [2]. Additionally, disasters from major industries (transportation, manufacturing, chemical, etc.) lead to wasted energy/materials, process shutdowns, environmental contaminations, and most importantly, loss of human lives [3]- [6]. These losses are key motivating factors driving condition-based and predictive maintenance research efforts aimed at pursuing maintenance policies based on the current and predicted system conditions, as assessed from the readings of the sensors mounted on the monitored system. Such information about the actual condition of the equipment can be used to make maintenance decisions that are optimally synchronized with human and material resources in the system, and the least intrusive on the overall operations.

Generally speaking, engineered systems properly accomplish their functions according to design specifications, until degradation and component failures lead to undesired performance. System condition is commonly monitored using the difference in sensor features taken when the system operates currently in the field with features taken

during normal, or desired operations (these differences are often referred to as residuals [7]). Visualizing in three dimensions, Figure 1.1 shows a hypothetical system designed to follow a specified path in its feature space. The desired operation and two known faults are shown in the feature and residual spaces. By looking at the residual space, the normal operation of the system is centered around the origin, while the long term degradation is detected via trends of residuals deviating from this region in various directions, depending on the type of the underlying fault that causes the degradation. Proper maintenance brings the system back into its desired condition and brings the residuals back toward the origin. Systems exhibiting variable or unforeseen operating conditions should expect different normal operation residuals for the different regimes in the feature space and during transitioning periods. However, during normal operation the residuals should be as close to zero mean with as minimal variance as possible, with limits set based on the physics of the system to define what is “normal” for all operating paths.

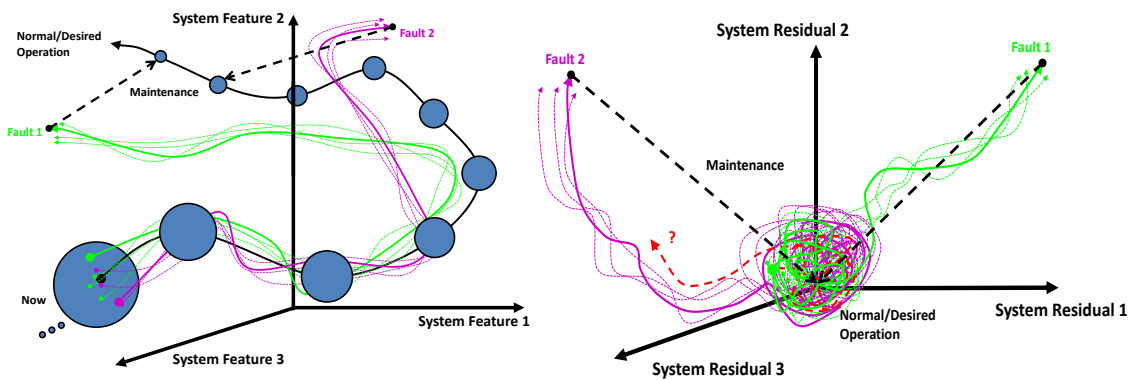


Figure 1.1: Three dimensional system experiencing long-term degrading performance shown in the feature and residual spaces.

The prediction accuracy of degrading performance, particularly in the long-term, is an essential part of postulating cost effective maintenance policies [8] - [10]. Predictive performance information about the monitored system allows one to predict its future degradation state, failure modes, and remaining useful life (RUL), thus enabling one to avoid sudden unexpected failures and allow optimal maintenance strategies [11], [12]. In particular, accurate predictions of the long-term performance of degrading equipment are necessary in order to have sufficient time to prepare maintenance operations [13], [14]. Hence, long-term prediction accuracy of the time-series' of sensory signatures indicative of the condition of the monitored process plays an important role for implementing predictive maintenance in many industries.

In addition to predicting the long-term performance of the sensed regions of degrading systems, the ability to maintain high throughput and fault tolerance in modern manufacturing systems depends on the ability to predict the quality characteristics of the products output by the system [15]-[17]. Commonly, quality measurements are performed with additional and often costly equipment and resources [17], which in most cases makes it economically, or sometimes physically impossible to obtain measurements of every product leaving the system. Instead, only some products end up being measured (sampling of products) and if unacceptable behavior occurs in-between samples, significant losses occur and failures may go undetected for prolonged periods of time.

These issues are highly impactful, especially in modern high tech manufacturing. This led to significant research in mathematical models that estimate quality variables using the continuously collected equipment signatures, without physically having to take the quality measurements. This is the well-known paradigm of Virtual Metrology (VM)

in manufacturing [17]-[20], utilized for quality control in many industries. Outside the scope of manufacturing, similar methods have been implemented for estimating un-measurable quantities under the concept of Virtual Sensing (VS) [21]-[23], with the main difference being the fact that VS does not take into account periodic measurements of the quantities being estimated and thus equivalence between the two concepts is not complete. Therefore, VS is outside the scope of this dissertation.

Since all major manufacturing industries deal with degrading systems continuously, investigating ways to predict future performance and estimate quality variables online is valuable to saving much energy and materials in the future. Advancing and integrating predictive condition monitoring methods into the VM concept in order to enable an integrated performance prediction and quality control methodology for modern manufacturing systems is the focus of this doctoral thesis.

1.2 Research Objectives and Challenges

The main objective of this research is to develop an integrated performance prediction and quality control methodology for manufacturing systems that can concurrently forecast, with high accuracy and computational speed, the behavior of distributions of the sensory signatures characteristic of the system condition, as well the quality of the products output by that system. Incorporated into this work are the following contributions:

1. A fast, analytically tractable algorithm for predicting long-term behavior of signatures describing the performance of a degrading dynamic system.

2. An accurate, dynamic VM model for estimating quality variables of the products output by the degrading manufacturing system.
3. Analytical approach for characterizing the propagation of uncertainty of sensory signature predictions through the dynamic VM model.
4. Application of the performance prediction and quality control methods to a real semiconductor fabrication process.

Figure 1.2 illustrates the concept and the objectives of this dissertation. The diagram shows a manufacturing system with its outputs continuously being monitored via residuals obtained through comparison of the current system behavior with the model of normal/desired operation. If designed correctly, the system will emit residuals that cluster around the origin and remain within some physically inspired or statistically determined tolerances when operating normally. It is assumed that the designed system has observed residual trajectories for the known degradation patterns either from physical models, experimentation, or field testing. If the system is recently built, the residual sets for common faults can be gathered during the initial operating periods between maintenances.

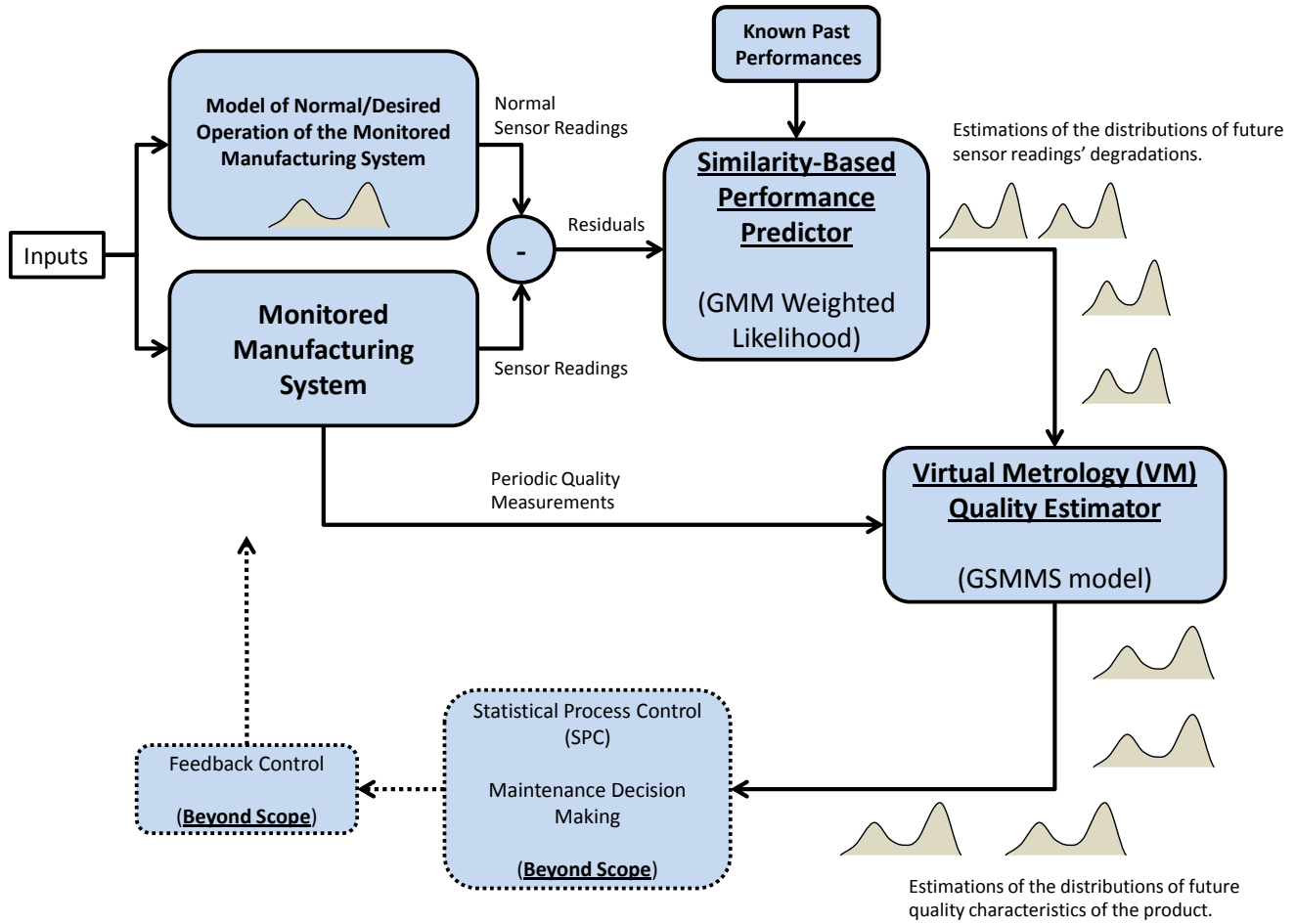


Figure 1.2: Flowchart demonstrating integrated performance prediction and quality estimation methodology developed in this work.

The first objective is accomplished by using a Similarity-Based Performance Predictor, recently reported in [24]. This algorithm continuously compares the dynamics of the currently forming trajectory of sensory readings indicative of system condition to the degradation trajectories seen in the past in order to predict distributions of those sensor readings in the future. The second objective is pursued via a VM method based on the estimation of quality measurements using a novel data driven model [25]. Objective

three is accomplished through the pursuit of analytical methods for estimating future distributions of quality variables by propagating the performance prediction uncertainties through the data driven VM model.

There are many challenges with the integrated VM modeling and prediction concept outlined in this thesis. The first challenge is to develop a suitable prediction method for non-stationary, non-Gaussian time-series, which can give accurate long-term predictions of the sensor signatures of the degrading manufacturing system in a computationally efficient way. In order to achieve this, one should take advantage of sensory time-series corresponding to various degradations and maintenances observed throughout the life of the manufacturing system. The second challenge is to formulate a VM model suitable for non-linear dependencies and non-stationary noise characteristics of the underlying relationship between equipment signatures and quality measurements, which is at the same time computationally feasible for online application in the manufacturing process. Furthermore, the method should be able to determine when the predictions coming from the model are not accurate, by recognizing previously unseen input patterns from the degrading system. Finally, the challenge of integrating the performance prediction methodology with the VM methodology is met by the development of a tractable method that does not compromise predictions though its assumptions on the uncertainty evolution and is still computationally feasible. This integration allows one to concurrently predict equipment signatures and quality measurements into the long-term using analytically tractable techniques that enable online and adaptive evaluations and predictions of equipment conditions and outgoing product quality.

1.3 Outline of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 presents a review of the relevant literature on prediction of time-series, quality estimation, and uncertainty propagation in dynamic systems. In Chapter 3, the methodology for similarity-based performance prediction is shown in detail along with results of applying this new methodology to an actual manufacturing process. Chapter 4 discusses the VM quality estimation methodology and results based on a locally tractable modeling framework applied to the same manufacturing process, while Chapter 5 describes the methodology and results for the integration of performance prediction and quality estimation using Monte Carlo uncertainty propagation. Finally, Chapter 6 details the scientific contributions of the doctoral research, and possible future work.

Chapter 2

Review of Time-series Prediction, Quality Estimation, and Uncertainty Propagation in Dynamic Models

As mentioned in Chapter 1, this doctoral dissertation focuses on integrating methodologies for predicting the long-term performance of degrading systems and estimating the quality characteristics of the products being produced by that system. As in the majority of the existing CBM literature, the system condition will be associated with signals emitted by the monitored system, in which case the performance prediction amounts to the problem of prediction of time-series of the relevant sensor readings. Hence, it is necessary to review the state of the art in time-series prediction. This review is given in Section 2.1. Also, tight control of quality is essential to ensure the desired performance of advanced manufacturing systems. Predicting quality measurements in manufacturing via the Virtual Metrology (VM) paradigm is one of the crucial methodologies facilitating advanced process control and an extensive review of the state of the art VM is given in Section 2.2. Finally, due to the stochastic nature of real world systems, uncertainty will inherently exist in the performance predictions and propagation of these uncertainties through the VM model is essential for accurate prediction of the quality variables using predicted equipment signatures. Hence, Section 2.3 gives a review of uncertainty propagation methods in stochastic dynamic models.

2.1 Time-series Prediction Methodology

Under the framework of condition-based monitoring (CBM) of dynamic systems which are degrading from their desired or designed performance, the condition of a monitored system needs to be deduced from the available sensor readings [14]. For example, small cracks in pipes and vessels of a chemical plant cannot be directly measured and their presence and severity need to be inferred from the strategically distributed sensing of flows and pressures across the system [7]. Generally, the CBM paradigm relies on relating the degradation process in a dynamic system with a set of features that can be extracted from the available sensor readings. In that context, degradation prediction boils down to prediction of the time-series of sensory features indicative of the system condition, such as vibration levels, forces, thermodynamic states, electrical states etc... [7]. Prediction of the time-series of features then allows one to predict future degradation patterns, fault modes, remaining useful life (RUL), or probabilities of unacceptable behavior of the system over time.

Numerous techniques for time-series prediction exist, involving various levels of assumptions, accuracy and computation complexity. One can separate these techniques into three general types: physics-based models, linear data-driven models and non-linear data-driven models.

Physical models predict the future behavior of a system by utilizing first-principle theories based on the fundamentals of physics and scientific theory [26]. However, identification of constituent equations and model parameters is not a trivial task for any real world system. In addition, for manufacturing equipment and other complex systems, there are sometimes thousands of interacting components with highly variable geometries

and uncertain forces affecting the system [7]. These conditions and requirements bring inherent problems to physical models, such as long computation times due to a large number of states, numerical instability, and inaccuracies due to a large number of estimated parameters and geometries. One can, however, still use a detailed physical model to see how certain parameters shift over time or how disturbances affect the system, and then simplify the model for predictive maintenance purposes [26]. Even though an accurate underlying physical model that captures relevant degradation phenomena is always the preferred option, various linear and non-linear data-driven models must be used in cases where the underlying performance is uncertain and/or requires too much computation to obtain accurate predictions from high fidelity physics-based models (which is the most frequent situation).

Linear data-driven prediction techniques include multiple-linear regression [27], least-squares regression [27], auto-regressive moving average (ARMA) models [28], and Kalman filters [29]. These methods work well for short-term predictions due to the assumptions that the data is generated from a linear system in the presence of stationary noise processes. Unfortunately, any real process with non-stationarities or non-linearities in the time-series will cause the prediction errors obtained using linear data-driven methods to increase drastically.

Non-linear parametric time-series prediction techniques also include numerous methods, such as non-linear regression [30], fuzzy ARMA modeling [31], [32], Bayesian curve fitting [33], wavelet methods [34], support vector machine [35], [36], probability density functions [33], [36],[37] and neural networks based techniques [38]-[40]. These methods have been shown to give better results in terms of modeling and predicting non-

stationary and non-linear time-series. However, compared to linear methods, they usually require much more computational effort in order to identify the time-series model, as well as to obtain prediction results from that model.

For both linear and non-linear data-driven approaches, selection of an appropriate model form is an important and difficult task. As an example, predictions of probability density functions (pdf) of future signatures can be accomplished by parameterizing them using Gaussian mixture models [37] or kernel density estimation [33], [36]. Zivkovic followed that approach in [36], offering a recursive finite mixture model algorithm for predicting multidimensional features that automatically selects the number of components in the mixture model. Bayesian curve fitting (Gaussian process) gets around this by allowing a distribution of solutions to be generated around the data structure [33], [41]. Yan et. al. [42] used a Markov model framework with a discrete number of states in order to predict remaining useful life of a car rotor. Sun et. al [43] used a non-linear state space model driven by output data and sequential Monte Carlo (SMC) for demonstrating performance prediction with a gas turbine simulation. Each one of these methods has their own advantages and drawbacks for particular problems.

Neural network based prediction methods represent an established family of non-linear data-driven time-series prediction approaches, with applicability to modeling and prediction of non-linear and non-stationary time-series [38]. The topology of interconnections among neurons, number of layers in the network, the number of neurons in each layer, and the type of transfer functions between neurons define the structure of these essentially non-linear parametric models, and commonly this structure is determined via *ad hoc* selection by the user. In this framework, each neuron has a transfer

function (linear, Gaussian, sigmoidal, tangential, etc. [38]), with parameters of the neural function and inter-neural weights usually determined via a gradient descent based algorithm [39].

Recurrent Neural Networks (RNNs) are the most common non-linear time-series prediction techniques in manufacturing due to their ability to model a variety of systems that cannot be modeled with first principles and their incorporation of internal dynamics into the model [44]. RNNs use internal delayed feedback links to take into account temporal dependencies in the data, which enables them to approximate a wide class of non-linear dynamic systems [40]. The gradient descent based training algorithms of these networks however have certain problems when dealing with long-term time dependencies, which can limit their accuracy. Also, choosing the number of layers, number of neurons, and feedback structure is still an active area of research [38].

Ferriera [44] gives a review of various neural networks leading up to the RNN, as well as the different learning algorithms for the RNN. He shows that with respect to a semiconductor deposition process, an RNN with a nonlinear ARMA structure performed better than an output error structured RNN. Furthermore, it was shown that training using an extended Kalman filter methodology works better than dynamic back propagation. The possibility of using a Kohonen self-organizing map (SOM)¹ neural network and local linear models within the SOM to identify areas of validity are mentioned for future comparison to RNNs. Bushman et. al. [45] showed that RNNs are more accurate than least squares linear models when applied to plasma etch semiconductor equipment, while additionally being fast enough for the manufacturing process and as for long-term

¹ A type of neural network that directly quantizes the observation space in an optimal manner.

predictions, Wen et. al. [46] demonstrated that RNNs are more accurate than feed forward neural networks on simulated non-linear time-series and rotating machinery in manufacturing environments. Xu et. al. [47] showed that RNNs train much faster than feed forward neural networks, while giving comparable accuracy when applied to predicting the cutting force of milling operations. Yu et. al. [48] develops confidence bounds on an Elman RNN for prediction of boring process signatures and showed its advantages over other models in maintenance decision making.

As for modifications to the basic RNN structure, Tian et. al. [49], [50] demonstrated that modifying the RNN with an additional context layer using self-feedback of delayed network outputs and output errors can increase accuracy over a fully connected RNN in the prediction of gear conditions. Finally, Yang et. al. [51] showed that an RNN does not work well for a non-stationary thermo-elastic system and that an integrated RNN (IRNN) approach should be used to eliminate any first-order differences between the current and previous input/output time-series.

All of these works demonstrate the universal applicability of RNNs over linear and other neural network methods. However extensive training is almost always needed, long-term forecasting can be inaccurate, and extrapolation outside of the known training data can be inaccurate [52]. It is commonly known that quality of the data and model training has a high impact on discriminating the performance accuracy between non-linear models, such as RNNs. Therefore, having a more tractable way of incorporating the historical knowledge into the model is desired.

Similarity-based methods for time-series prediction represent another powerful family of non-linear data-driven time-series prediction methods. These methods make

forecasts using a similarity metric of the currently observed time-series realizations with historical records of prior realizations of those time-series. In terms of performance prediction in condition-based maintenance, these approaches predict the future degradation states of the monitored system using time-series corresponding to past degradation patterns and fault modes of the monitored system. Mahalanobis and Euclidean distances [27] have been used extensively for finding the similarity between vector trajectories in the multidimensional space in which the time-series resides [14], [53]-[57]. Similarity based approaches are of particular importance in manufacturing applications. Namely, in manufacturing environments, systems experience multiple degradation cycles, failures, and maintenance events in their lifetimes [14], [58]. Signatures extracted from sensor readings over the life of a monitored system yield vital historical databases that can be used to predict the performance of the upcoming operations based on the similarity of prior operations [14], [58].

Liu *et al.* [14] used similarity measures of past time-series trajectories and heuristically inspired methods to weight the forecasts toward the most similar time-series trajectories observed in the past. This method was shown to have much lower long-term mean prediction errors than ARMA and RNN models. Wang *et al.* [56] offer another example of similarity-based time-series prediction methods. The authors accomplish time-series prediction using Euclidean distance measures and optimized alignment of the currently observed signature patterns with the past patterns of the time-series. The method was demonstrated in RUL prediction based on signatures emitted by turbo fans and was shown to improve the long-term predictions over an existing exponential curve fitting procedure. Yu [59] demonstrated a similarity-based technique on bearing

monitoring using overlaps of Gaussian mixture models, which greatly simplified the computation efforts. You et. al. [60] developed a proportional hazards model based on Weibull distributions for similarity comparisons in a simulated degradation process. While the aforementioned methods achieve long-term accuracy and mathematical transparency, it is obtained at the cost of longer computation times, which hampers prediction in very highly dimensional spaces, and the prediction of fast degradation processes.

The review of available literature implies that long-term prediction of time-series with complex dynamics and noise characteristics can be accomplished utilizing some similarity measure between the current and past time-series behavior, even though the benefits of such approaches come with a prohibitively high computation cost. The goal of the method pursued in this research is to achieve long-term prediction accuracy of performance signatures by incorporating the similarity of historical degradation processes, while at the same time achieving a level of analytical tractability that accelerates the process of postulating and updating such similarity based predictions. Specific details of the methodology for similarity based performance prediction pursued in this work are given in Chapter 3. One should note that the work reported in Chapter 3 is based on a recently published journal paper [24].

2.2 Quality Estimation using Virtual Metrology

The quality characteristics of a degrading manufacturing machine are commonly estimated online with models that estimate the quality measurements taken during inspection using only the equipment process data taken during operation, which bypasses

the need for physical measurements of the product quality variables. As mentioned in Chapter 1, this paradigm in manufacturing is commonly referred to as VM (Virtual Metrology), due to its ability to predict metrology outputs using statistical and/or dynamic models based on the easily obtainable equipment sensor readings. Implementing a proper VM system allows manufacturing process control with fewer investments in the physical measurement equipment and less time necessary for inspection. This enables higher equipment availability, more effective maintenance, higher productivity, and improved quality.

Figure 2.1, which is adapted from [20], illustrates the VM concept, where the upper part depicts the monitor measurement scheme adopted by many designs. The quality measurements are replaced by the VM model estimates based on the equipment sensor readings emitted during operation. VM quality predictions are occasionally compared with the actual measurements in order to ensure the VM model accuracy. The models are usually trained either using data corresponding to the normal system operation, specially designed experiments, or using a physical model, if available [17].

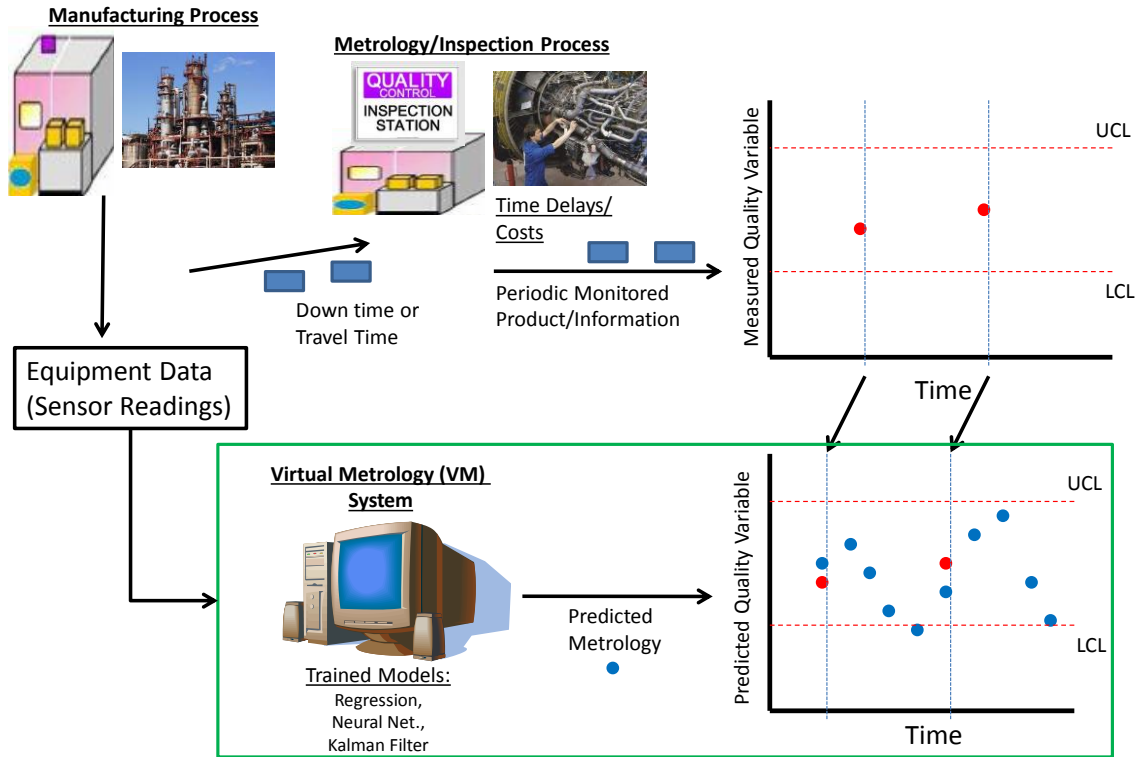


Figure 2.1: Concept of replacing quality measurements with statistical models utilizing equipment data (Virtual Metrology) (modified from [20]).

Various statistical approaches for VM have been proposed in the literature. As in the case of time-series prediction, data driven models represent the majority methodology. VM has been applied to broad areas such as machine coordinate predictions [61], [62], predictions of the distribution of agricultural products [63], and estimations of material properties [64], [65]. For example, Hughes et. al. [61] utilized a virtual metrology metal frame for coordinate measurements where relative displacements of reference features were determined periodically and used to self-calibrate the frame. Also, Chanal et. al. [62] was able to optimize kinematics machine tools by taking into account surface defects without looking at the entire tool positioning errors as in the case of most coordinate positioning techniques. Albiero et. al. [63] was able to estimate the

longitudinal distribution of seeds being distributed with a central ring seeder in order to ensure quality of agricultural products. Suh et. al. [64] identified optimal process parameters in order to estimate solar array material properties and performance characteristics, while Yang et. al. [65] was able to estimate the friction coefficient of Cr-Al-C thin films using regression networks find optimal process parameters. In addition to VM, as mentioned in the introduction, vast applications are found under the term Virtual Sensing (VS) [66]-[79], which utilize similar techniques without accepting periodic measurements of the estimated variables.

Due to the potential economic and safety benefits to be gained from measurement costs and faulty production throughput, most of the current VM applications are concentrated in semiconductor manufacturing [80], [81]. The processes utilized in the manufacture of microprocessors are highly complex, uncertain, and energy intensive, bringing about many incentives to implement data-driven VM models into the semiconductor manufacturing facilities [82]. Measuring the physical wafer metrology parameters is performed on monitor wafers that are periodically selected by sampling in the production equipment for each lot (usually 1 wafer per 25 or more wafer lot) [80], [81]. When the equipment has faults and the abnormality is not detected in time, many defective wafers could have been produced before the next measurement, resulting in large amounts of wafer scrap and cascading errors. In addition, work by Khan et. al. [17], [18] and Kang et. al. [83], [84] has shown the benefits that VM can bring to process control in semiconductor manufacturing. The available VM literature is focused on the common processes used in the semiconductor manufacturing industry - mostly chemical vapor deposition (CVD) and etch processes [82]. In the realm of VM for CVD, relevant

metrology variables, such as film thickness, film uniformity, particle counts, and reflectivity are estimated using the process data obtained during the deposition process, such as temperatures, pressures, flow rates, radio-frequency (RF) power in the plasma generation system, etc. As for etch processes, metrology variables, such as etch rate, particle count, surface resistance, critical dimensions, and endpoint are estimated using similar process sensors as CVD, along with additional optical emission information for plasma tools.

The vast majority of work in VM focuses on increasing prediction accuracy using so-called global models², which can be classified either as linear models, such as multivariate linear regression (MLR) [33], and partial least squares (PLS) regression [87], or non-linear models, such as back propagation (feed-forward) neural networks (BPNN) [33],[88], radial basis neural networks (RBNN) [33],[88], Gaussian process regression (GPR) [33],[106], and support vector regression (SVR) [33],[103],[104].

Linear models, such as MLR and PLS, are the most commonly used VM models due to their tractability, and ease of implementation and adaptation to variable operating regimes of the underlying machine. These types of models predict well the dynamics of systems that behave linearly within the region in which they are trained, but extrapolation beyond the training set can certainly hurt predictions, if the underlying dependencies are non-linear. In the area of CVD, Olsen et. al. [102] describe updating PLS models with a moving window to predict wafer thicknesses in six chambers, which yielded better accuracy than what was achieved without windowing and updating the model. More recently, Bernard et. al [101] compared PLS with a decision tree method and showed

² Models for which the regression from tool signatures (VM inputs) to metrology variables (VM outputs) is expressed via a single model valid throughout the input space.

comparable accuracy in film thickness VM on a plasma enhanced CVD (PECVD) process. As for VM on etch tools, Lee et.al. [85] compared three linear regression models against a BPNN, showing that a PLS type model can be just as accurate as BPNN when predicting etch rates on an experimental dataset. Work by Lynn et. al. [107] examined a weighted PLS implemented within a moving window, determining that updating a weighted PLS was slightly more accurate than global PLS or using a moving window alone.

Besides linear regression models, neural networks are also very commonly used models in VM, primarily due to their well-known ability to approximate non-linear dependencies. Generally speaking, neural networks comprise numerous interconnected computational nodes representing non-linear functions of relevant inputs, and weighted sums of the output of those nodes are used to obtain a functional relationship for modeling the desired output. Within VM, two types of neural networks have been studied: BPNN, and RBNN. BPNNs are trained by back propagating prediction errors until the network parameters converge, while RBNNs use a specific form of the node activation functions, which enables their training using relatively tractable and fast matrix manipulations. In CVD processes, Cheng et. al. [90] developed a dual phase VM scheme that uses one model to estimate the quality of the current lot of wafers, while at the same time using another model to predict the quality of the next lot using the available historical quality measurements. A BPNN was used as the conjecture model, while an exponentially weighted moving average (EWMA) time-series model was used to predict the quality of the next lot, facilitating significant improvements in VM estimates. In [93], Su *et al.* introduce a reliance index for a BPNN based dual phase VM. This index was

used to determine situations when an abnormal set of inputs (equipment signatures) is observed and was calculated using the distance of the currently observed inputs away from the Gaussian distribution modeling “normal” tool operation. When these distances were unusually high (reliance index was unusually low), the VM model could not be trusted anymore and new physical measurements would have to be taken. A similar index can be found in Cheng et. al. [91] and was used for the same purpose. More recently, Hung et. al. [96] also adopt the dual phase VM scheme and report that a RBNN with adaptable weights is faster and more accurate than BPNN. As for neural networks being used in etch processes, Zeng et. al. [105] report that BPNN is more accurate than PLS for a variety of moving window updating implementations.

An obvious weakness of all linear regression based VM methods is their inability to deal with nonlinear dependencies between equipment signatures and metrology variables. In addition, the major issues related with neural networks for modeling of general non-linear dependencies, such as the choice of the network topology³, selection of the type of the activation function, initialization of network parameters, as well as problems associated with extrapolation⁴, and over fitting⁵, are inherited in the realm of NN based VM. The abovementioned drawbacks of linear regression models and neural networks motivated further research and recent introduction of GPR [106], SVR [103] and Kalman filter [109] [110] based models into VM. GPR and SVR methods provide a non-linear modeling frameworks with more tractability and transparency than neural

³ Number of nodes, their allocation into layers and connections between the nodes and layers.

⁴ Utilizing and believing neural network outputs for inputs significantly different from those observed during the training process.

⁵ Understanding when the neural network captured the underlying system dynamics and when further training is not needed.

networks, while still capturing non-linear relationships with excellent accuracy [104]. On the other hand, the Kalman filter brings measurement feedback into the model, which can provide better estimation than static models.

The work by Lynn et. al. [106] can be seen as the earliest example of GPR based VM. They report that GPR gives better accuracy than PLS and BPNN, while at the same time yielding a global estimate of model uncertainty. As for SVR, Chou et. al. [103] combine a genetic algorithm (GA) with the SVR to search for the best number of inputs and model parameters. They show better accuracy than both BPNN and RBNN on a CVD process, though it is acknowledged that retraining the SVR model can be slower in the VM context. Additionally, Purwins et. al. [104] compare SVR with different linear regression models and show that SVR is more accurate and is more robust in adapting to process changes than all the models they benchmarked against. Finally, Gill et. al. [110] implement a multivariate Kalman filter based VM, predicting etch rate and resistance in an etch process. On a dataset consisting of a few hundred wafers, their method yielded more accurate VM results than VM based on MLR, PLS, recursive PLS and time-series moving average models [28]. It should be noted that though it is dynamic in nature, a major drawback of the Kalman filter based approach to VM remains the fact that the underlying model does not take into account non-linear dependencies, or non-stationary noise characteristics in the underlying VM dependency.

Besides the obvious emphasis in the relevant literature on achieving highest possible accuracy of VM models, another major focus of the research is enabling VM models to detect new (unforeseen) operating conditions and adapt to them. Namely, new operating conditions occur as the underlying machine degrades and/or process parameters

drift over time. This could lead to changes in the dependencies between equipment signatures and the corresponding metrology (VM inputs and outputs), necessitating new physical measurements of the relevant quality variables, as well as VM model updates. In order to accomplish this, various forms of moving window based adaptations [102], [94], [108] or on-line adaptations [110] were implemented within the framework of global regression or dynamic models, where a single functional relationship is used to represent the VM dependency over the entire input space. The main complication with this type of models is that all of its model parameters are potentially affected whenever the system enters a regime outside of where the model is trained, thus potentially disturbing the model parameters even for inputs for which the model was already established. Such a global adaptation approach that potentially affects all model parameters hampers the adaptation process by requiring significant amounts of data and measurements for a new operating regime to be modeled well. Furthermore, and perhaps even more importantly, as the process drifts back into the previously visited (“normal”) regimes, which could happen due to process control or maintenance interventions, the disturbed VM model needs to readjust to this condition, even though before the excursion that led to initial model adaptations, the VM model may have already been trained for such inputs. This obviously lowers VM accuracy and requires more physical measurements to be taken.

An alternative approach to model adaptations can be sought within the so-called divide and conquer modeling paradigm, in which a set of interconnected, locally valid models is used to approximate an input-output dependency [150],[161]. Such a VM modeling paradigm enables localized model adaptations, only in areas near the inputs that are deemed unusual, without perturbations of model parameters far away from those

unusual inputs. Though divide and conquer models received significant attention recently, the only example of such an approach in the VM context can be found in [95]. In this paper, authors offer a VM method for critical dimensions of a photo-lithography process, using a combination of an MLR and a BPNN model. During model building process, an MLR and a BPNN VM model are built to model the entire training dataset and then for any given input vector, the model that yielded better VM prediction at that point was deemed valid for that input. VM predictions for new inputs were obtained by either selecting a model corresponding to the nearest input vector from the training set (either an MLR or a BPNN), or as a weighted sum of MLR and BPNN VM predictions, with weights determined using Mahalanobis distances from the inputs observed in the training set. This is indeed a VM modeling framework consisting of 2 distinct models, combined either in the piecewise manner, or as a sum of weighted contributions from the two models, similar to what is seen in recent advances in dynamic systems modeling literature [143]. For both processes they analyzed, the authors report accuracy improvements for the multiple model scheme over sole use of either of the constituent VM methods.

Nevertheless, the number of models in the divide and conquer scheme reported in [95] was obviously limited to 2 and no adaptation scheme for the resulting VM scheme was discussed. One difficulty that could be foreseen if model adaptations were indeed pursued in [95] is that adaptation of a BPNN model is in no way trivial, since its form is inherently complex and nonlinear. A more common situation when divide and conquer models are used in the general modeling literature is the use of simpler, analytically tractable constituent models. A degree of tractability in local models is important in order

to facilitate easy local adaptation of model parameters in situations when the model validity is poor, while model adaptations should not perturb VM performance for inputs for which a model was already established. These problems will be addressed in Chapter 4 of this doctoral dissertation using the recently introduced growing structure multiple model system (GSMMS) paradigm [143]. This divide and conquer modeling approach solves the problem of describing complex non-linear dynamic dependencies by employing a set of simple (linear) models, each of which is responsible for approximating the target dynamics only within a limited domain of inputs. As will be seen in Chapter 4, the locally tractable character of GSMMS models will enable overcoming the problems mentioned above when GSMMS is used as the foundation for VM modeling.

2.3 Uncertainty Propagation Methods for Dynamic Systems

Even though much work has been done in these two separate areas, a natural link between performance prediction and VM quality estimation has never been established. Let us elaborate on the benefits of that link in some detail. Namely, changes in equipment signatures indeed are related to the condition of the tool, but that relation is not direct, which means that time-series prediction as is done in a large portion of predictive monitoring research alone does not address the need of predictive CBM. On the other hand, the condition of the monitored system is directly visible in the quality measurements in the sense that for highly sophisticated systems, like semiconductor manufacturing machines, equipment faults show up as non-conforming products, while producing good quality products implies a good condition of the tool. Hence, a VM that realizes the link between equipment signatures and its condition can be used again as the

link between predicted equipment signatures and the predicted equipment condition. In essence, the opportunity is to feed predicted equipment signatures through a VM model to obtain predicted distributions of quality characteristics of the product and thus predict likelihoods of unacceptable tool behavior in the future. The main challenge is that both the prediction of future equipment signatures and the VM model bring their own uncertainty into the prediction of the quality measurements. Characterizing the uncertainty obtained from the predictions of the future equipment performance and understanding how that uncertainty is propagated through the VM model, which itself is uncertain, represents the main problem that needs to be solved to realize this potentially very lucrative vision of predictive VM.

Predicting the evolving behavior of uncertain inputs through dynamic systems is an extensive area of research [111]-[118]. In almost any modeling situation, including the predictive VM scheme proposed in this paper, there will be uncertainty in the inputs, initial conditions, and parameters of the model. When the inputs or initial conditions of the system are probability density functions (pdfs), the system becomes stochastic and the state of a stochastic dynamic system, x , must be characterized by its time dependent pdf, $p(t, x)$ [111]. Knowing the time evolution of this pdf is important for quantifying the uncertainty of the underlying dynamic system in a future state.

Numerous fields have dealt with the problem of pdf evolution through nonlinear dynamic systems with stochastic excitation or uncertain initial conditions. Applications include determining of the response of engineering structures under random excitation [119], fluid dynamics [120], [121], gas dispersion estimation [122], [123], and 3D visual tracking [124].

For general nonlinear systems, the exact description of the transition pdf is provided by the Fokker Planck Kolmogorov Equation (FPKE) [125]-[127]. Analytical solutions exist only for stationary pdfs and are restricted to a limited class of dynamic systems [126]. Thus, researchers are looking actively at numerical approximations to solve the FPKE, for more general problems [117] , [126]. The Finite Difference Method (FDM) and the Finite Element Method (FEM) have only proved practical for two and more recently three dimensional systems due to the exponentially growing computational load induced by meshing in a multidimensional space..

Several other numerical techniques exist in the literature to approximate the evolution of a pdf through a dynamic model, including Monte Carlo (MC) simulation based methods [119], [128]-[130], particle filtering [123], [124], [131], [132], Gaussian closure [118], [113], [133], polynomial chaos [116], [120], [121], linearization [111], [116], [134], and stochastic averaging[135],[136]. MC simulation based estimates are used often to deal with analytically intractable stochastic problems, but they require extensive computational resources and effort, and become increasingly infeasible for estimating high-dimensional outputs involving a large number of samples. As for the other methods listed above, they are similar in several respects, and are suitable only for linear or moderately nonlinear systems, because the effect of higher order dynamics can lead to significant errors. Furthermore, all these approaches provide only an approximate description of the uncertainty propagation problem by restricting the solution to a small number of parameters - for instance, the lowest-order moments of the sought pdf.

Several analytical methods can be found in literature, using fairly strong assumptions on the underlying model and distribution being propagated. Terejanu et. al.

[117], [118], [123] have done novel work in the field of uncertainty propagation through high fidelity dynamic models. The work is concerned with improving the approximation to the forecast density function using a finite Gaussian mixture and a parameter updating scheme. The methods give analytical tractability, efficient computation, and enough complexity to handle large scale non-linear systems. However, the mathematics behind propagating the Gaussian mixture weights remains a challenge. Besides Gaussian mixtures, Terejanu et. al. [117], [118], [123] have tried polynomial chaos, particle filters, and extended Kalman filter [29] to model propagation of uncertainties through dynamic models.

Based on the review of the relevant literature, a need for an integrated approach that can predict the performance of complex manufacturing systems and at the same time estimate future quality characteristics of the system has been seen and will be developed in this doctoral research. In this work, the VM model will have a single output to be predicted. Therefore, the Monte Carlo method will be appropriate to obtain estimates of the propagated uncertainty in a timely manner. If this is not feasible for reality, possible future avenues will be to apply the appropriate assumptions to the pdf based on the model and pursue an analytical method. Chapter 5 will discuss the results obtained in predictive VM using Monte Carlo simulation of the predicted tool signatures.

Chapter 3

Analytical Approach to Similarity-Based Performance Prediction of Monitored Systems

3.1 Methodology Overview

As mentioned in Section 2.1, similarity-based time-series prediction methods are capable of modeling non-linear, non-stationary, and multi-dimensional time-series more accurately than traditional neural network, or regression models. Therefore, a similarity-based methodology will be pursued in this work in order to achieve the goal of accurate long-term prediction of performance signatures of degradation processes, while at the same time achieving a level of analytical tractability that accelerates the process of postulating and updating such similarity based predictions.

Let us first introduce the following terminology that will be used throughout the work:

- The term *feature vector* is used to indicate the signatures extracted from the raw sensors mounted on the monitored system that are known to characterize the condition of that system. Evolution of these signatures is then indicative of system degradation and their behavior needs to be predicted.
- The term *cycle* is used to indicate a single operation by the system, emitting a single feature vector. This can be any manufacturing operation or single use of a

product. Repeated cycles degrade the system, causing the signatures to evolve and ultimately lead to maintenance events.

- The term *run* is used to indicate the time interval between two consecutive maintenance events. These maintenance events can be component replacements, repairs, cleaning, etc. Thus, a time-series of feature vectors in a past run represents a particular degradation trajectory known from historical data.

The method described in this Chapter is relevant for any dynamic system that has existing records of signal feature trajectories corresponding to the degradation patterns/failure modes observed in the past or physical models of degradation dynamics. It is assumed that the degradation process is described by a time-series of feature vectors extracted from the sensors relevant to the behavior of the dynamic system. For example, Figure 3.1 illustrates a multitude of possible degradation trajectories using features from a simulation of a hypothetical non-linear two-output system [7]. The complexity of the degradation and failure modes of a system, such as wear, aging, pitting, cracking, corrosion and other mechanisms [7], often makes the resulting time-series of features indicative of system degradation non-stationary and non-linear in their dynamics. As mentioned in Chapter 2, similarity-based methods with an underlying non-linear parametric model can be effectively utilized to account for non-linear, non-stationary degradation dynamics and thus result in high long-term prediction accuracy. This involves the comparison of the large number of historical runs and combining that information into a dynamic model that can predict the current run's degradation process.

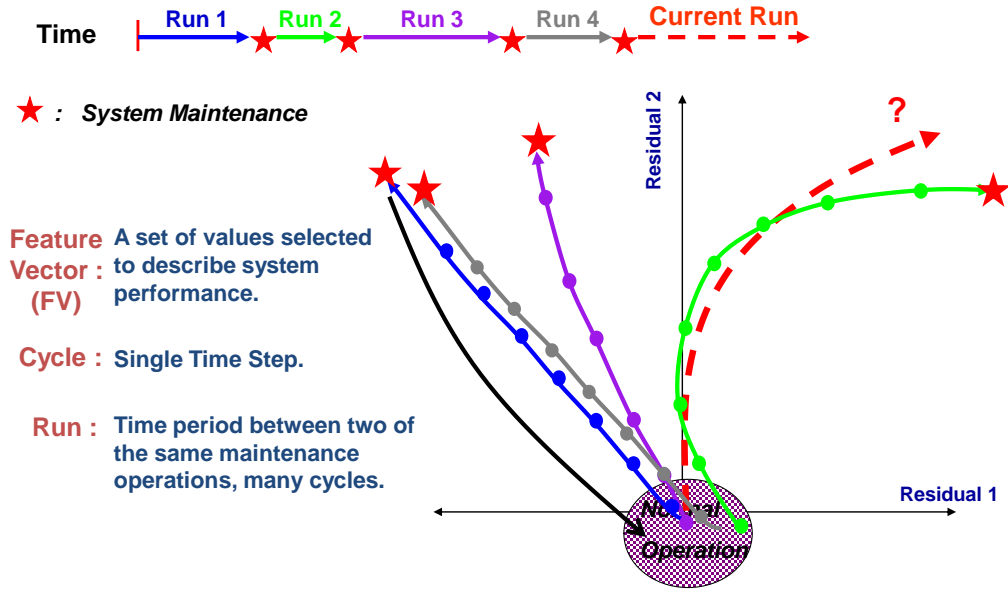


Figure 3.1: Conceptual illustration of degradation dynamics for a system with different fault modes, monitored using two features (residuals).

The method described in this work follows the aforementioned paradigm and uses pdf-s representing feature vectors from the previous runs as the models for evaluating similarities between the newly observed trajectory of signatures and those observed in the past. The pdf-s are approximated using Gaussian mixture models (GMM) due to their ability to model any distribution within a desired accuracy, given enough Gaussian components [37]. Figure 3.2 conceptually describes the newly proposed time-series prediction algorithm. Right after a maintenance operation (i.e. just before a new run starts), the only information known about the run that is about to start are feature vector realizations observed during the previous runs of the monitored system. At each cycle, GMMs of feature vectors corresponding to that cycle in the previous runs can be formulated. As the current run progresses, feature vectors from more and more cycles are

observed and similarity measures between those feature vectors and all feature vectors corresponding to that cycle in past runs can be evaluated. Consequently, at cycle number i , we can observe the set of vectors \vec{s}_c^* ; $c \in \{0,1,2,\dots,i\}$, composed of similarities between feature vectors observed at cycle number c , with feature vectors observed at cycle c in all previous runs ⁶. These similarity measures can then be used to skew the GMMs of feature vectors corresponding to future cycles of the current run (cycles $i+1$, $i+2$...) towards feature vectors from previous runs that in the past cycles showed more similarity with the current run. Thus, as time progresses and more and more signatures are collected during the current run, the feature models shift toward the most similar runs observed in the past. When the current run is completed, it can be incorporated into the library of previous runs, thus enabling continuous learning as the system progresses through its lifetime.

⁶ Each vector \vec{s}_c^* will have as many elements as there are past runs that performed c cycles.

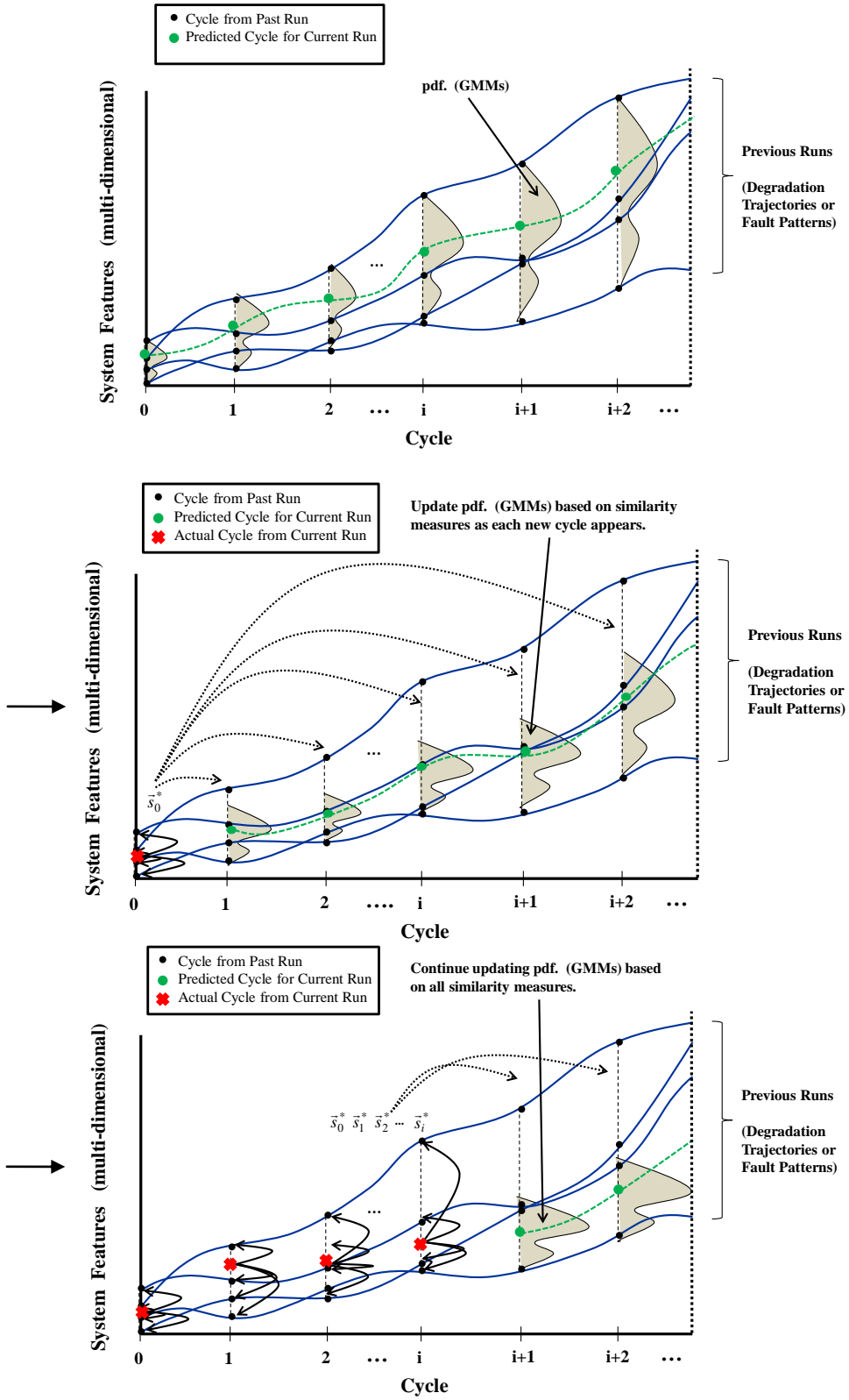


Figure 3.2: Steps for the similarity-based prediction algorithm described in this work.

Details of the prediction method conceptually described above will be given in the remainder of this Chapter, which is organized as follows. Section 3.2 discusses the calculation of the similarity measures used in updating of the GMMs. Sections 3.3 and 3.4 review the method of modifying the log-likelihood function of a pdf in order to handle weighted data points and also discusses the modifications to the well-known Expectation-Maximization (EM) algorithm [138] used to fit a GMM to the library of past feature vectors, while incorporating the similarity measures. The following is based on the recently submitted journal paper [24].

3.2 Concept of Similarity Vectors

Let us assume that the monitored system has just performed cycle i of the current run and let $\bar{x}_{i,current}$ denote the corresponding feature vector consisting of D components (i.e., feature vectors are of dimensionality D). Given K previous runs in the historical database, one can compute distance measures, d_{ik} , between the current feature vector and the i^{th} feature vectors \bar{x}_{ik} from all previous runs $k, k \in \{1,2,3..K\}$. Following Liu *et al.* [14], in this work the Mahalanobis distance is used

$$d_{ik} = \sqrt{(\bar{x}_{ik} - \bar{x}_{i,current})^T \bar{\Sigma}^{-1} (\bar{x}_{ik} - \bar{x}_{i,current})} \quad (3.1)$$

where the scaling covariance matrix $\bar{\Sigma}$ is characteristic of the normal operation of the monitored system, i.e. estimated from feature vectors observed soon after maintenance operations [10]. Depending on the application and topology of the feature space, other

distance measures could be used [30]. The similarity s_{ik}^* of the i^{th} cycle of the current run with the k^{th} previous run can be expressed as

$$s_{ik}^* = \exp(-d_{ik}) \quad (3.2)$$

which gives values close to one when the Mahalanobis distance between the two feature vectors is small, and approaches zero as the distance grows large.

The K similarities can be combined into a vector $\vec{s}_i^* = [s_{i1}^* \ s_{i2}^* \ \dots \ s_{iK}^*]^T$, containing similarities for cycle i in the current run. Observations from Liu *et al.* [14] and our own experiments showed that the similarity vectors \vec{s}_i^* tend to be rather noisy and a filtering step is needed before those similarities can be used for updating of predictions. In this work, an exponentially weighted moving average (EWMA) [139] with smoothing factor λ was used as a simple way to combine all of the similarity vectors $\vec{s}_0^* \dots \vec{s}_i^*$ into an *overall similarity vector*, \vec{s}_i , that can be used in predicting future cycles of the current run (i.e. feature vectors for cycles $i+1, i+2, \dots$). The overall similarity vector \vec{s}_i contains filtered information about similarities between all the past cycles of the current run and corresponding cycles in the past runs of the monitored system, with similarities corresponding to the most recent cycles being more emphasized by the EWMA filter. Based on the aforementioned concepts of feature comparison between the past and current degradation trajectories (runs) of the monitored system, we can now describe the use of the similarity vectors \vec{s}_i for prediction of future feature vectors for the current run.

3.3 Weighted Likelihood Estimation of Mixture Gaussians with Similarity Vectors

Once again, let us assume that the monitored system just completed cycle i of the current run and that a filtered vector of similarities, \bar{s}_i , has been obtained using methods described in Section 2.2. At any future cycle (i.e. cycle, $i+1$, $i+2$, ...), a multi-dimensional GMM, $f(\bar{x})$, of feature vectors will be pursued in the form [37]

$$f(\bar{x}) = \sum_{m=1}^M w_m N(\bar{x} | \bar{\mu}_m, \bar{\Sigma}_m) \quad (3.3)$$

where M is the total number of Gaussian components in the GMM, w_m is the mixture weight corresponding to the m^{th} component of the GMM and

$$N(\bar{x} | \bar{\mu}_m, \bar{\Sigma}_m) = \left(\frac{1}{(2\pi)^{D/2} |\bar{\Sigma}_m|^{1/2}} \right) \exp \left[-\frac{1}{2} (\bar{x} - \bar{\mu}_m)^T \bar{\Sigma}_m^{-1} (\bar{x} - \bar{\mu}_m) \right] \quad (3.4)$$

denotes the m^{th} Gaussian component with mean $\bar{\mu}_m$ and covariance matrix $\bar{\Sigma}_m$. The reader should be reminded that D is the dimensionality of the feature vectors and thus represents the dimensionality of the GMMs describing feature distributions at each cycle.

Let us assume we need to fit a GMM to the observed feature vectors \bar{x}_k ; $k \in \{1, 2, \dots, K\}$. It is easy to show that the log-likelihood $\ln \Pr(X | \theta)$ of $X = \{\bar{x}_1 \dots \bar{x}_K\}$ is

$$\ln \Pr(X | \theta) = \sum_{k=1}^K \ln [f(\bar{x}_k)] \quad (3.5)$$

In order to fit a GMM, this function is commonly maximized by a gradient ascent algorithm that seeks parameters $\theta = \{\bar{\mu}_m, \bar{\Sigma}_m, w_m\}$, $m = 1 \dots M$, that maximize the likelihood (3.5) for the given data X [33], [37].

The log-likelihood function of a pdf can always be modified with a prior distribution if one has information about the system behavior. Since we have prior information about system behavior from the previous runs, we will use this opportunity to skew the GMMs of future cycles of the current run towards the corresponding cycles of the previous runs that resemble the current run the most.

Weighted Likelihood Estimation (WLE) is a Bayesian estimation methodology based on modifying the likelihood function of a pdf with some weighting distribution, vector, or observation-dependent function that is *a priori* known [140], [141]. Various applications in statistical analysis, machine learning, medicine, and gambling have used these techniques to obtain better prediction accuracy from numerous probabilistic model forms[140]-[142]. One of the most common formulations of WLE is to assign each feature vector a unique weight α_k based on the prior information about the data (higher α_k meaning higher prior probability for a given observation). The log-likelihood function of the data X for this type of a WLE takes the following form

$$\ln \Pr(X | \theta)_{WLE} = \sum_{k=1}^K \ln \left[(f(\bar{x}_k))^{\alpha_k} \right] = \sum_{k=1}^K \alpha_k \ln \left[\left(\sum_{m=1}^M w_m N(\bar{x}_k | \bar{\mu}_m, \bar{\Sigma}_m) \right) \right] \quad (3.6)$$

In our context, *the weight α_k is the k^{th} component of the overall similarity vector \bar{s}_i at cycle i* . In such a way, the similarities with previous runs can be used to modify the log-likelihood function in a way that emphasizes the most similar previous runs and *vice*

versa, thus increasing the prediction accuracy. Based on the definitions of the WLE of a GMM, one can now state the necessary modifications for the gradient-based EM algorithm used for fitting the GMM parameters.

3.4 Expectation-Maximization Algorithm Modifications for the Weighted Likelihood Estimation of Mixture Gaussians with Similarity Vectors

The so-called Expectation Maximization (EM) algorithm is commonly utilized for fitting GMM parameters to a data set [37], [138]. The EM algorithm is an iterative method for estimating the maximum likelihood solution for a set of parameters in a statistical model. In this subsection, a modification of this algorithm is introduced, which will enable the inclusion of similarities of the current run with the previous runs into the estimation process.

Let us assume that a feature distribution for any given cycle in the current run is composed of M Gaussians⁷ and that we have the corresponding initial guesses for the M mean vectors $\bar{\mu}_m$, and M covariance matrices $\bar{\Sigma}_m$ [37]. The procedure for finding the GMM components $\theta = \{\bar{\mu}_m \quad \bar{\Sigma}_m \quad w_m\}$, $m = 1 \dots M$ consists of two steps. In the first step, referred to as the Expectation calculation, the likelihood that a given feature vector in the dataset X belongs to each GMM component is evaluated. The next step, referred to as Maximization, updates the GMM parameters in order to maximize the likelihood function

⁷ Literature has shown works that find optimal numbers of components in mixture models in an unsupervised manner, and therefore this assumption is not particularly restrictive [33], [36].

of the data, given the probabilities from the Expectation step [138]. The algorithm repeats the Expectation and Maximization steps until convergence, with possible convergence criteria being the number of Expectation-Maximization iterations, bounds on the change in the log-likelihood function, or the change in GMM parameters [138].

In this work, the WLE log-likelihood function (3.6) replaces the standard formulation and the EM process will be consequently modified by maximizing (3.6) with respect to $\theta = \{\bar{\mu}_m, \bar{\Sigma}_m, w_m\}$, $m = 1 \dots M$. For a given set of GMM parameters, the Expectation step can be performed for a GMM in standard fashion, by taking the weighted sum of the different Gaussian component contributions to each feature vector

$$\Pr(m | \bar{x}_k, \theta) = \frac{w_m N(\bar{x}_k | \bar{\mu}_m, \bar{\Sigma}_m)}{\sum_{m=1}^M w_m N(\bar{x}_k | \bar{\mu}_m, \bar{\Sigma}_m)} \quad (3.7)$$

where $\Pr(m | \bar{x}_k, \theta)$ is the probability that feature vector k is produced by the Gaussian component m . In order to obtain the maximum likelihood parameters, $\hat{\theta}_{ML}$, one must maximize the log-likelihood function with respect to the model parameters:

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}} \{ \ln \Pr(X | \theta)_{WLE} \} = \underset{\theta}{\operatorname{argmax}} \left\{ \sum_{k=1}^K \alpha_k \ln \left[\sum_{m=1}^M w_m N(\bar{x}_k | \bar{\mu}_m, \bar{\Sigma}_m) \right] \right\} \quad (3.8)$$

Based on the procedure proposed in [37] for the EM based identification of GMM parameters, the EM algorithm for the modified likelihood function (3.6) yields the following iterative procedure for finding GMM parameters:

At each iteration l , the current solution for the GMM parameters $\theta^l = \{\bar{\mu}_m^l \quad \bar{\Sigma}_m^l \quad w_m^l\}_{m=1,2,\dots,M}$ is transformed into the next iterative approximation of the solution $\theta^{l+1} = \{\bar{\mu}_m^{l+1} \quad \bar{\Sigma}_m^{l+1} \quad w_m^{l+1}\}_{m=1,2,\dots,M}$ according to formulae

$$\bar{\mu}_m^{l+1} = \frac{\sum_{k=1}^K \bar{x}_k \alpha_k \Pr(m | \bar{x}_k, \theta^l)}{\sum_{k=1}^K \alpha_k \Pr(m | \bar{x}_k, \theta^l)} \quad (3.9)$$

$$\bar{\Sigma}_m^{l+1} = \left\{ \frac{\sum_{k=1}^K \alpha_k \Pr(m | \bar{x}_k, \theta^l)}{\left[\sum_{k=1}^K \alpha_k \Pr(m | \bar{x}_k, \theta^l) \right]^2 - \sum_{k=1}^K [\alpha_k \Pr(m | \bar{x}_k, \theta^l)]^2} \right\} \sum_{k=1}^K (\bar{x}_k - \bar{\mu}_m^{l+1})(\bar{x}_k - \bar{\mu}_m^{l+1})^T \alpha_k \Pr(m | \bar{x}_k, \theta^l) \quad (3.10)$$

$$w_m^{l+1} = \frac{\sum_{k=1}^K \alpha_k \Pr(m | \bar{x}_k, \theta^l)}{\sum_{m=1}^M \sum_{k=1}^K \alpha_k \Pr(m | \bar{x}_k, \theta^l)} \quad (3.11)$$

where

$$\Pr(m | \bar{x}_k, \theta^l) = \frac{w_m^l N(\bar{x}_k^l | \bar{\mu}_m^l, \bar{\Sigma}_m^l)}{\sum_{m=1}^M w_m^l N(\bar{x}_k^l | \bar{\mu}_m^l, \bar{\Sigma}_m^l)} \quad (3.12)$$

The Expectation and Maximization steps are repeated until a convergence criteria is met [138]. Possible convergence criteria are bounds on changes in the likelihood function or bounds on changes in the model parameters. The modified EM algorithm for

estimating GMM parameters based on the WLE formulation (3.6) is summarized in the pseudo code shown in Figure 3.3.

In this work, at any cycle i , the WLE-modified EM algorithm is used to rapidly estimate GMM parameters for feature vectors corresponding to cycles $i + 1, i + 2, \dots$. When the feature for cycle i in the current run is observed, the similarity vector \vec{s}_i is updated using the procedure described in Section 3.2 and the modified EM algorithm described in this section is invoked to rapidly update the GMMs for the cycles $i + 1, i + 2, \dots$

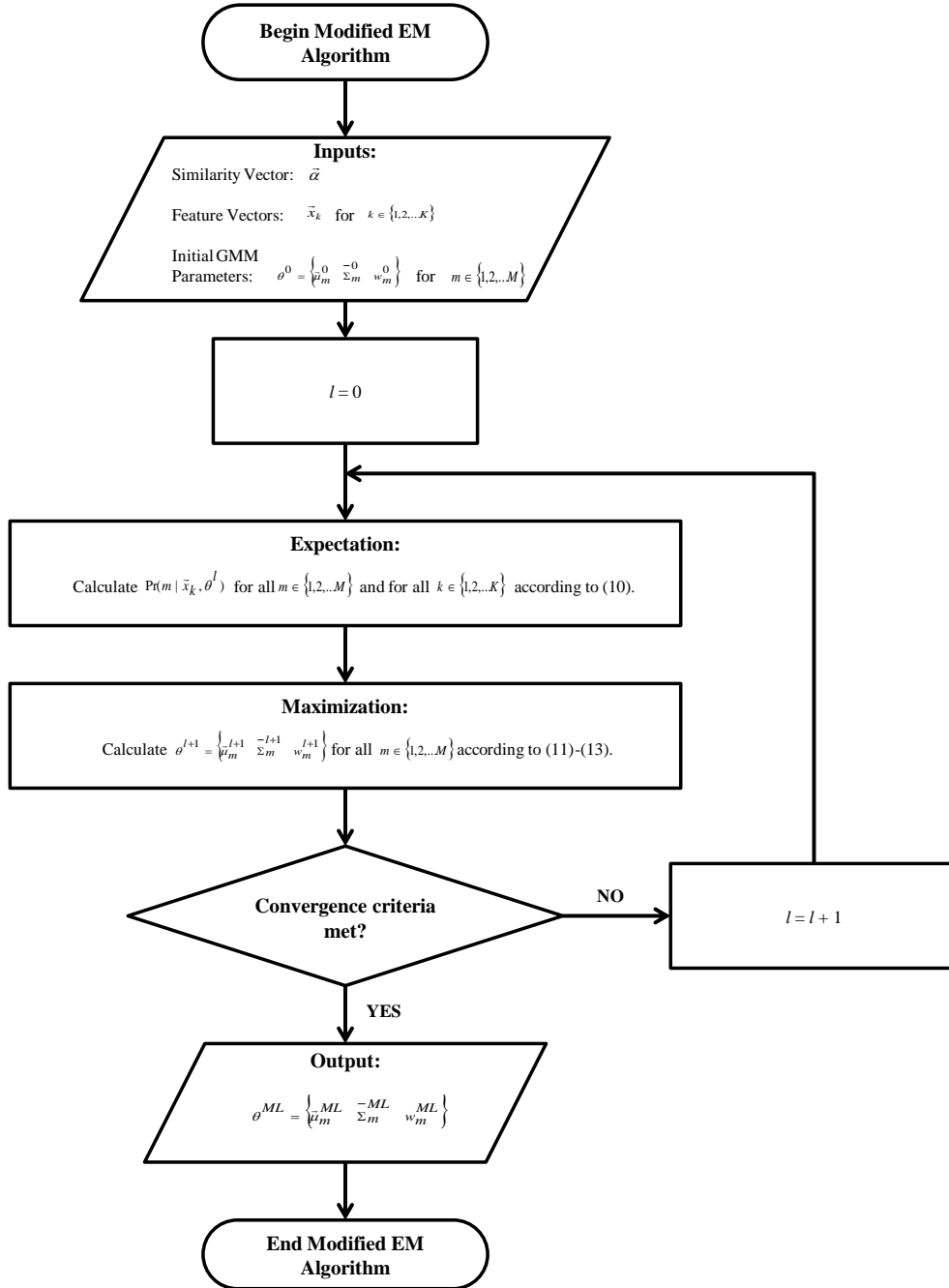


Figure 3.3: Pseudo-code for the modified EM algorithm utilized in this work for GMM fitting.

3.5 Results of Similarity-Based Performance Prediction

3.5.1 Experimental System Description

The experimental system to be utilized in this work is a Plasma Enhanced Chemical Vapor Deposition (PECVD) tool commonly found in the semiconductor manufacturing industry. PECVD tools are used for depositing thin films onto silicon wafer substrates, which is one of the crucial steps in manufacturing of microelectronic circuits and solar cells. It is the most common method for producing conductors and dielectrics with excellent film growth properties necessary for small chip components [82]. Inside a PECVD tool chamber, reactive gases pass over silicon wafers and are absorbed onto the surfaces to form a thin layer. The gases are excited through radio frequency (RF) electrical power that creates energetic plasma used to deposit the film on the wafers. The plasma state allows the reaction to take place at lower temperatures, more suitable for large silicon wafers. Ultimately, many stacked layers of conducting and insulating films with etched patterns between them (forming thousands of microscopic electrical components) form an integrated circuit [82].

3.5.2 PECVD Tool Subsystems and Sensors

A general PECVD tool is composed of a reaction chamber, radio frequency (RF) plasma generation system, gas delivery system, wafer load locks, and a robotic arm to carry wafers to and from the tool. Figure 3.4 shows a diagram of the main components of a PECVD tool. The RF matching network for generating plasma is shown on the top-left of the diagram. The high frequency energy is sent through two matching network capacitors (load and tune capacitors) that control the power delivered to the chamber. By

varying their capacitances, the capacitors try to tune the impedance of the circuit to the load impedance of the chamber and thus deliver maximum RF power to the gases in the chamber [82]. The RF energy excites the flowing gas into the plasma state necessary for lower temperature depositions [82]. The gas delivery system is depicted on the right of Figure 3.4. It consists of mass flow controllers (MFCs) for each gas used in various depositions. Gas flows over precise time intervals to ensure processing of specific thin-film recipes. A control valve (bottom of Figure 3.4) controls the chamber pressure and evacuates deposition gases from the chamber. Temperature controlled top and lower chamber plates enclose the chamber and the walls are heated to minimize on-wall deposition, as well as to speed up the reaction during the automatic cleaning process.

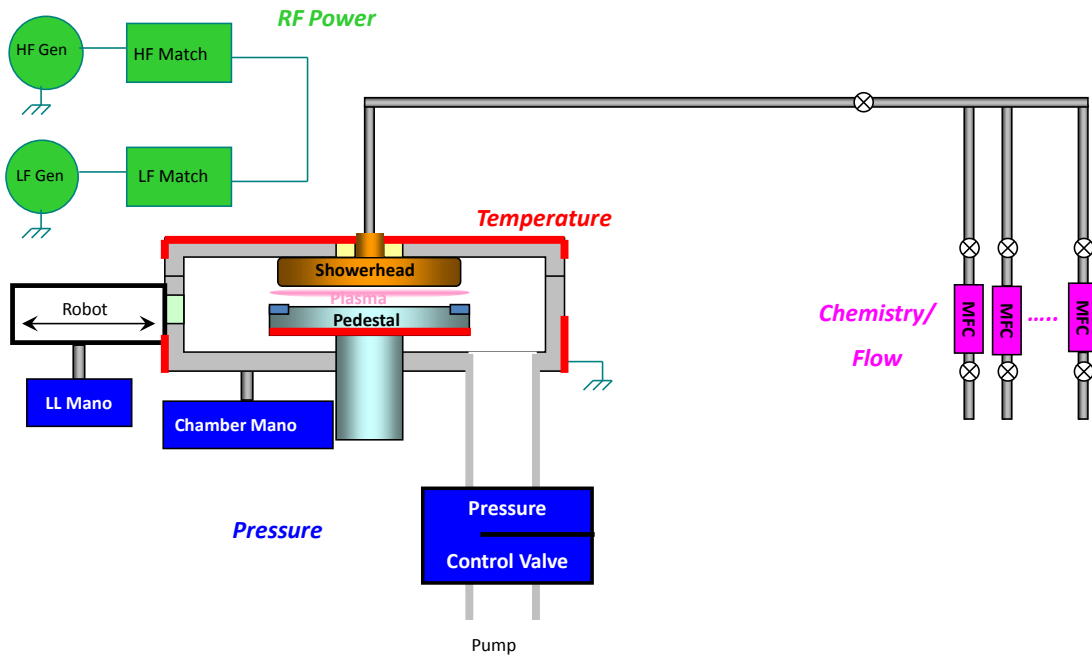


Figure 3.4: Diagram representing the various components that make up the PECVD tool.

The PECVD tool used in this study was a standard 300mm wafer tool with numerous in-place sensors measuring the physics of the process (to ensure real-life applicability of our study, only the standard on-board sensors were analyzed and no additional sensors were considered). The signals used in this study were the RF power characteristics (forward, load, and reflected power), voltages for the capacitors of the RF matching network, MFC flow rates, top plate temperature, chamber temperature, pedestal temperatures, chamber pressure, and the pendulum valve angle totaling 14 sensors. RF power, flow rate, chamber temperature, and chamber pressure are all controlled at desired recipe set points during deposition. All sensor readings were concurrently collected using a 10Hz sampling rate, which is an order of magnitude higher than the prevalent 300mm fab standards.

3.5.3 Tool Operation and Maintenance Schedule

Various chemical compounds can be deposited using PECVD tools. Silicon Nitride (Si-N) and Silicon Dioxide (Si-O₂) are some of the most common thin films deposited in these tools, even though other compounds can be used, depending on the conductivity, mechanical and reliability requirements on the film [82]. Silane (SiH₄) and Tetraethyl Orthosilicate (TEOS) are common reactants used to produce these films.

In addition to the deposition cycles that contribute to the process of chip-making, PECVD tools in semiconductor manufacturing facilities also perform automatic in-situ cleaning programs after a predetermined total film accumulation limit (corresponding to approximately 25-100 wafers, depending on the film thickness). The usual way of performing the in-situ cleans is by flowing plasma-excited Fluorine (F⁻) into the chamber

to eat away films deposited on the tool surfaces. These cleans are performed periodically in order to bring the tool back into a lower state of degradation. Unfortunately, this tool-cleaning procedure is imperfect since residual films can be left in parts of the chamber and at the same time, some tool surfaces can also be etched away during the process. This results in a long-term degradation of the tool, which over time leads to the production of wafers with noticeable defects, unless preventive maintenance (PM) actions are undertaken. Thus, besides the short-term accumulation drifts caused by successive wafer depositions and remedied via in-situ cleans, one can also observe a long-term drift of the tool condition as numerous in-situ clean cycles are executed.

Figure 3.5 illustrates the scheduling for different levels of PECVD tool cleaning and maintenance. An automatic in-situ clean program is performed after depositions on a predetermined number of wafers (approximately, every 25-100 wafers). The tool loops through these programs using different chemistries and physical parameters until the fixed-time maintenance schedule requires a long-term PM intervention (approximately, every 25,000-100,000 wafers). This long-term PM action usually consists of a physical wipe down of the chamber as well as repairs and replacements of various critical tool components.

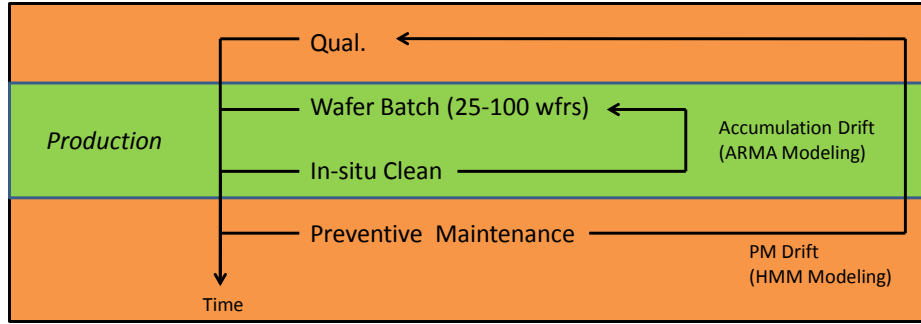


Figure 3.5: Production cycle of the PECVD tool.

The proposed similarity based performance prediction algorithm was recently demonstrated using a PECVD tool dataset coming from a major semiconductor manufacturing fab. The details of feature extraction and algorithm demonstration are described in the next few sections.

3.5.4 Results from Experimental Study

Extraction of signal features that are the most descriptive of machine performance is one of the key elements of CBM [10]. Useful information can be gathered from extracting dynamic features by utilizing them in prediction models for capturing the features' time-series performance. Predictions of system degradation, remaining useful life, and sudden developing failures are all examples of critical information that are found through dynamic features extracted from raw data. In previous work, numerous features from multiple sensor readings were extracted, including dynamic features such as rise-time, overshoot, and steady state values, along with statistical features, such as mean value, variance and range [33]. These features were chosen due to their commonplace

use in dynamic systems and stochastic process theory. The 40 features extracted from PECVD tool sensor readings in this study are summarized in Table 3.1.

Table 3.1: Features extracted from the data analyzed in this study.

Signal	Signal Features				
Top Plate Temperature	Mean	Minimum	Amplitude		
Chamber Temperature	Mean	Minimum	Amplitude		
Pedestal 1 Temperature	Mean	Minimum	Amplitude		
LF Forward Power	Steady State Error	Tune Time			
LF Load Power	Steady State Error	Tune Time			
LF Reflected Power	Steady State Error	Tune Time	Maximum		
HF Forward Power	Steady State Error	Tune Time			
HF Load Power	Steady State Error	Tune Time			
HF Reflected Power	Steady State Error	Tune Time	Maximum		
Load Capacitor Voltage	Steady State	Tune Time	Overshoot High	Overshoot Low	
Tune Capacitor Voltage	Steady State	Tune Time	Overshoot High	Overshoot Low	
Pendulum Valve Angle	Steady State	Maximum			
Process Chamber Pressure	Steady State Error	Rise Time	Overshoot	Minimum	
Liquid Flow Rate TEOS	Steady State Error	Rise Time	Overshoot		

Each feature is calculated from its corresponding sensor monitoring the tool, while these sensors were selected by the manufacturer to monitor the physics relevant to the processes. The features chosen in this study are extracted from signals corresponding to consistent processes (for example, depositions, pre-coats, in-situ clean, etc), by comparing the features of the same process throughout the tool operation. Signal features that are the most descriptive of tool condition and degradation were sought within this comprehensive feature set.

In order to perform the multivariate analysis of the numerous features extracted, the features were standardized to eliminate the physical units and thus make them

dimensionally homogeneous. Standardization of each feature was accomplished by subtracting its mean and dividing that difference by the standard deviation of that feature, where the mean and standard deviation were calculated from the data set considered to be representative of normal operation. Features with zero standard deviation during normal conditions were ignored.

The data utilized for the analysis corresponds to about 80 wafer batches (40 wafers per batch), with in-situ cleans performed between each batch. In this relatively short period, corresponding to a few weeks of production⁸, long-term feature drifts or sudden changes were not detected, and the analysis can be considered to address only the short-term accumulation drifts between in-situ cleans.

Sensitivity analysis to degradation between in-situ cleans was performed by applying linear discriminant analysis (LDA) to classes formed with the feature set obtained *just prior to the in-situ cleans* (the last 5 depositions just before each of the cleans in the training set) and *depositions just after the in-situ cleans* (the first 5 depositions just after each of the cleans in the training set). In total, there were 400 deposition cycles composing each of the classes which come from the beginning of the entire dataset. Table 3.2 lists the top 10 sensitive features along with the unit principal vector that points in the direction of most sensitive features obtained from the LDA.

⁸ Due to the proprietary nature of this information, we cannot be more specific.

Table 3.2: Results of the sensitivity analysis between pre and post In-situ clean features.

Principal Vector (w)	Feature Name
0.60	Load Capacitor Overshoot Low
0.41	Load Capacitor Overshoot High
0.40	Load Capacitor Steady State
0.27	HF Load Power Steady State Error
0.26	Top Plate Temperature Mean
0.25	Top Plate Temperature Minimum
0.22	Pedestal 1 Temperature Minimum
0.21	Pedestal 1 Temperature Mean
0.13	LF Reflected Power Tune Time
0.06	Process Pressure Minimum

Table 3.2 indicates that the Load Capacitor features make up the top sensitive features. This is plausible because the load capacitor signal sees a different chamber impedance before and after the in-situ cleaning, and hence its tuning characteristics change in order to match the impedance of the chamber and deliver the maximum RF power to the plasma. Several temperature related features also appear to be sensitive to degradation between in-situ cleans. This also matches the engineering intuition since progressive depositions leave increasingly thicker films on the pedestal, showerhead, and chamber walls of the tool, thus gradually changing their thermal emissivity, which is mirrored in the corresponding temperature features.

Figure 3.6 shows three distinct evolution trajectories (runs) of two of the top sensitive features listed in Table 3.2 (the Load Capacitor Steady State Voltage and the Top Plate Temperature mean). From this figure, one can see that the trajectories through feature space can show vastly different behavior as the runs progress. In addition, Figure 3.6 also shows obvious recoveries at the beginning of the runs, which all begin in similar areas of the feature space before progressing into different directions, indicating that various degradation patterns occurred during the course of many inter in-situ clean runs observed in this data set. Thus, Figure 3.6 illustrates that this dataset is a typical situation that necessitates the use of similarity based prediction methods.

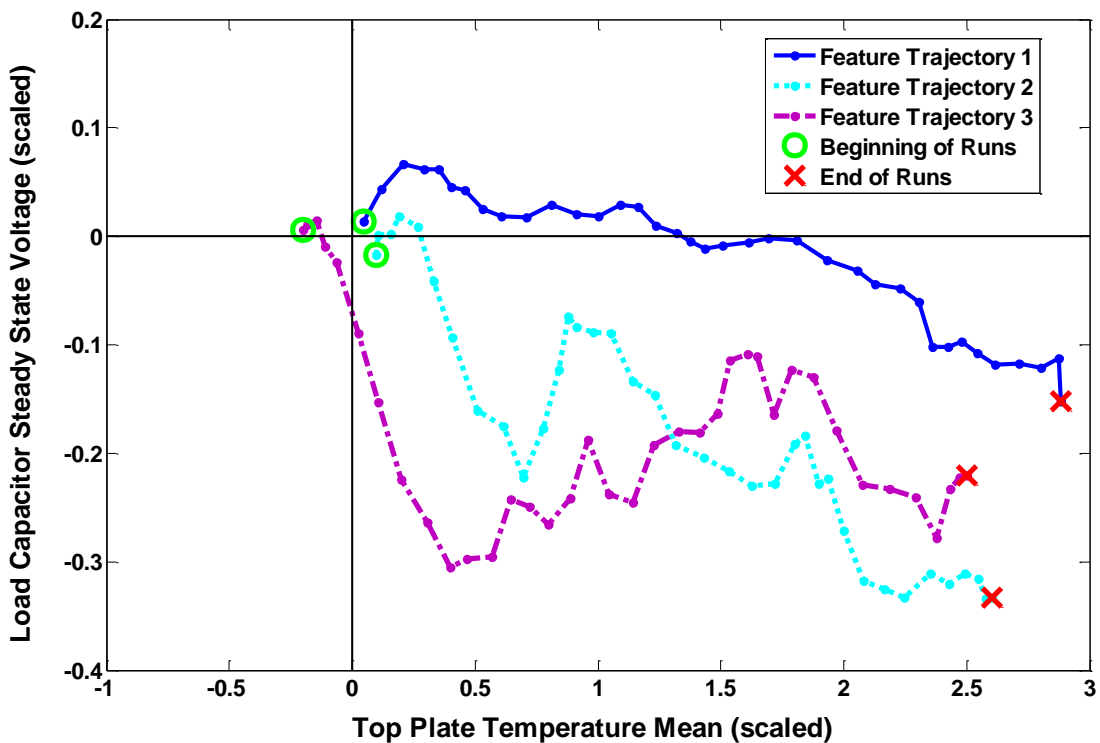


Figure 3.6: Various degradation paths for three select runs of the PECVD tool using two sensitive dynamic features.

The newly introduced similarity-based time-series prediction methodology was used to predict the behavior of the features listed in Table 3.2. The new method is compared against two other time-series prediction models - the standard Auto-Regressive Moving-Average model based prediction [28] and a recently introduced prediction method also based on the use of similarity matrices [14]. Mean squared errors and computation times associated with each method are used as comparison metrics. Signatures corresponding to wafer depositions from initial 30 runs were used to form the historical database of past runs, while signatures from another 40 runs were used for testing of the prediction methods. Within each test run, predictions of feature vectors were made up to 35 cycles (wafers) ahead, after which the squared prediction errors and times associated with the computation of all the predictions were averaged over the 40 test runs. Figure 3.7 shows the results for mean square errors (averaged over the scaled 10 dimensional feature space) and computation times for each of the three algorithms for the case when predictions were made starting at cycle (wafer) 11 in each of the test runs. Figure 3.8 presents the same metrics for the case when predictions were made starting at cycle 20 in each of the test runs.

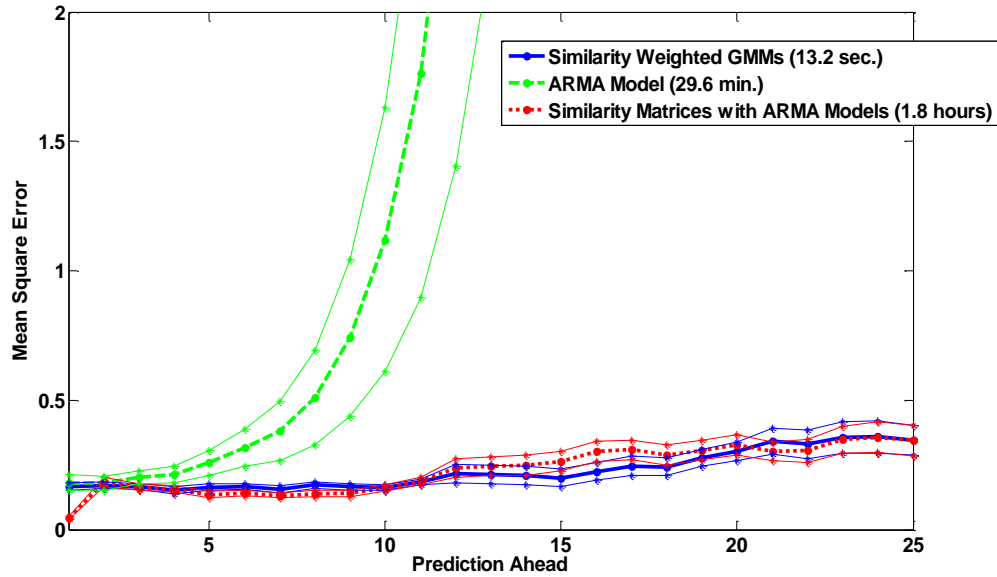


Figure 3.7: Mean squared errors and computation time for the three prediction algorithms, starting at *cycle 11* in each of the 40 test runs and predicting 25 cycles (wafers) ahead. Results were averaged over 40 test runs.

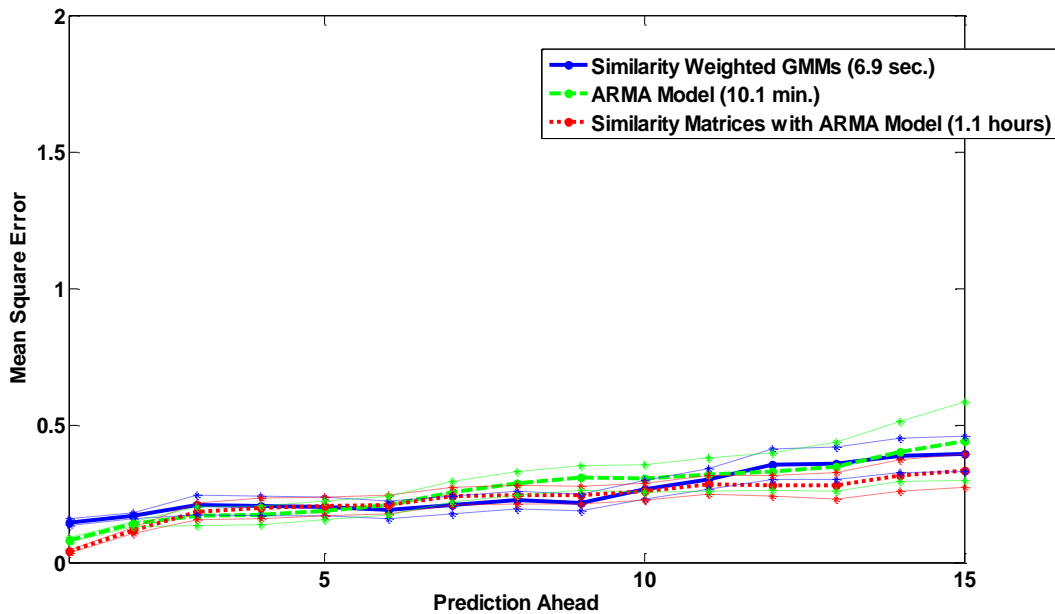


Figure 3.8: Mean squared errors and computation time for the three prediction algorithms, starting at *cycle 21* in each of the 40 test runs and predicting 15 cycles (wafers) ahead. Results were averaged over 40 test runs.

One can clearly see that ARMA prediction errors grow rapidly as the prediction horizon increases in Figure 3.7. The reason is that ARMA prediction uses only dynamics of the features observed in one specific run and does not fully exploit the rich information available in the previous run history. The newly developed method and the similarity matrix based approach from [14] both have much smaller errors and are comparable with each other in terms of prediction accuracy and confidence. However, the weighted GMM based algorithm introduced in this doctoral research takes an order of magnitude less time to compute the results than the other two methods (minutes instead of hours necessary to obtain prediction results for all 40 test runs). When predictions start from further ahead in the run, the prediction errors are comparable for all three methods, even though one can observe that the ARMA prediction errors start slightly drifting after predicting more than 10 cycles ahead. Once again, the ARMA and similarity matrix methods take an order of magnitude longer to compute the prediction results for all 40 test runs than the new algorithm. The computational advantage comes from the analytical character of the GMM based estimation of future feature distributions used in the new similarity based prediction method, which bypasses the need for Monte-Carlo sampling utilized in Liu et al. [14]. Thus, the new method incorporates the rich information from the past runs, but, unlike what is seen in [14], it does not do that at the expense of tremendous computational effort and sacrificing analytical tractability.

3.5.5 Conclusions of Similarity Based Prediction Results

In this Chapter, a novel time-series prediction algorithm capable of dealing with a long-term prediction of non-stationary multivariate time-series was presented. The

method is based on the concept of similarity-weighted Gaussian mixture models (GMMs) obtained via comparisons of signatures describing the current degradation process with those observed on the same machine/process in the past. It provides one with a natural way to derive the predicted feature distributions over time, which could be used to obtain information about the remaining useful life, predicted probabilities of failure, or unacceptable behavior. The new method was tested in predicting signatures extracted during the operation of an industrial Plasma Enhanced Chemical Vapor Deposition (PECVD) tool. The results showed that the newly developed prediction method yields noticeably smaller mean squared errors, compared with ARMA based prediction and comparable mean squared errors to another recently introduced similarity-based time-series prediction model. However, the analytical structure of the method computes the prediction distributions an order of magnitude faster.

An avenue for possible future work is potential grouping of similar degradation trajectories (runs) having similar evolutions of the time-series of sensory features, which will enable one to reduce the number of degradation trajectories that need to be kept in the historical database and used for predictions. In that context, a new run would be added to the library of past runs only after a degradation trajectory is observed that is “sufficiently different” from the ones seen in the past. Another avenue is to incorporate maintenances that do not bring the system back to a consistent state. A longer-term drift will be present if this occurs and will have to be taken into account. These problems are outside the scope of this research, but carry significant potential benefits.

Chapter 4

Virtual Metrology (VM) Quality Estimation using Local Dynamic Model Paradigm

4.1 Overview and Motivation

As discussed in Section 2.2, literature shows that various linear and non-linear models have been utilized in VM, including neural networks [65], regression models [104], Gaussian processes [106], time-series [100], and Kalman filter implementations [109]. In general, any model that can accurately and in a timely manner predict physical metrology measurements from the easily available equipment data will be a good candidate for VM. However, VM model characteristics such as model robustness to process variations, sensitivity to process variations in the presence of faults and assessment of the model confidence and validity, become increasingly important for the implementation of VM systems.

It can be concluded from the literature review that most of the published work utilizes models that do not incorporate process dynamics or metrology measurement feedback into the VM framework, as well as that the main focus is placed on prediction accuracy, which is evaluated on relatively small datasets. The current state of the art in VM is in need of models that can efficiently incorporate non-linear dynamic dependencies with non-stationary noise characteristics, as well as having the ability to know when the model is invalid because the underlying machine signatures show unusual

patterns for which the VM model was not trained. Most models mentioned previously do not have the ability to quantify the model validity over the entire operating space accurately, and a localized modeling approach will be pursued in this doctoral research to tackle this problem.

Since the increasing complexity of system dynamics often prevents one from building an accurate model based on first principles, data-driven approaches have been extensively employed for the modeling of complex dynamic systems [143]-[147]. As mentioned in Chapter 2, neural networks, such as multilayer perceptron (MLP) networks, radial basis function (RBF) networks, and recurrent neural networks (RNN), are probably the most extensively applied among techniques for the modeling of complex nonlinear systems due to their universal functional approximating capabilities [148]- [149]. Unlike feed-forward networks, such as RBF and MLP, which have limitations of identifying temporal dynamics in a time-series, RNNs take into account temporal dependencies through local or global internal feedback connections in the network, which enables a good approximation of a wide class of nonlinear dynamical systems [40]. However, the commonly used gradient descent algorithms used to train RNNs exhibit problems during training, such as having difficulty dealing with long-term temporal dependencies and over-fitting [148]- [150]. In addition, finding a suitable number of hidden neurons and appropriate RNN structure remains a challenging problem. Furthermore, even after a model is obtained, the residual errors between the model and the actual system still have to be properly interpreted for the purpose of anomaly detection and fault diagnosis. To achieve the sufficient detection and identification accuracy, one may need even more

sophisticated decision making algorithms to cope with modeling uncertainties, time varying process noise, etc... [151].

An alternative for modeling nonlinear dynamic system is to divide the whole system operation space into sub-regions, with a different dynamic model being valid in each of these sub-regions. In general, if the operation space is divided into the right number of properly placed and properly shaped sub-regions, local linear models in each region can approximate the underlying function to an arbitrary accuracy [143],[144]. In addition, local tractability of linear models is a highly alluring proposition for condition monitoring and control of systems modeled using such an approach. Nevertheless, it is obvious that proper partitioning of the operating space is crucial for success of this modeling paradigm and is one of the major challenges in the area of piecewise dynamic models.

A variety of method based on the multiple model systems framework have been proposed in the past for modeling of general nonlinear dynamic systems. Takagi and Sugeno [152] describe a model consisting of a number of implications that form a fuzzy partition [153] of the input space. The fuzzy partitions are often identified off-line or by trial and error, and consequently only a limited number of algorithms exist that can deal with both on-line structure and parameter learning. Johansen and Foss [154], [155] also adopted the multiple local model strategy using smooth interpolation functions between the local models. This work is based on the decomposition of the system's operating range into a number of smaller operating regimes, coupled with the use of simple local models to describe the system dynamics within each regime. However, the decomposition is not done in a structured manner.

Instead of dividing the operating space or input variables into fixed regions determined by the designer, vector quantization techniques, such as SOMs [156], have been proposed to directly partition the operation space through the Voronoi tessellation⁹ [143], [150], [157], [158]. However, in [150] and [157] the network topology and number of regions still have to be fixed in advance.

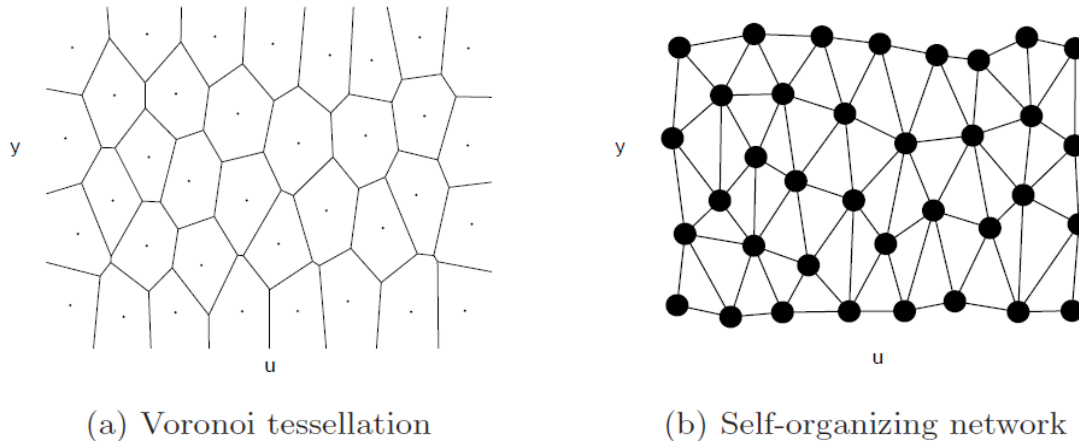


Figure 4.1: Partition of input-output space using self-organizing network; *a*) Voronoi tessellation and *b*) self-organizing network.

Recent developments in *growing* SOMs, such as growing neural gas [159] and growing cell structures [160], [161], impose fewer constraints on the network structure by incorporating growth and deletion mechanisms for SOM nodes and connections. Such networks can automatically determine the number of nodes once a stopping criterion, such as the maximum tolerable quantization error, is provided. Thus, growing SOMs are able to reflect the inherent structures of the data with fewer underlying assumptions than what is seen with conventional fixed structure SOMs.

⁹ Figure 4.1a illustrates an example of partitioning the input-output space into a set of SOM-induced Voronoi regions with disjoint interiors. The corresponding SOM is also shown in Figure 4.1b, where each node in the network is associated with a weight vector.

In this work, growing SOMs will be used for partitioning of the operation space into smaller regions and a least square algorithm will be used for estimation of local model parameters within each region. This effectively yields the GSMMS framework as the foundation of the VM model in which equipment and process signatures are used as GSMMS inputs to predict the corresponding metrology variables as GSMMS outputs. By taking advantage of the local modeling framework of GSMMS, one can accurately represent non-linear dynamic VM dependencies. Furthermore and perhaps more importantly, the local modeling character of the GSMMS will allow detection of abnormal equipment inputs or quality measurements that have not been observed in the training data, in which case the corresponding VM outputs should not be trusted. Finally, local model tractability and SOM growth can be used to efficiently update the GSMMS based VM model, as new equipment signatures and physical metrology measurements become available.

Figure 4.2 illustrates the analysis of local model residuals which can be used for discerning validity of the GSMMS VM model over the entire operating space. If the input vector for the VM model lands in a region in which confidence in the local linear model is low (variance of the estimated parameters is high), or it lands in one of the open regions far away from any input vectors observed during training, then the VM model outputs should not be trusted.

In the VM literature, only Gaussian process regression can provide this kind of model validity assessment. However, this method is not suitable for high-dimensional modeling (large number of input variables) and does not model dynamic dependencies [106], [107].

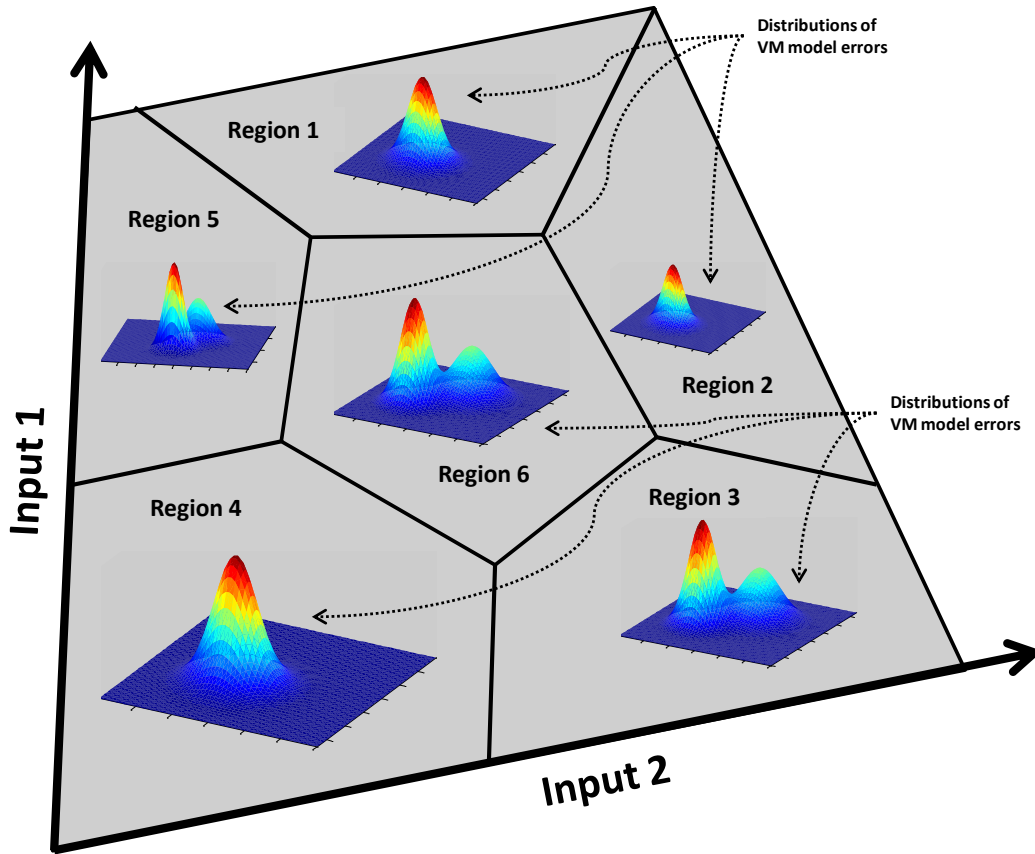


Figure 4.2: Illustration of localized model residuals for help in discerning model validity over the entire operating space.

The remainder of this Chapter is organized as follows. Section 4.2 describes the GSMMS methodology. Section 4.3 describes the results found from preliminary research conducted in VM and details scenarios of how a GSMMS-based VM methodology will be tested in this doctoral research.

4.2 Growing Structure Multiple Model System (GSMMS) Methodology

Following the multiple model strategy of Johansen et. al. [154], [155], let us describe the output of the system (estimate of the quality variable) $y(k)$ as

$$y(k) = \sum_{m=1}^M v_m(\vec{s}(k)) \vec{F}_m(\vec{s}(k)) \quad (4.1)$$

where

$$\vec{s}(k) = [\vec{u}^T(k), \dots, \vec{u}^T(k - n_b)]^T \quad (4.2)$$

is a vector consisting of input vectors¹⁰ $\vec{u}(k) = [u_1(k) \dots u_p(k)]^T$ at consecutive time samples k , while $v_m(\vec{s}(k))$ describes the validity of a local model $\vec{F}_m(\vec{s}(k))$ for the operating regime defined by the input vectors contained in $\vec{s}(k)$.¹¹

A variety of model structures can be utilized to describe local dynamics $\vec{F}_m(\vec{s}(k))$

. Notably, if the linear local model form,

$$\vec{F}_m(\vec{s}) = \vec{b}_m + \vec{a}_m^T \vec{s} \quad (4.3)$$

is used, where vectors \vec{b}_m and \vec{a}_m are model parameters, then the model parameters can be efficiently estimated by minimizing in the least square sense the modeling errors in the training set.

As for the validity function $v_m(\vec{s}(k))$, when it satisfies

¹⁰ Signatures extracted from sensors sensing the equipment and process parameters.

¹¹ Current and previous inputs, as well as previous outputs are usually used in the GSMMS framework [143]. However, in the context of VM, outputs (physical product measurements) are only measured sporadically, at a much lower rate than inputs, which is why vectors $\vec{s}(k)$ contain only the current and past inputs (equipment signatures from the current and past products).

$$v_m(\vec{s}(k)) = \begin{cases} 1 & \text{if } \vec{s}(k) \in V_m \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

where V_m , $m = 1, 2, \dots, M$ describes a disjoint partition of the operating space in which vectors $\vec{s}(k)$ reside, that is, when each model $\vec{F}_m(\vec{s}(k))$ is valid in only one region V_m and not contributing to model outputs in other regions, then, the model (4.1) can be seen as a set of local linear approximations of the system dynamics in each operational region V_m . This simple piecewise modeling strategy is encountered in most of the GSMMS literature and is employed in this work.

In addition, in the context of GSMMS models, sets V_m , $m = 1, 2, \dots, M$ are defined as:

$$V_m = \left\{ \vec{s} : \left\| \vec{s} - \vec{\xi}_m \right\| \leq \left\| \vec{s} - \vec{\xi}_i \right\|, \forall i = 1, \dots, m-1, m+1, \dots, M \right\}, \quad (4.5)$$

where $\left\{ \vec{\xi}_m, m = 1, \dots, M \right\}$ are weight vectors of a growing SOM used to partition the operating space of the model. An illustration of GSMMS with two inputs, one output and five local regression models is given in Figure 4.3, schematically showing how a non-linear dependency is approximated by a set of local linear models.

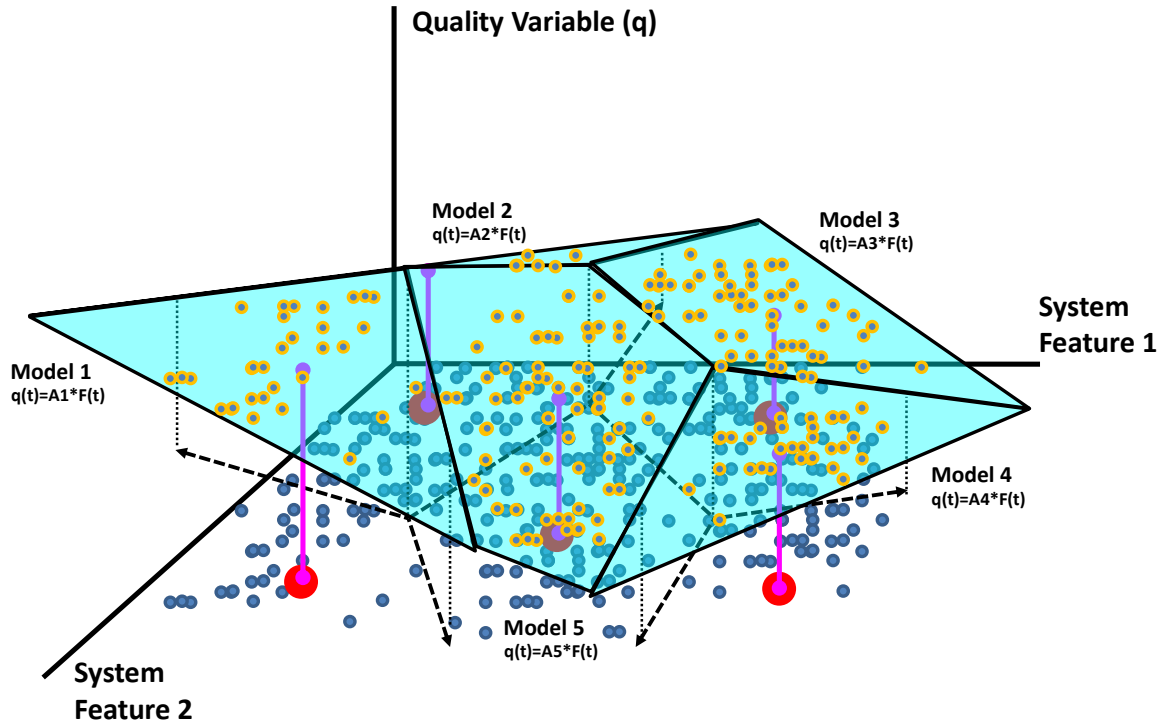


Figure 4.3: Divide and conquer approximation of non-linear behavior using GSMMS with two inputs, one output, and five local regression models.

For the purpose of identification, one has to determine the following parameters using the input-output data:

1. Structural parameters, which include the number of regions M and weight vectors

$$\{\bar{\xi}_m, m = 1, \dots, M\}$$

2. Parameters of local models.

Following [143], in this thesis, the structural parameters of the GSMMS model are found by soft competitive learning with modifications to the learning rate needed for balancing between the local modeling errors and the number of activations each model receives. The learning rates affecting the movement of SOM nodes are made dependent

on the local modeling errors, encouraging movements of the SOM weight vectors towards regions with high local modeling errors. Growth of the underlying SOM (addition of new local models) followed the well-known growing SOM mechanism from Alahakoon et. al. [161], while local model parameters are found by minimizing the weighted squared output errors of the model in its region of validity. For more details on the process of identification of GSMMS models (growth and movement of SOM nodes, as well as identification of local model parameters), please refer to [143] and references therein.

4.2.1 Structural Model Parameters

Given a fixed number of regions M , locations of those regions need to be adjusted in an appropriate way. The standard sequential updating equation for the weight vector $\vec{\xi}_m$ at training step k for a SOM is typically

$$\vec{\xi}_m(k+1) = \vec{\xi}_m(k) + \alpha(k)h(k, dis(m, c))[\vec{s} - \vec{\xi}_m(k)] \quad (4.6)$$

where $\alpha(k)$ is the learning rate, which is a non-increasing function of k , and $h(\cdot, \cdot)$ is the neighborhood function [156]. For soft competitive learning, a common choice for $h(\cdot, \cdot)$ is of the Gaussian-like form:

$$h(k, dis(m, c)) = \exp\left(\frac{-dis(m, c)^2}{2\sigma^2(k)}\right) \quad (4.7)$$

where $c(k) = \arg \min_m \|\vec{s}(k) - \vec{\xi}_m\|$ is the best matching unit (BMU) of the training vector \vec{s} , $\sigma^2(k)$ is a non-increasing function of time that defines the width of the effective range of the neighborhood function, and $dis(m, c)$ denotes the shortest path between node m and

$c(k)$ on the SOM graph, as illustrated in Figure 4.4. The shortest distance between the neighboring nodes and the BMU can be calculated efficiently using the Breadth-first algorithm [156] from the adjacency matrix of the graph that encodes the neighborhood relation of the SOM.

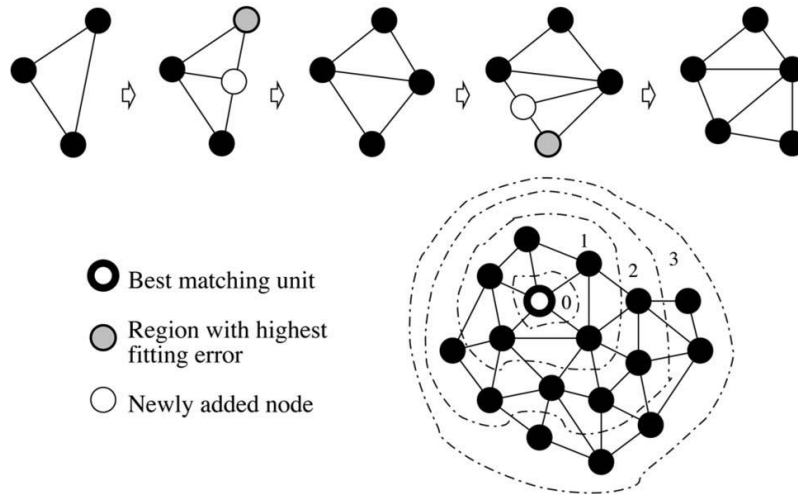


Figure 4.4: Illustration of the training procedure for the GSMMS model and the breadth-first algorithm for calculating the shortest distance between the BMU and its neighboring nodes on the SOM.

Most of the local modeling techniques utilizing SOM in the literature separate the modeling procedure into two independent stages: regionalization and local model fitting. The conventional SOM utilized for unsupervised clustering is normally aimed at minimizing the expected square of the quantization error. However, non-uniformity in the distribution of the input vectors in the training data set may lead to more weight vectors being associated with the regions visited frequently during training rather than where modeling errors are large. This may result in regions, which are highly nonlinear, but not

frequently visited, being poorly approximated by fewer local models. Therefore, it is clear that in order to achieve a better dynamic modeling performance, one needs to balance between the visiting frequencies and modeling errors across different regions. This can be realized by adding a penalty term to the learning rate of the weight vector

$$\vec{\xi}_m(k+1) = \vec{\xi}_m(k) + \tilde{\alpha}(k)\zeta_m(k)h(k, \text{dis}(m, c))[\vec{s} - \vec{\xi}_m(k)] \quad (4.8)$$

where $\zeta_m(k)$ is the penalty term penalizing the modeling error in a given region.

Following [143], the normalized modeling errors will be used in this work to form the penalty term:

$$\zeta_m(k) = \frac{e_m^{EWMA}(k)}{\max_i e_i^{EWMA}(k)} \quad (4.9)$$

$$e_m^{EWMA}(k) = \lambda w_m(\vec{s}(k))e_m^{EWMA}(k-1) + [1 - \lambda w_m(\vec{s}(k))] \cdot \|e_m(k)\| \quad (4.10)$$

where $e_m(k) = \hat{y}_m(k) - y(k)$ is the output error for the m^{th} local model at training step k and $\lambda w_m(\vec{s}(k))$ is the forgetting factor. Such a penalty term will encourage movements of the weight vectors towards regions with high modeling errors. One should note that including the term $w_m(\vec{s}(k))$ into this formulation makes the updating rate for e_m^{EWMA} become less significant when the corresponding node gets farther away from the BMU on the SOM¹².

In the case of batch training, the updating equation at each training epoch k for the weight vector $\vec{\xi}_m$ in the m^{th} region can be simplified as follows:

¹² Superscript ‘‘EWMA’’ means exponentially weighted moving average.

$$\vec{\xi}_m(k+1) = \vec{\xi}_m(k) + \zeta_m(k)h(k, \text{dis}(m, c))[\bar{s}_m - \vec{\xi}_m(k)] \quad (4.11)$$

where \bar{s}_m is the sample mean of the training vectors that fall in the m^{th} region. Compared with the online GSMMS training, no learning rate is involved in the weight vectors $\vec{\xi}_m$. In addition, the normalized modeling error $\zeta_m(k)$ at training epoch k can be calculated as

$$\zeta_m(k) = \frac{\bar{e}_m(k)}{\max_i \bar{e}_i(k)} \quad (4.12)$$

which is similar to the normalized modeling error in the case sequential training with e_m^{EWMA} being replaced by \bar{e}_m , the sample mean of $\|e_m\|$ in the m^{th} region.

4.2.2 Local Model Parameters

A widely accepted method for local model identification is to find the model parameters that minimize the sum of the weighted squared output errors in each operation region.

$$J_m(\vec{\theta}_m) = \frac{1}{k} \sum_{i=1}^k w_m(\vec{s}(i)) \lambda^{k-i} \|y(i) - \hat{y}_m(i)\|^2 \quad (4.13)$$

where $\vec{\theta}_m$ denotes the model parameters that need to be estimated for the m^{th} region, $y(i)$ is the corresponding model output, λ is the forgetting factor that emphasizes the most recently observed signals, and $w_m(\vec{s}(i))$ is the weight describing how the i^{th} observation affects the model parameters in the m^{th} region. Without loss of generality, for notation convenience it is assumed that the dimension of the output is one. It is straight forward to extend the method to the case of multiple outputs.

It should be noted that coefficients $w_m(\vec{s}(i))$ facilitate model building in which a new training pair ($\vec{s}(i) \rightarrow y(i)$) affects not only the local model in the region to which the training sample belongs, but also the local models in the neighborhood regions. A similar procedure has been employed by Martinetz et al. [163] to achieve significantly faster convergence of the model learning process. The coefficients $w_m(\vec{s}(i))$ should emphasize models near the BMU of the training input $\vec{s}(i)$ and de-emphasized models that are far away from in, with notions of “near” and “far” being based on the connectivity of the underlying SOM [143]-[145]. A common selection of coefficients $w_m(\vec{s}(i))$ facilitating such “cooperative learning” is:

$$w_m(\vec{s}(i)) = \exp\left(\frac{-dis(m,c)^2}{2\sigma^2(k)}\right) \quad (4.14)$$

where σ^2 defines the effective range of the weighting function, and $dis(m,c)$ denotes the SOM graph distance between local model m and the BMU c of the input vector $\vec{s}(i)$.

In the case of batch training, where all the training samples are assumed to be available at the beginning of the training process, the local model parameters can be estimated using conventional weighted linear least square method. During each training epoch (passage through the training set), the estimates of the local model parameters $\hat{\theta}_m$ in the m^{th} region can be updated using all the training observations $\vec{s}(i)$, weighted properly by weights $w_m(\vec{s}(i))$ which are determined in the same way as in sequential training.

Besides location of the regions of validity of local models (SOM Voronoi-sets, which also determines their shape), one also has to determine an appropriate number of regions (appropriate number of SOM nodes). Too coarse a partition may result in poor approximations in the region where system dynamics are difficult to model, while too fine a partition may result in partitions that have few or no training samples, which leads to poor local models in those regions. One strategy is to start with a small number of regions and then let the model grow by adding more regions, if necessary. This will be realized in this doctoral research by inheriting the growing mechanism from growing SOMs [143], and exploring necessary modifications.

4.2.3 Updating Methodology for VM Model

In a realistic production environment, it is highly desirable to have VM models that autonomously recognize unusual situations and automatically adapt the relationships between equipment signatures and metrology variables. The most common approach for VM model updating one can encounter in the industry is updating based on periodically scheduled measurements of the quality variables [94], [108]. One disadvantage to using this approach is that most of the time, the underlying manufacturing process is in control and the newly acquired measurements do not offer any new information to the VM model (i.e. those physical measurements are unnecessary). In addition, products whose measurements could potentially yield new information are likely to be missed if they are processed in between the scheduled measurements.

An alternative approach to updating VM models could be to seek and use physical measurements of only those products that could potentially provide new information to

the model. One way to do this is to monitor the sensory features coming from the manufacturing tool (VM inputs), and determine if these inputs are unusually far away from what was seen in the past. In the context of GSMMS based VM, this corresponds to a situation when VM inputs happen to be unusually far away (above some threshold) from the best matching unit (BMU) in the GSMMS model, implying a potential for a poor model output due to the VM inputs being too far from the data on which the VM model is trained.

Consequently, in this work it is proposed that whenever such an input pattern is observed, corresponding physical measurements should be taken and the resulting input-output pair can be added to the VM training set in order to adapt the underlying GSMMS model. The advantage to this updating approach is that only equipment signatures and metrology measurements with potentially new information in the input space are acquired and used for model updating. Furthermore, the GSMMS modeling paradigm naturally supports this process because of its ability to efficiently repartition the input space of the VM model (rebuild the underlying SOM), fit new local linear models corresponding to this new partition and potentially add another local model (SOM node) if sufficiently many novel input-output pairs are presented to it (following the growing SOM method from [161] and implemented within the GSMMS framework in [143],[146]). All these adaptations take place near the newly observed data and taper off with the distance away from them, leading to localized model adaptations that do not disturb previously trained areas of the model. This is in stark contrast compared to the often computationally cumbersome and potentially detrimental global updating involved with the traditional VM methods based on global modeling approaches. Figure 4.5 depicts the localized

updating strategy for GSMMS based on unusual inputs. It illustrates how the SOM grows to incorporate a new operating regime when a set of input patterns (equipment signatures) that are too far outside the original training set appears.

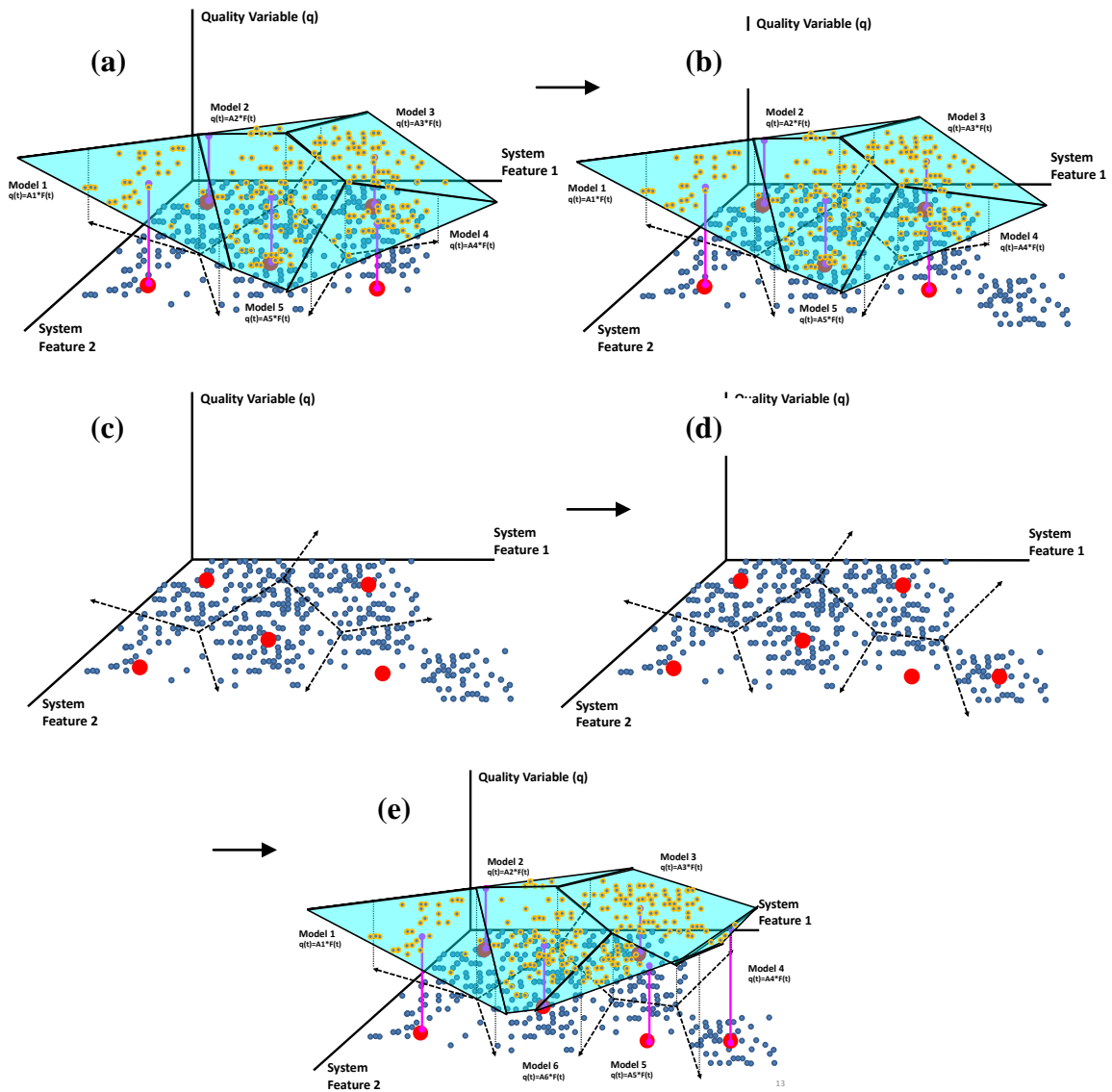


Figure 4.5: Illustration of updating GSMMS based on unusual input patterns. The original GSMMS model in plot (a) estimates the quality variable based on the equipment signatures available at that moment. When a number of equipment signatures appear outside any previously seen local region as shown in plot (b), the original map can grow following the growing method from [159] (plot (c-d)) to incorporate the new operating regime and refit the model for better quality estimation in the new regime (plot (e)).

4.3 Results in VM Quality Estimation using Local Dynamic Model Paradigm

As mentioned in Chapter 3, an extensive dataset was collected from a well-known semiconductor manufacturer, allowing the release of almost a year of PECVD process data from a single tool working in a 300mm wafer fab. It contains equipment signatures corresponding to more than 112,000 processed wafers, which is an order of magnitude larger than any dataset seen in the VM literature thus far¹³. In addition, metrology information (film thickness mean and range) was collected for a significant portion of those wafers. E.g., for the thickest film depositions, 100% metrology information was available (thickness and uniformity were collected after every deposition corresponding to that thickness). Therefore, a deep and comprehensive (perhaps most comprehensive) VM study can be performed with this dataset on one of these complex manufacturing tools. For a more in depth description of the PECVD tool and semiconductor manufacturing process, see [82].

In this study, several months of production was focused on, which yielded data corresponding to about 30,000 wafers undergoing the same TEOS film deposition. In addition, metrology information in the form of mean film thickness was available for each wafer (100% metrology information), allowing a deep and comprehensive VM study¹⁴.

¹³ This can perhaps be blamed on the proprietary nature of semiconductor manufacturing data.

¹⁴ One should remark that this is an order of magnitude larger dataset than any dataset seen in the VM literature thus far, which can perhaps be blamed on the often highly proprietary nature of semiconductor manufacturing data.

From this tool, RF power characteristics (forward, load, and reflected power), voltages of RF matching network capacitors, flow rates in the Liquid Flow Controllers (LFC), top plate temperature, chamber temperature, pedestal temperature, chamber pressure and the pendulum valve angle were concurrently collected at a 10Hz sampling rate. From these sensor readings, 49 features were extracted using expert knowledge and are listed in Table 4.1. Thus, a film deposition onto each wafer could be characterized by 49 sensory features obtained from the mounted sensors on the PECVD tool, along with the post-process metrology measurements of mean film thickness. For more information about the feature extraction see [24], [168].

Table 4.1: Features extracted from the sensor readings to be used as inputs for the VM model.

Sensor	Feature Extracted from Sensor Trace									
Chamber Temp 1	Mean	Range								
Chamber Temp 2	Mean	Range								
Pedestal Temp 1	Mean	Range								
Pedestal Temp 2	Mean	Range								
HF Power	Mean Reflected Dep 1	Mean Reflected Dep 2	Range Reflected Dep 1	Range Reflected Dep 2	Timing Dep 1	Timing Dep 2				
LF Power	Mean Reflected Dep 1	Mean Reflected Dep 2	Range Reflected Dep 1	Range Reflected Dep 2	Timing Dep 1	Timing Dep 2				
Load Capacitor Voltage	Mean Dep 1	Range Dep 1	Max Dep 2	Range Dep 2						
Tune Capacitor Voltage	Mean Dep 1	Range Dep 1	Max Dep 2	Range Dep 2						
LFC Flow Rate	Mean	Range	Overshoot	Rise Time						
Chamber Pressure	Timing Pump Up	Timing Pump Down	Peak Pump Up	Mean Dep 1	Range Dep 1	Timing Dep 2	Mean Dep 2	Range Dep 2	Min	
Pendulum Valve Angle	Timing Pump Up	Max Pump Up	Mean Pump Up	Range Pump Up	Mean Dep 2	Range Dep 2	Max Pump Down	Mean End		

Additionally, we wanted to explore if the current wafer characteristics depend not only on the equipment signatures observed during its production, but also on the equipment signatures observed during the production of several recent wafers. In order to include such dynamic dependencies into the VM modeling, the aforementioned features

obtained from wafers up to 25 cycles before the wafer whose metrology variables are estimated by VM (26 wafers, including the current wafer) were included into the set of potential inputs for the VM model, yielding 1274 possible VM inputs¹⁵.

Due to the large number of inputs, a variable reduction technique was used to reduce the input feature set and thus improve modeling accuracy and speed. In this work, a baseline PLS regression model within a forward selection wrapper in a 10 fold cross validation [33] was used for selecting the most relevant inputs for the model. Akaike Information Criteria (AIC) [33] was utilized to stop the forward selection, yielding a feature subset that achieves a tradeoff between the prediction accuracy and complexity of the resulting VM model. Table 4.2 lists the features yielded by the aforementioned procedure as the VM inputs. One can see that as many as 7 out of these top 10 features come from equipment signatures corresponding to previously manufactured wafers, which shows that there are indeed strong wafer to wafer dependencies contributing to the film thickness quality of a wafer. These are the features used as inputs to the VM models built on the PECVD dataset and analyzed in the next section.

¹⁵ There are 49 features times 26 wafers, yielding the total of 1274 features. This is obviously a large input vector containing current and past equipment signatures

Table 4.2: The features obtained using forward selection applied to the original set of 1274 features and used as inputs for the VM study.

<p><u>Top Features:</u></p> <ol style="list-style-type: none">1. HF Reflected Timing Dep One @ (t-4)2. Tune Capacitor Volt Max Dep Two @ (t)3. Load Capacitor Volt Mean Dep One @ (t-10)4. Tune Capacitor Volt Max Dep Two @ (t-6)5. LF Reflected Timing Dep Two @ (t)6. LFC Flow Mean @ (t-17)7. Tune Capacitor Volt Mean Dep One @ (t)8. Tune Capacitor Volt Mean Dep One @ (t-20)9. PValve Angle Max Pump Up @ (t-1)10. LF Reflected Mean Dep One @ (t-15)
--

4.3.1 Description and Results of VM Comparison Between PLS, T-PLS and GSMMS

Ability of the GSMMS based framework to improve predictions of mean film thickness measurements and to adapt to variable operating conditions was evaluated on the abovementioned PECVD tool dataset. The newly developed GSMMS based VM method was compared to the traditional PLS regression, as well as a recently introduced VM method using PLS regression and process monitoring based on the total projection onto latent structures (T-PLS) method [108],[169]. This VM approach reported in [108], which can be referred to as T-PLS based VM, can be seen as a variation of the well-known PLS regression that uses T-PLS based process monitoring to eliminate outliers from the training dataset, before fitting a PLS model to the remaining data. As reported in [108], the T-PLS based process monitoring can also be used to identify situations when the VM model should be adapted, since its outputs could not be trusted any more, though

details as to how that can be done were not discussed in [108]. The comparison between these three models was made on the basis of mean squared error (MSE) of predicted mean wafer thickness, number of physical measurements necessary for obtaining a particular MSE, and percentage of wafers for which the VM predictions had relative errors¹⁶ below 3%, and below 1%.

The first set of analysis results was obtained using unusual inputs as the only way of calling for physical wafer measurements and triggering model updates. Specifically, the original models were built using the first 100 wafers and the models were simulated such that only wafers associated with abnormal equipment signatures (VM inputs) are physically measured, added to the training set and used for updating. The PLS based VM model updating is triggered when an unusual input is detected based on the traditional T-squared statistic [87]. T-PLS used the same criterion to identify unusual VM inputs, except that T-PLS process monitoring method was used for rejection of outliers in the training dataset before an updated model was fit to the data [108]. Figure 4.6 shows the results of comparison of the GSMS, PLS and T-PLS based VM, using the above-described updating strategy. The plots detail the actual mean wafer thickness data along with the model predictions for all 30,000 wafers. The MSE is scaled to the worst case model. One can see that GSMMS gives around 5% lower MSE than PLS, while T-PLS-based VM yielded MSE that was even lower than that of the GSMMS based VM (by about 3%). The reason for this slight increase can be seen in the fact that T-PLS process monitoring rejects outliers from the model-training dataset, thus enabling improved modeling of the in-control data, which naturally dominate the dataset and this pushes the

¹⁶ Absolute value of the difference between VM prediction and actual metrology measurement, divided by the actual metrology measurement.

MSE further down. However, the number of physical measurements required to achieve these MSE levels went from 576 for PLS-based VM, to 133 for T-PLS based VM, to just 24 for GSMMS. Another benefit that can be noticed for the GSMMS based VM is that physical measurements are called for less and less frequently as more wafers pass by¹⁷, with no wafers being called for measurement in the last 1/3 of the data, while PLS based VM schemes continue to call for measurements throughout the dataset.

¹⁷ Because more regions are built into the GSMMS model as it learns new operating regimes for the VM model.

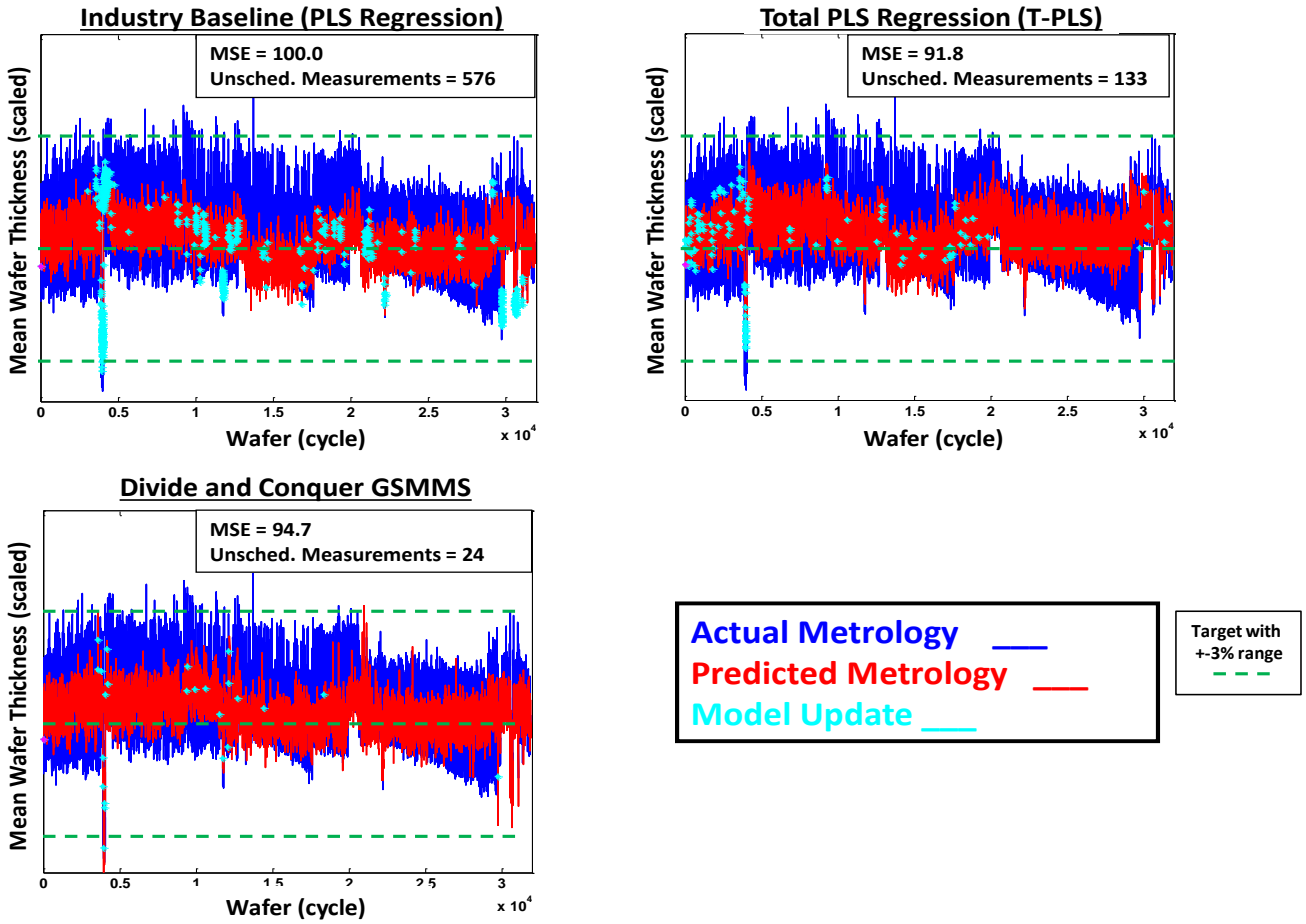


Figure 4.6: VM results of comparing PLS, T-PLS and GSMMS for prediction of mean film thickness using VM updating based on unusual inputs¹⁸.

Next, the performance of these three VM models was compared for the situation when physical measurements of film thicknesses were conducted at a fixed schedule of 1 every 25 wafers. Once again, original models were built using the first 100 wafers and the models were updated if the relative VM prediction error on a scheduled measurement was above 1%. Figure 4.7 shows the mean wafer thickness prediction results of comparing the PLS, T-PLS and GSMMS based VM using this updating strategy. One can

¹⁸ Unscheduled measurements shown in the figure refer to times when unusual inputs are detected, triggering a wafer measurement and a model update consecutively.

see that due to many more physical wafer measurements, the MSE is about 15% lower than what was obtained using the unusual inputs based updating alone and the three models have comparable accuracies in terms of MSE. One disadvantage of this updating method is that only around one sixth of the physical measurements were actually used for model updating for each of the modeling schemes considered, since many times the measurements were taken, the VM model did well predicting them (relative prediction error was less than 1%). This weakness is common to all three models and must be attributed to the updating scheme, which was based on a fixed schedule of measurements, rather than potential novelty a physical measurement could bring to the VM model.

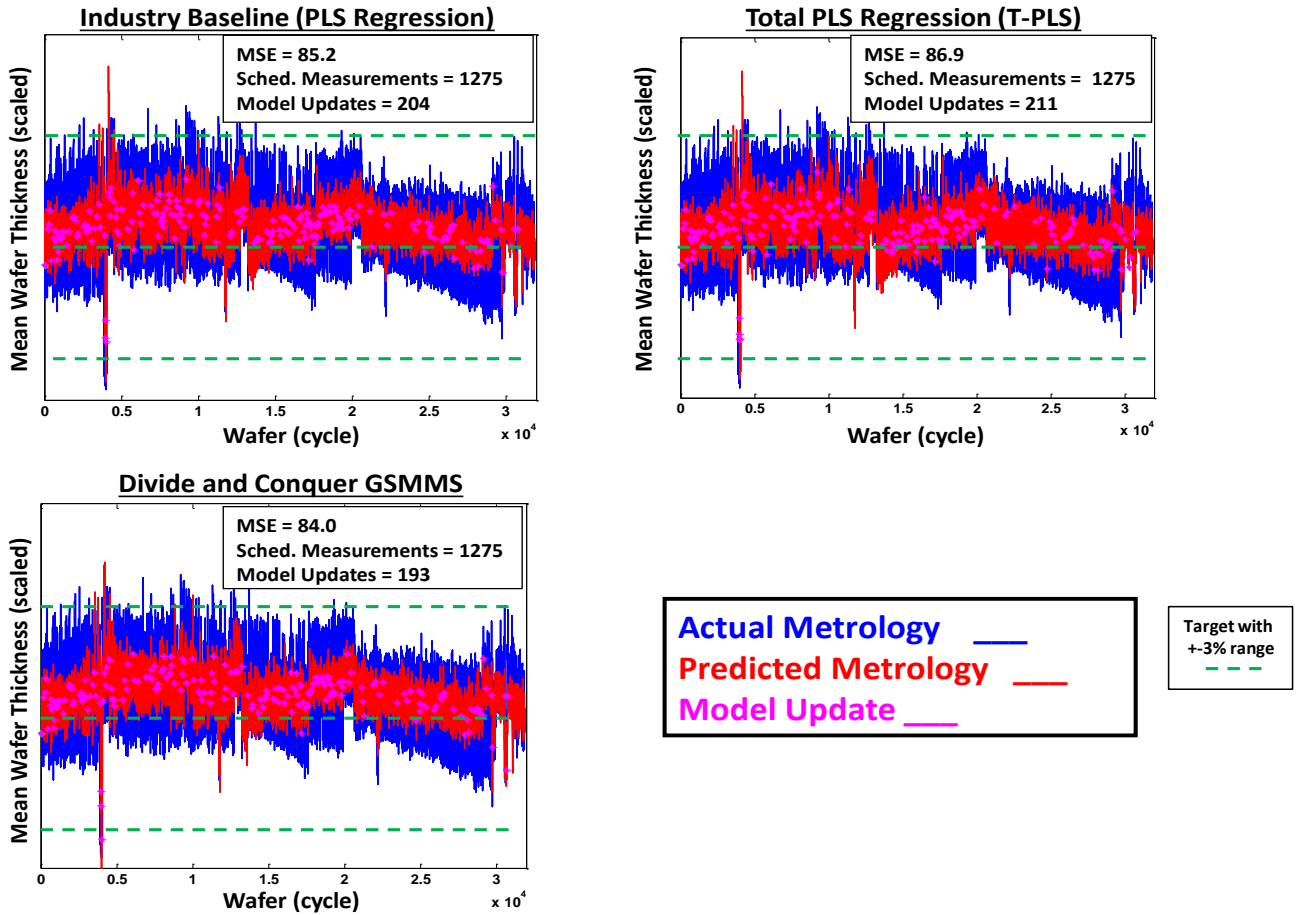


Figure 4.7: VM results of comparing PLS, T-PLS and GSMMS for prediction of mean film thickness using VM updating based on fixed schedule measurements. Please note that “Model Updates” counts only the updates from the scheduled measurements (triggered from the 1% prediction error threshold).

In order to combine the benefits of the two aforementioned updating schemes for VM models, performance of the two VM methods was evaluated using an updating scheme that combines them both. As before, the original models were built using the first 100 wafers, while the physical measurement were assumed to be taken when unusual inputs are observed, as well as at a fixed schedule of 1 out of every 25 wafers being physically measured. The VM models were updated when an unusual input was flagged,

as well as when the relative VM prediction error of a scheduled measurement was above 1%. Figure 4.8 shows VM predictions of the mean wafer thickness using PLS, T-PLS and GSMMS based VM, with the combined updating strategy.

One can see that in terms of MSE, the GSMMS based VM is about 7% more accurate than PLS-based VM, and 4% more accurate than T-PLS based VM. What is particularly interesting is that when unusual inputs are included as a criterion for VM model updates, the MSE of the PLS model increased by 4%, the MSE of T-PLS based VM remained virtually unchanged, while the MSE of the GSMMS-based VM decreased about 2%.

This can be explained by the way the 3 models handle unusual inputs. When the inputs that are far outside the training regime are included to update the PLS and T-PLS models, the entire model has to be refit and these unusual VM inputs may spoil the way the model fits the rest of the data, giving higher errors in those regions. By contrast, whenever unusual inputs are detected, the GSMMS model retrains the underlying SOM using the expanded training set and refits the local linear models, potentially adding another model, following the growth mechanism from [159]. These adaptations are the most intense close to the newly added input-output pairs and taper off further away from them, thus preserving to a large degree the previously learnt model regimes. This localized adaptation mechanism accommodates the new information in the least intrusive way, thus keeping the MSE lower than PLS or T-PLS based method. Localized adaptations enabled the GSMMS-based VM to call for only 18 physical measurements beyond the initial 100 wafers. In the same time, the PLS called for 598 and T-PLS called

for 168 additional measurements, and yet the PLS based VM schemes showed noticeably higher MSE.

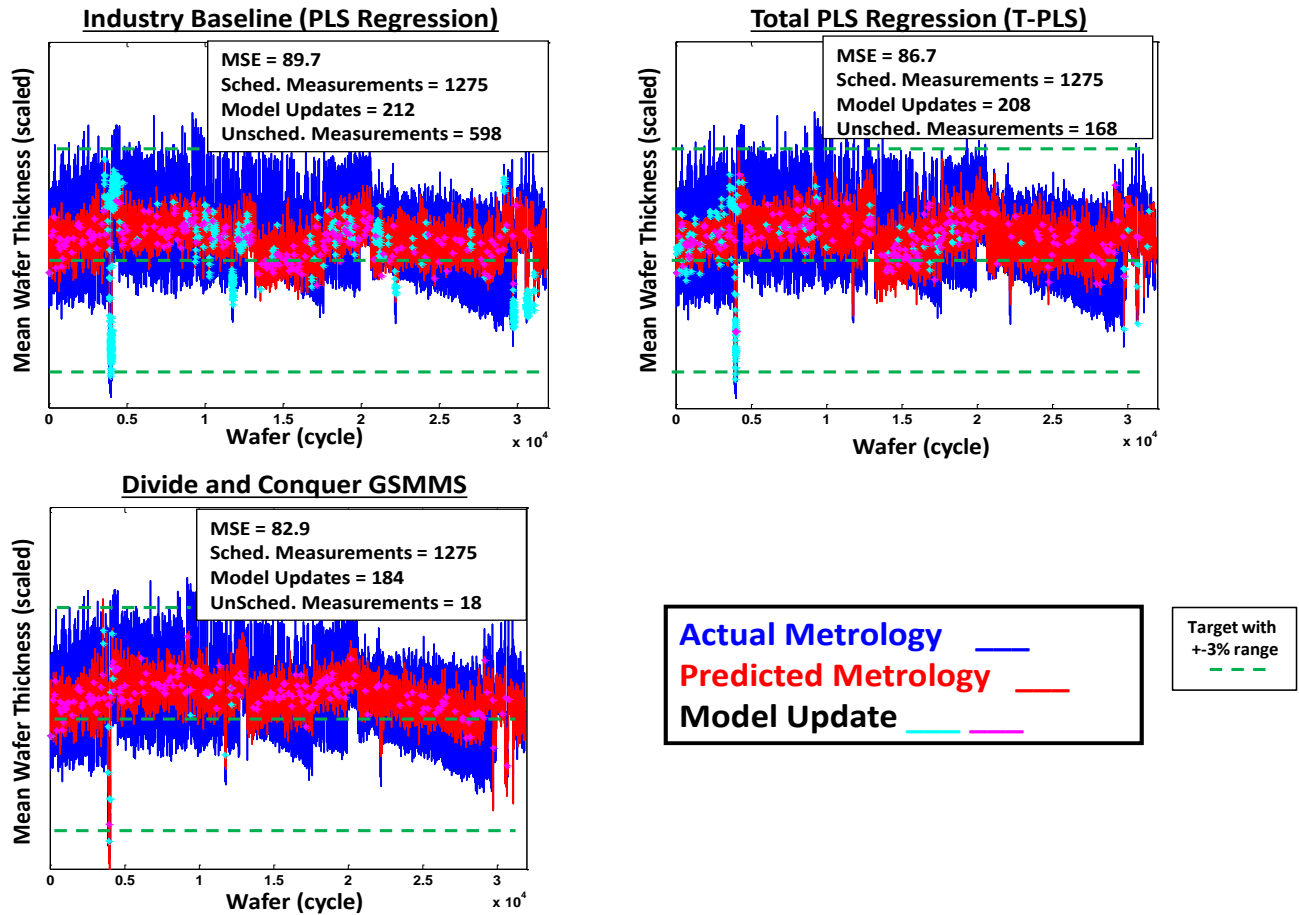


Figure 4.8: VM results of comparing PLS, T-PLS and GSMMS based VM for prediction of mean film thickness, using concurrent model updating based on fixed schedule measurements and based on unusual inputs. Please note that “Model Updates” counts only the updates from the scheduled measurements (triggered from the 1% prediction error threshold). Also, the unscheduled measurements bring their own model updates which should be added to this number to find the total model updates.

The results shown in the previous three pictures just show a snapshot of different updating policies between the PLS, T-PLS and GSMMS based. A full comparison of these methods was conducted to see how the accuracy of each model changes as the number of scheduled physical wafer measurements is decreased from 1 out of every 8 wafers, to 1 out of every 200 wafers. All models were initially built using the first 100 wafers, while updating was done using only scheduled measurements, as well as using the combination of measurement acquired at a fixed, as well as whenever unusual VM inputs (equipment signatures) were observed. Figure 4.9 shows the MSE of PLS, T-PLS, and GSMMS as a function of the number of wafers between the scheduled physical measurements. The red solid line with star markers shows the MSE of the PLS based VM updated based on scheduled physical measurements only, while the dashed red line shows MSE of the same VM method, but with updating based on a combination of scheduled measurements and triggering off unusual VM inputs. In a similar way, the green solid line with triangle markers shows the MSE of the T-PLS based VM updated based on the scheduled physical measurements only, while the dashed green line shows MSE of the T-PLS based VM method, with combined updating using fixed scheduled measurements and unusual VM inputs. Finally, the blue solid line with star markers shows the MSE of the GSMMS based VM with updating based on the scheduled physical measurements only, while the dashed green line shows MSE of the same VM method, but with updating based on a combination of scheduled measurements and triggering off unusual VM inputs. For more precise numerical analysis, one can refer to Table A1 in the Appendix, which tabulates the results shown in Figure 4.9.

One can see that in all but 3 situations, GSMMS based VM gave better accuracy than PLS and T-PLS, regardless of the updating scheme. Furthermore, one can see that the GSMMS based VM with the combined updating strategy consistently outperforms all other VM models and updating strategies. As a comparison to having a zero order model (i.e. predicting every wafer to be at the mean value) the MSE would be at 122, meaning these models are making a difference in prediction accuracy. Effectively, results from Figure 4.9 (Table A1) illustrate that GSMMS based VM requires often significantly less measurements than PLS type models in order to achieve the same levels of accuracy.

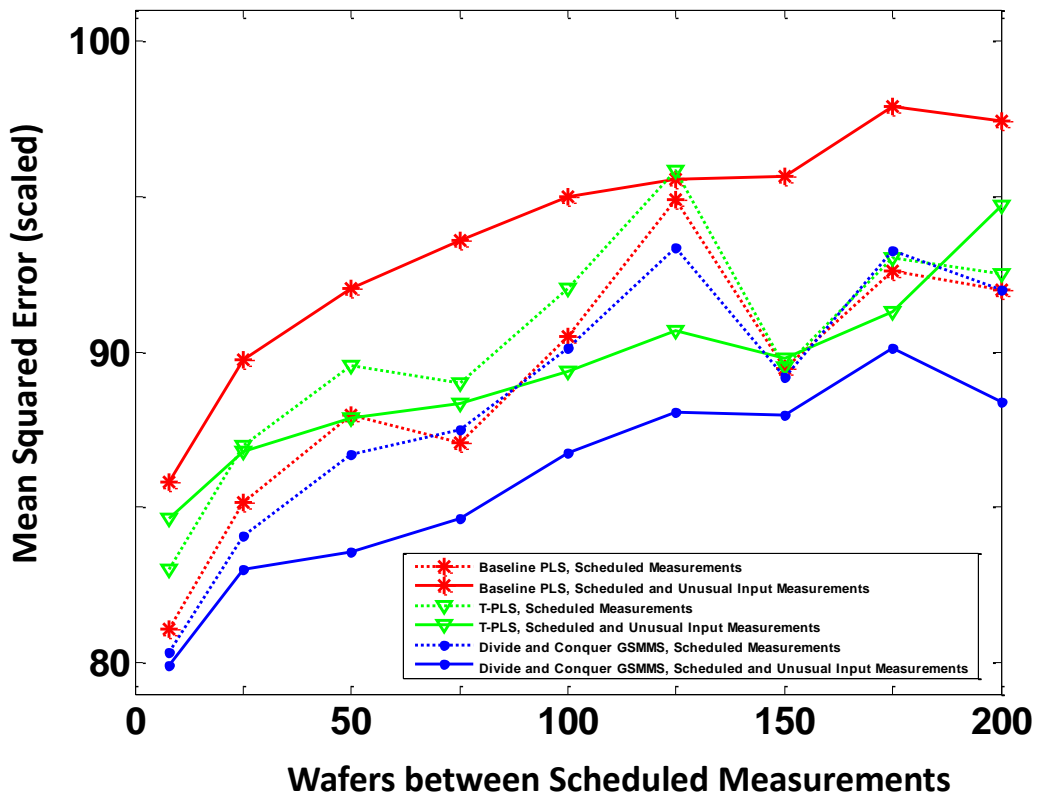


Figure 4.9: Mean squared error for predicting mean wafer thickness of PLS, T-PLS and GSMMS models as the time between scheduled physical wafer measurements increases from 1/8 wafers to 1/200 wafers. GSMMS using both types of updating outperforms the other methods, providing equal accuracy with significantly fewer measurements.

As mentioned in Section 4.1.1, moving windows have been recently used in VM to obtain better accuracy when conditions suddenly change in the system [94], [108]. In our work, the previous analysis was also performed with moving windows of various sizes ranging from 25 to 300 wafers with the PLS and T-PLS models in order to investigate the benefits of using simple moving windows with this dataset. A 100 wafer window size was found to give the best results for both PLS and T-PLS based VM and Figure 4.10 shows MSE results for the windowed PLS and T-PLS based VM methods, along with the MSE results of the GSMMS based VM approach. As visible in Figure 4.9, GSMMS based VM with the combined updating strategy still outperforms the windowed PLS-based VM methods, regardless of how measurements were acquired for model updating. Table A3 in the appendix lists the data for this analysis.

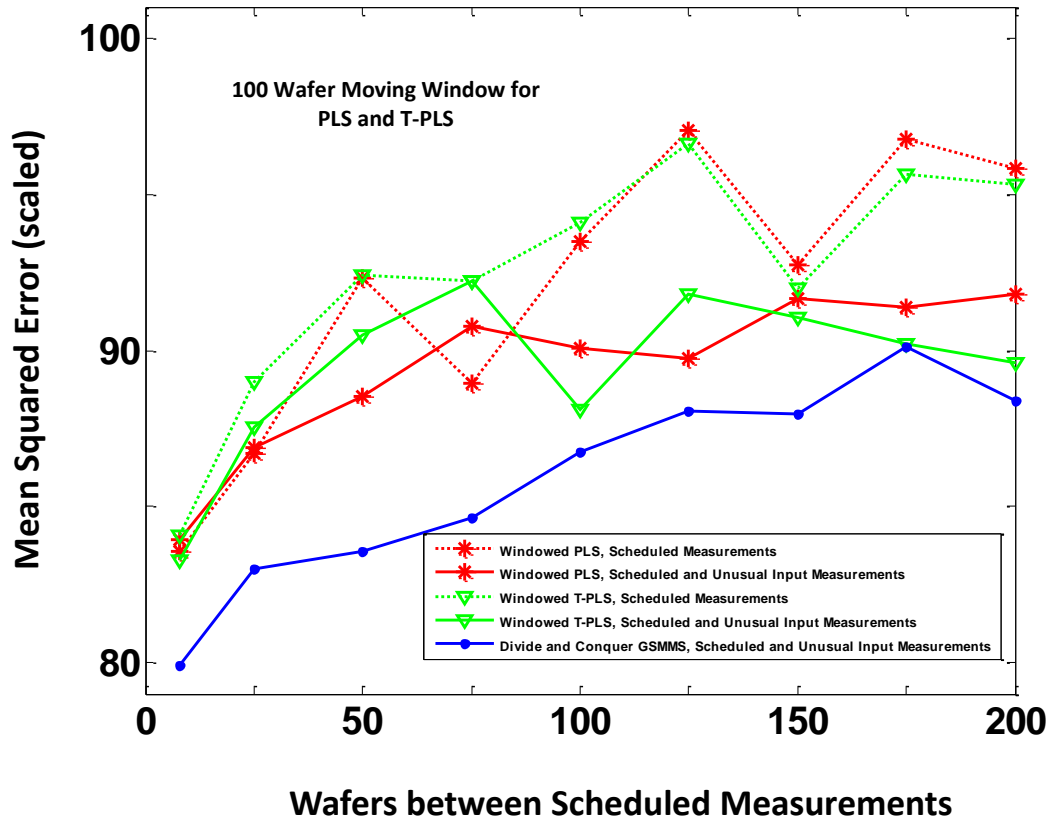
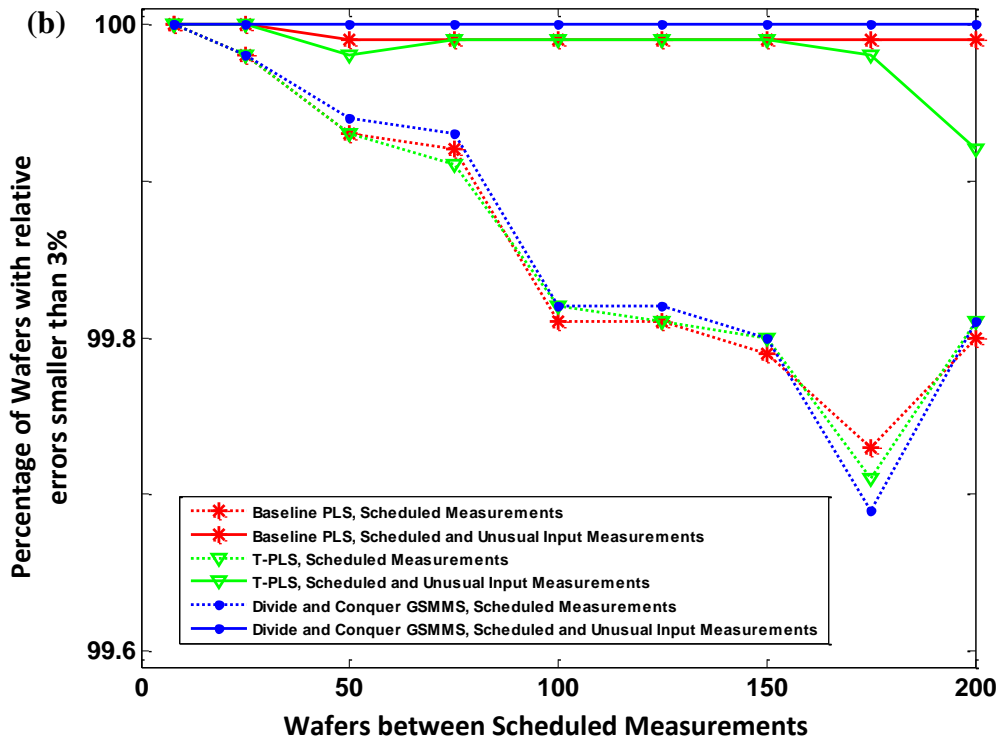
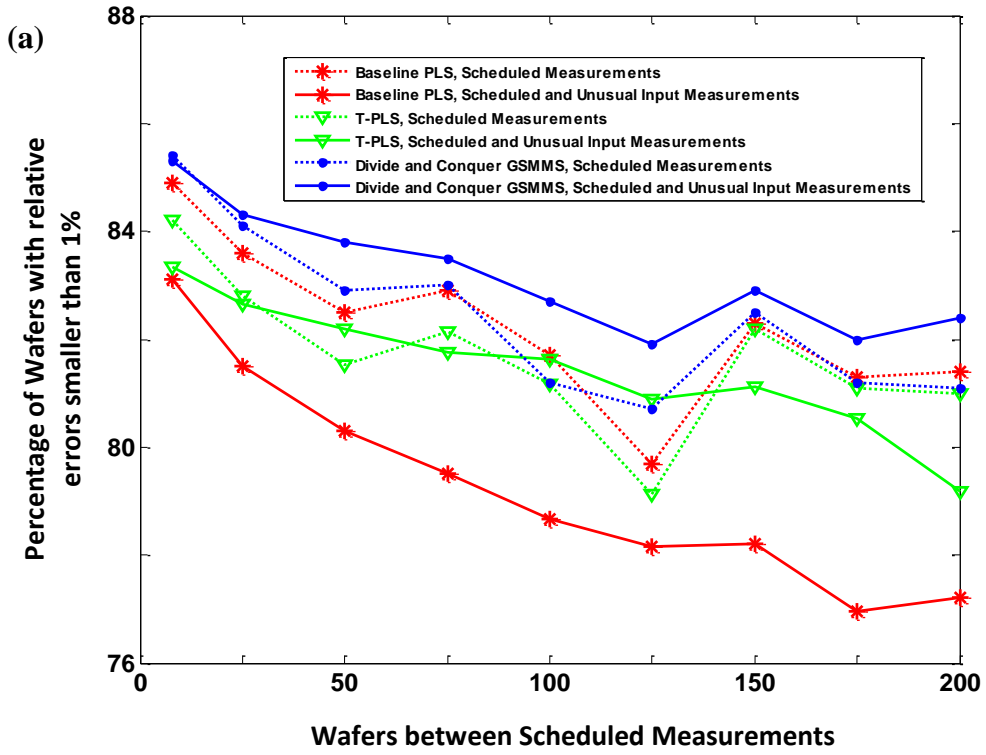


Figure 4.10: Mean squared error results similar to the previous figure using a 100 wafer moving window for updating the PLS and T-PLS VM models. MSE curve for the GSMMS based VM method with the combined measurement acquisition strategy is superimposed to illustrate its superiority over the PLS based methods.

Another standard and convenient way to evaluate a VM model is to tabulate percentages of products for which relative VM errors fall below certain levels. For both updating schemes considered in Figures 4.9 and 4.10, Figures 4.11 and 4.12 show percentages of wafers for which PLS, T-PLS, and GSMMS based VM had relative errors below 1% (plot a) and below 3% (plot b), while plot c of the same figure shows maximal relative errors. One can see that GSMMS based VM with concurrent updating based on fixed schedule measurements and unusual equipment signatures consistently outperforms

all other VM methods, regardless of the updating scheme. Furthermore, unlike the PLS based VM methods, it consistently benefited from added consideration of unusual equipment signatures (VM inputs) as a criterion for model updating¹⁹. This is yet another proof that it effectively uses the divide and conquer paradigm to enable localized model adaptations to newly observed VM inputs, while similar adaptations in the global-style PLS models sometimes deteriorated the resulting VM performance. As a comparison to having a zero order model (i.e. predicting every wafer to be at the mean value) the percentage of wafers with relative errors smaller than 1% and 3% are 63.1% and 99.8% respectively, which is much smaller on the 1% accuracy metric meaning these models are making a difference in how many wafers are predicted within specifications. A maximum error of 3.57% was found for a zero order model, which is higher than using the dual updating strategy. Table A2 and A4 in the Appendix tabulates the results shown in Figure 4.11 and 4.12 respectively.

¹⁹ In plots a and b of Figure 8, this is evident in blue solid curves being always above the dashed blue curves, while in plot c, this is evident in the dashed blue line being above the solid blue line. Such pattern could not be observed with curves corresponding to the PLS based VM methods.



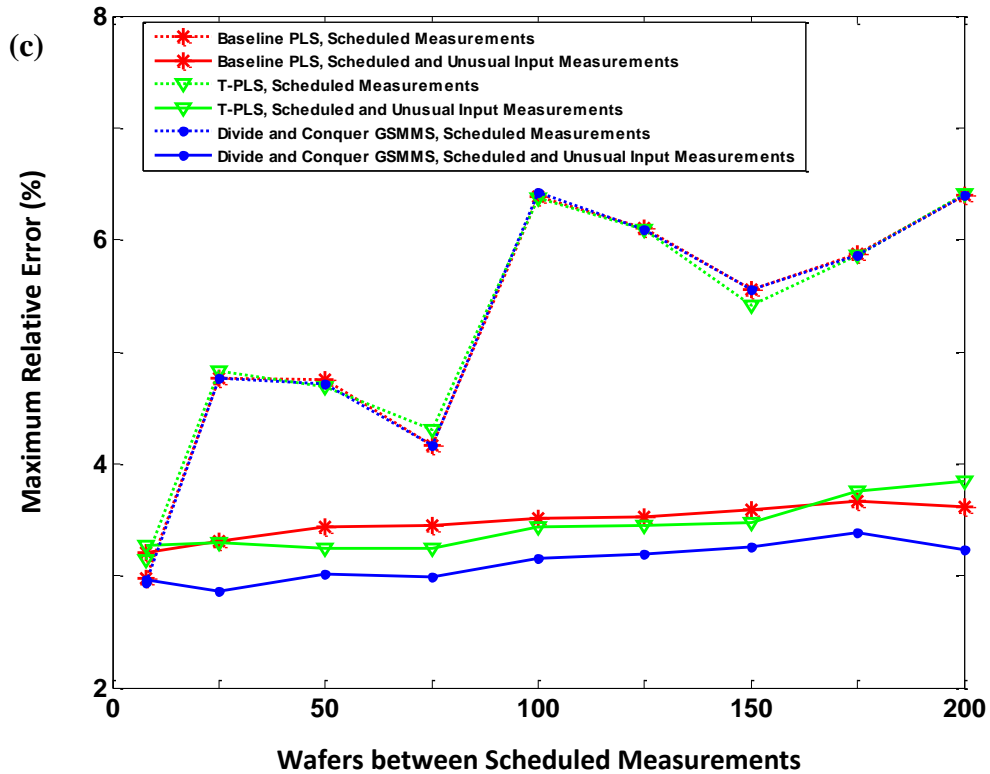
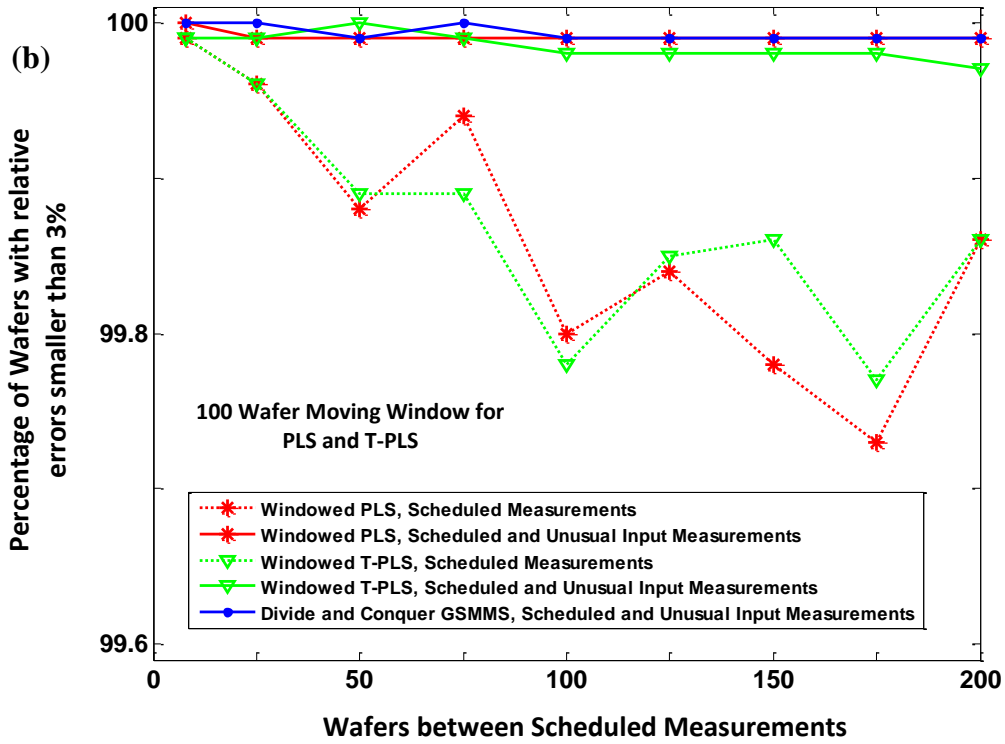
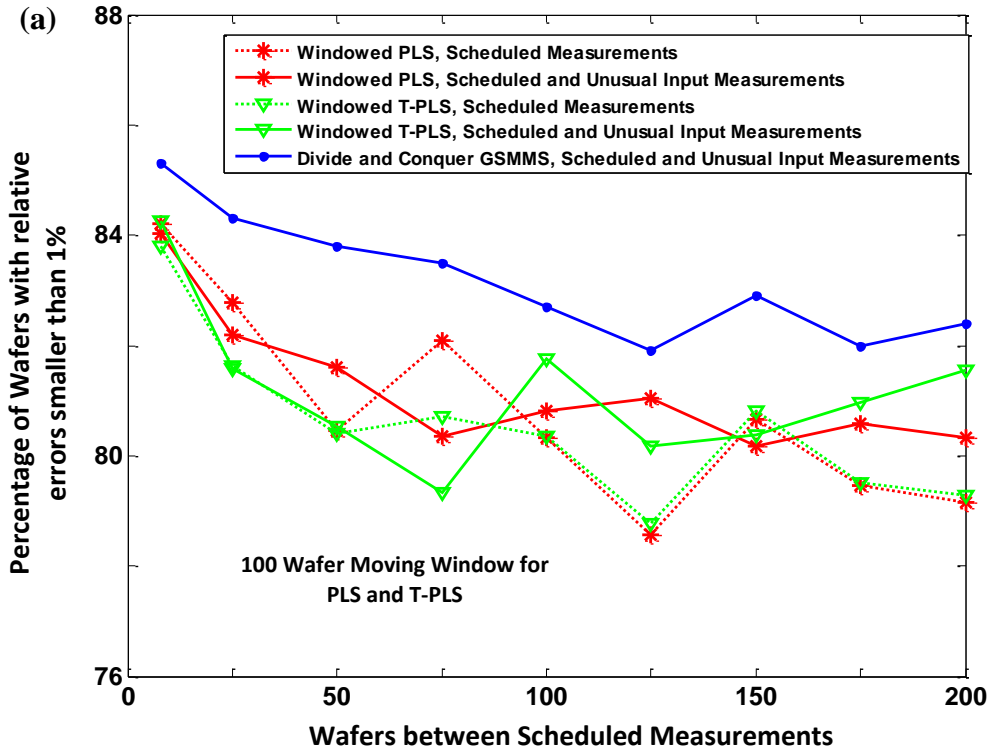


Figure 4.11: Plot (a) shows percentages of wafers for which PLS, T-PLS, and GSMMS based VM had relative errors below 1%, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this paper. Plot (b) gives analogous results to those from plot (a), except that we see percentages of wafers with relative errors of VM below 3%. Plot (c) shows maximal relative errors corresponding to the PLS, T-PLS, and GSMMS based VM methods, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this paper. Please note that a 1% difference in the number of wafers in this dataset is around 300 wafers.



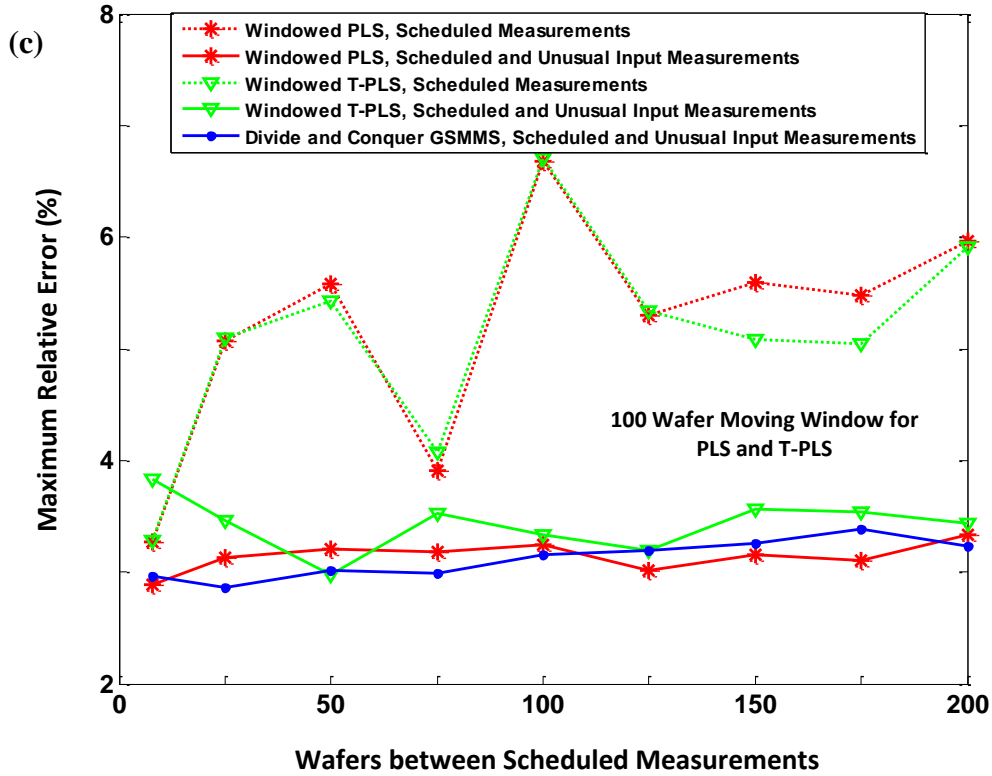


Figure 4.12: Plot (a) shows percentages of wafers for which windowed PLS, windowed T-PLS, and GSMMS based VM had relative errors below 1%, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this paper. Plot (b) gives analogous results to those from plot (a), except that we see percentages of wafers with relative errors of VM below 3%. Plot (c) shows maximal relative errors corresponding to the windowed PLS, windowed T-PLS, and GSMMS based VM methods, for various frequencies of fixed schedule physical measurements and for both VM updating strategies considered in this paper. Please note that a 1% difference in the number of wafers in this dataset is around 300 wafers.

For a final test on the adaptability of VM models, using the fully trained models of PLS, T-PLS and GSMMS after applying the dual updating scheme throughout the entire 30,000 wafer dataset were simulated through the dataset for a second time. Figure 4.13

shows the results of applying the dual updating scheme (using a 1 out of 25 wafer updating schedule and unusual input updating) to each of the three models on the second round of data beginning with the fully trained models at the end of Figure 4.8. One can see that the GSMMS updates much less on the second run through the dataset and only one time due to unusual inputs. As for PLS and T-PLS, the models both update in a similar fashion as the first time going through the data because of their inability to grow and adapt without compromising predictions in other operating regimes. This demonstrates that GSMMS learned the behavior of the PECVD tool and was able to take this learned model into predicting similar operation, which was not shown to occur in the global PLS and T-PLS style models.

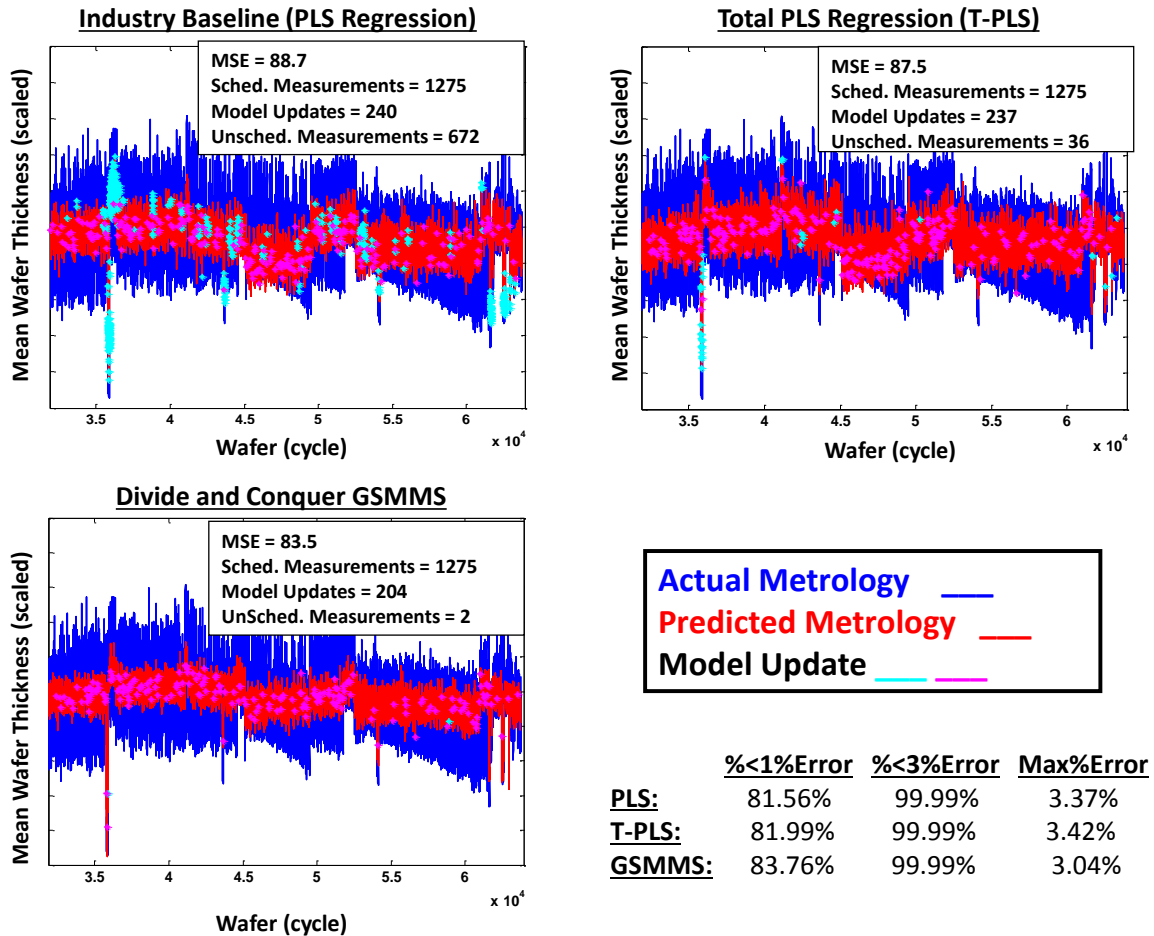


Figure 4.13: VM results of comparing PLS, T-PLS and GSMMS based VM for prediction of mean film thickness, by demonstrating how much updating must be needed when simulating the entire dataset for a second time around. Updating here uses concurrent model updating based on fixed schedule measurements and based on unusual inputs exactly as Figure 4.8.

4.3.2 Conclusions for Comparison of PLS, T-PLS and GSMMS VM Models

This Chapter introduced the concept of growing, locally linear dynamic models for VM applications. The newly developed VM method was evaluated on a uniquely large dataset with equipment signatures and mean film thickness measurements obtained from over 30,000 wafers processed on a single PECVD tool in a major semiconductor

manufacturing fab. This dataset was used to build and evaluate the GSMMS based VM methodology, as well as to compare it against the more traditional PLS-based approaches. The results show that in terms of modeling errors and amount of necessary physical wafer measurements, the newly introduced GSMMS-based VM models outperform PLS and T-PLS regression based VM. Furthermore, when GSMMS based VM model was updated using scheduled measurements and whenever unusual VM inputs appeared, it consistently outperformed PLS-based VM methods, regardless of the metric used to describe model accuracy and regardless of the updating scheme of the VM model.

The main reason why GSMMS outperformed the PLS based VM approaches and is likely to outperform all VM approaches that use global modeling paradigm is in the specific way its adaptations occur. When inputs that are far outside the training set and the corresponding metrology measurements are used to update a model, global models such as PLS and T-PLS, have to refit the entire model, often disturbing model fits corresponding to the rest of the data and thus leading to higher prediction errors. GSMMS, by contrast, adapts only local model parameters near the input-output pairs newly added to the training set (i.e. input-output pairs that triggered model adaptations), with adaptations tapering away with the distance from those input-output pairs. In effect, new information is accommodated via localized model adaptations and growth of new modeling regimes, without disturbing the previously learnt model structures, which keeps the modeling errors and number of necessary physical measurements lower than what is seen with global modeling strategies.

Chapter 5

Integration of Performance Prediction and Quality Estimation of Monitored Systems using Uncertainty Propagation Techniques

5.1 Uncertainty Propagation through Virtual Metrology Model Methodology

When it comes to uncertainty propagation for the purpose of predictive VM pursued in this doctoral research, the underlying manufacturing systems are inherently non-linear with non-stationary and non-Gaussian noise, which means that analytical uncertainty propagation techniques with their strong mathematical assumptions necessary for their implementation cannot be met in reality. In addition, the VM model pursued in this dissertation, and used as the foundation of the predictive VM sought, just like most VM models encountered in the literature, is used to predict a single output, meaning that the problems associated with multiple outputs of the dynamic model, which plague MC simulation based methods, do not exist. Therefore, the MC simulation based method will be employed to obtain the estimates of the pdf-s of future metrology variables using predicted pdf-s of the equipment signatures, as obtained using methods from Chapter 3 and feeding those pdf-s into the VM model developed in Chapter 4. The remainder of this Chapter will present the methodology behind this approach and results applied to a semiconductor manufacturing process.

In prior work by Bleakie et. al. [24], we derived a novel similarity-based time-series prediction method in order to predict the performance of features extracted from sensor readings obtained from the monitored machine. This method was demonstrated to be effective for modeling multivariate time-series with non-linear dynamics, and non-stationary, non-Gaussian noise characteristics and enabled highly accurate long term time-series prediction with an order of magnitude faster computation time than the other comparable models. Therefore, a similarity-based methodology will be pursued in this work in order to achieve the goal of accurate long-term prediction of performance signatures of degradation processes as the first step in achieving a predictive VM framework.

As for the VM modeling framework to be pursued in this Chapter, we will focus on the divide and conquer modeling strategy introduced in the previous Chapter and in Bleakie et. al. [25]. In this work, a growing self-organizing map (SOM) is utilized for partitioning of the VM input space, while inside each partition a local linear dynamic model is used for predicting the metrology outputs. This model referred to as the Growing Structure Multiple Model System (GSMMS) allows the underlying SOM to grow and thus to adapt to new operating regimes that the system under study may experience over time. Our previous work demonstrated that these favorable properties of the GSMMS based VM lead to better VM accuracy with fewer physical metrology measurements compared to the traditional VM global models.

As mentioned in the introduction, the fully integrated VM quality estimation scheme will involve accurate prediction of sensor readings being emitted from the manufacturing systems, while also feeding these sensor predictions through an accurate,

adaptable VM model in order to obtain long term predictions of the hard to measure metrology characteristics of the system. Figure 5.1 schematically illustrates the predictive VM scheme pursued in this Chapter.

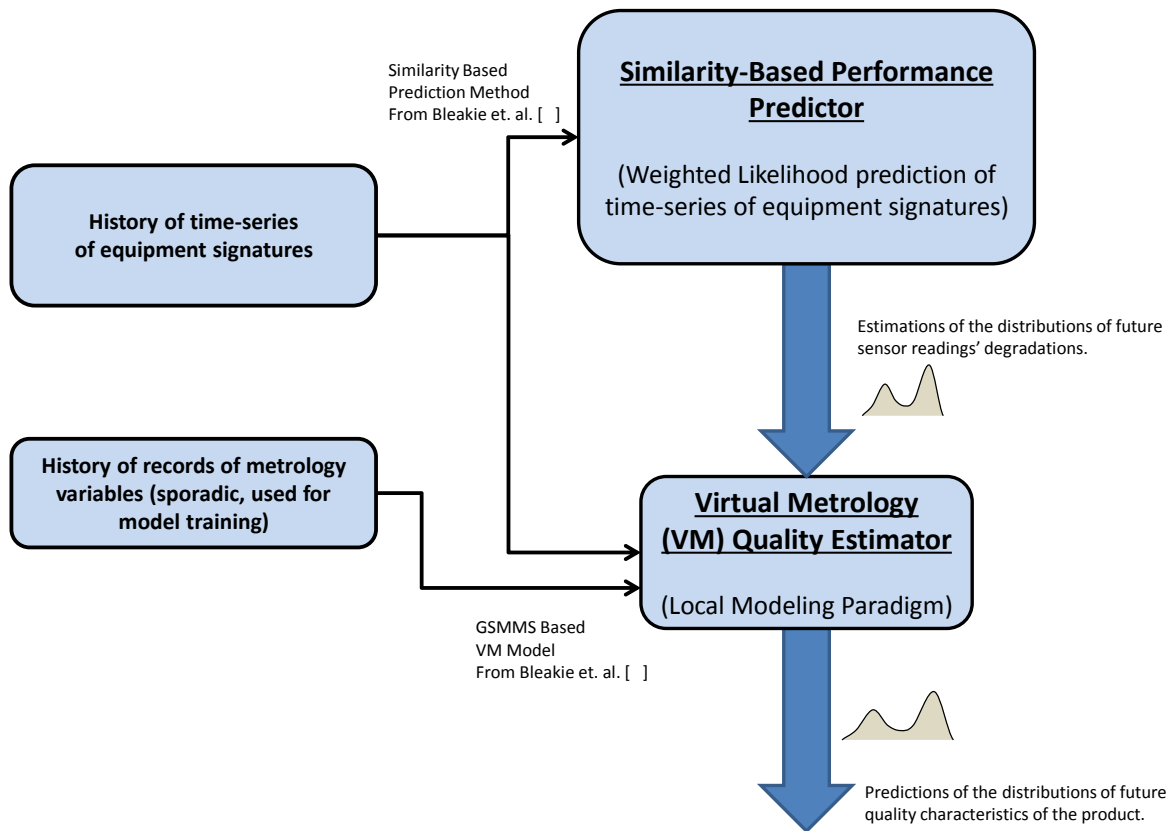


Figure 5.1: Demonstration of the predictive VM framework developed in this Chapter.

The Similarity Based Performance Predictor uses past historical performances in order to predict the remaining performance of the current operations based on the similarity of the currently forming operation to the past performances. Previously defined in Chapter 3, the following terms must be known in order to place this predictive VM

analysis into context. The term *feature vector* is used to indicate the signatures extracted from the raw sensors mounted on the monitored system that are known to characterize the condition of that system. Evolution of these signatures is then indicative of system degradation and their behavior needs to be predicted. The term *cycle* is used to indicate a single operation by the system, emitting a single feature vector. This can be any manufacturing operation or single use of a product. Repeated cycles degrade the system, causing the signatures to evolve and ultimately lead to maintenance events. The term *run* is used to indicate the time interval between two consecutive maintenance events. These maintenance events can be component replacements, repairs, cleaning, etc. Thus, a time-series of feature vectors in a past run represents a particular degradation trajectory known from historical data.

Condensing the methodology of Chapter 3, the Similarity Based Performance Predictor uses probability density functions (pdf-s) representing feature vectors from the previous runs as the models for evaluating similarities between the newly observed trajectory of signatures and those observed in the past. The pdf-s are approximated using Gaussian mixture models (GMM) due to their ability to model any distribution within a desired accuracy, given enough Gaussian components [37]. Figure 3.2 previously described the newly proposed time-series prediction algorithm from [24]. Right after a maintenance operation (i.e. just before a new run of operations begins), the only information known about the run that is about to start are feature vector realizations observed during the previous runs of the monitored system. At each cycle, GMMs of feature vectors corresponding to that cycle in the previous runs can be formulated. As the current run progresses, feature vectors from more and more cycles are observed and

similarity measures between those feature vectors and all feature vectors corresponding to that cycle in past runs can be evaluated. Consequently, at cycle number i , we can observe the set of vectors \bar{s}_c^* ; $c \in \{0,1,2,\dots,i\}$, composed of similarities between feature vectors observed at cycle number c , with feature vectors observed at cycle c in all previous runs. These similarity measures can then be used to skew the GMMs of feature vectors corresponding to future cycles of the current run (cycles $i+1$, $i+2$...) towards feature vectors from previous runs that in the past cycles showed more similarity with the current run. Thus, as time progresses and more and more signatures are collected during the current run, the feature models shift toward the most similar runs observed in the past. When the current run is completed, it can be incorporated into the library of previous runs, thus enabling continuous learning as the system progresses through its lifetime. For more details of this method, refer to [24].

GSMMS based estimates of quality variables introduced in the previous Chapter is pursued in this work due to its ability to adapt and grow in the data, while modeling non-linear dynamic systems with high levels of accuracy and significant levels of local model tractability. GSMMS uses a growing Self-Organizing Map (SOM) [145] to adaptively partition the operating space of the system into regions of behavior that can be locally described using analytically tractable, linear dynamic models. Essentially, it “tiles” a nonlinear surface with flat (linear) tiles, whose number, size, position and shape are determined through the growth and adaptation of the SOM. The evolution mechanism of the growing SOM introduced in [144] ensures that highly curved surface areas receive more tiles of smaller size, while linear areas are approximated by fewer tiles of larger size. In the context of VM, the signatures extracted from the sensors mounted on the

manufacturing equipment whose VM model is being built are used as VM model inputs, based on which the SOM partitions the operating space of the model into regions within which local linear models are built to relate those equipment signatures (VM inputs) with the metrology variables (VM outputs). Such piecewise modeling approach enables accurate modeling of non-linear dynamic dependencies, noise characteristics being possibly different in different operating regions of the model (i.e. with noise being non-stationary) and understanding of local model confidence due to local model tractability. That in turn enables one to recognize situations when the GSMMS based VM model cannot be trusted anymore because one operates in a regime where a reliable local model does not exist (when equipment signatures inputs lead us to a region where the local model is not identified with high enough confidence). Figure 4.3 schematically illustrates a GSMMS model with 2 inputs, one output and 5 local linear regression models approximating a nonlinear dependency, while Figure 4.5 illustrates the growth of this model when novel VM input signatures are observed. More details about the GSMMS based VM can be found in Chapter 4 of this dissertation, as well as in [25].

In this Chapter, the novel time-series prediction method and the newly introduced GSMMS based VM approach are combined into an integrated, predictive VM framework illustrated in Figure 5.2. The predicted distributions of equipment signatures are propagated through the GSMMS VM model to obtain the predicted quality distributions, which enables one to continuously understand and track the behavior of quality variables. Such capability could in term lead to greatly improved run-to-run control of the underlying process, though such work is outside the scope of this doctoral research.

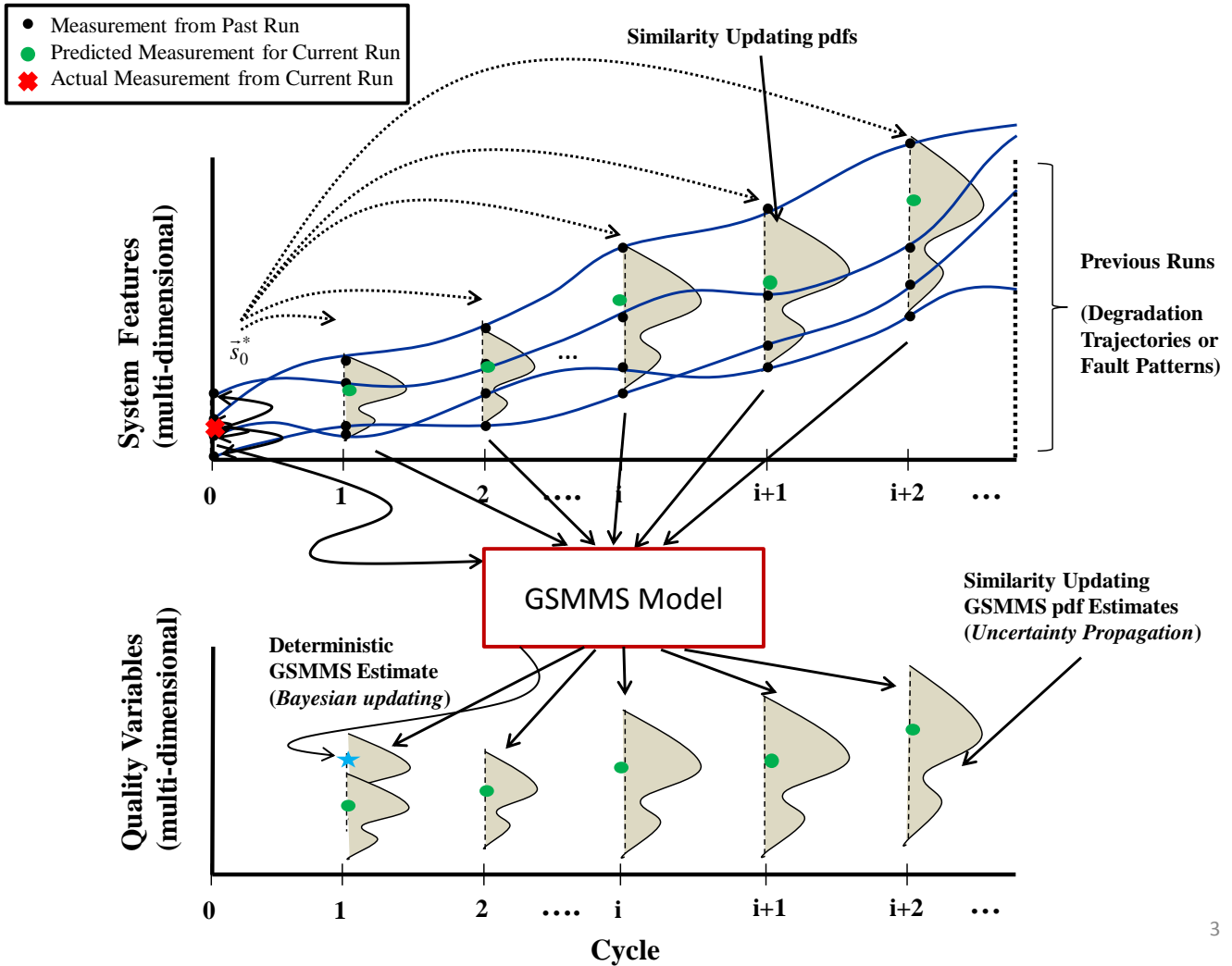


Figure 5.2: The process of using the GSMMS VM model directly to predict the quality measurements, while using uncertainty propagation through the GSMMS to obtain and update long-term prediction distributions of the quality measurements.

As mentioned in the introduction, due to the complexity of manufacturing systems and the single output VM model being pursued in this work, a Monte Carlo sampling based approach is chosen as an appropriate way to obtain the predicted metrology distributions from the predicted equipment feature distributions. Figure 5.3 below illustrates the method for calculating the predicted metrology distributions. First, after the

similarity based prediction algorithm yields the predicted distributions of sensor readings, each distribution is sampled randomly using the MC method. Next, all of these samples are fed through the GSMMS VM model in order to obtain the predicted metrology samples. Finally, each local GSMMS model is associated with region specific noise characteristics, which must be accounted for in order to obtain the proper predicted output distribution. This region specific noise is added to the corresponding samples produced by the GSMMS model via MC sampling from the region specific noise distribution. This process is repeated each time a new equipment signature is extracted from the system and distributions of predicted equipment signatures are updated following the methodology from Chapter 3.

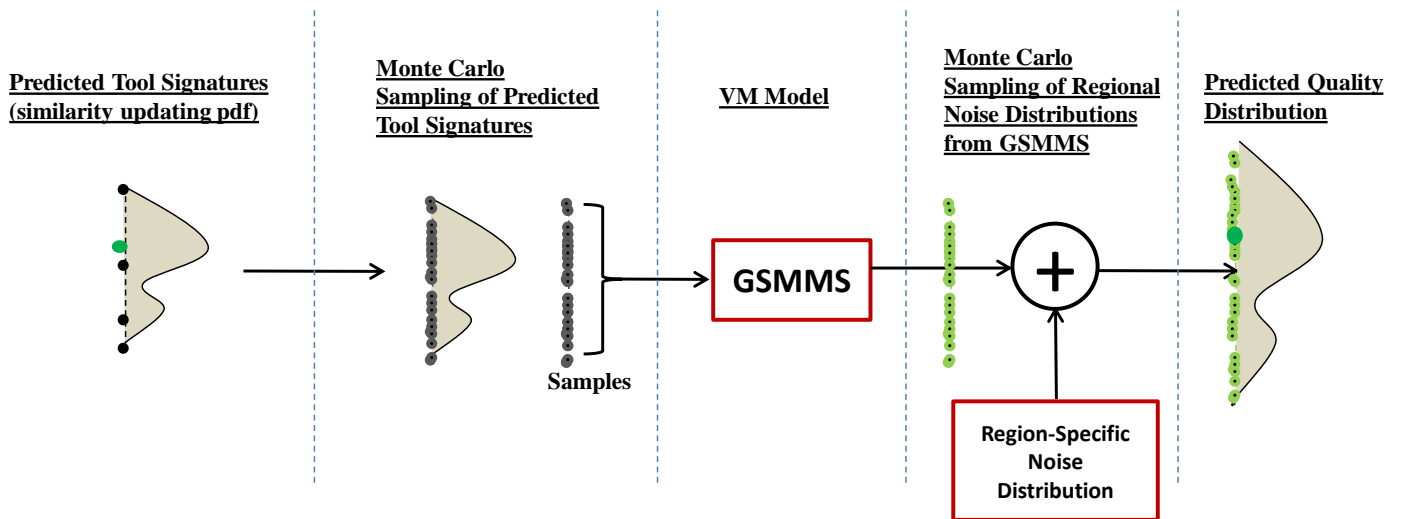


Figure 5.3: Flowchart of the methodology realizing the newly derived predictive VM concept. It uses Monte Carlo random sampling from the predicted tool signature distributions which are then fed through the GSMMS based VM model. To each VM sample, region-specific noise samples are added to produce samples for the predicted quality variables. These steps are performed consecutively each time a new set of

equipment signatures arrives and the predicted tool signatures are updated following the methodology from Chapter 3.

5.2 Results on Applying Integrated Performance Prediction and Quality Estimation to Semiconductor Manufacturing

The analysis presented in this Chapter focuses on implementation of the predictive VM methodology, a study of its ability to predict in the long-term, mean wafer thicknesses throughout maintenance periods, and testing the ability to become more accurate as the current operations progress. As mentioned in the previous section, a *run* is defined a period of time between two consecutive maintenances which consists of many products being processed. Each time a product is processed on the system under study, we refer to that as a *cycle*. When it comes to the PECVD dataset used throughout this dissertation, around 25-50 wafers are processed in between automatic in-situ cleans which are referred to as the cycles and run periods respectively. This terminology may be different from what is established within the semiconductor industry.

The focus on the study presented here was on predicting the mean wafer thicknesses of the approximately 30,000 wafers of the same PECVD recipe, which is the same dataset studied in Chapter 4. The previous Section presented the methodology of using MC simulation to sample the predictive equipment signature distributions and to feed these samples through the GSMMS model while taking into account the local modeling uncertainty in order to obtain the final predicted quality distributions. First, the dataset was organized into the corresponding runs and the first 25 runs were used as training data for GSMMS and for historical trajectories for the similarity based prediction

method. Next, each new run is predicted, simulating the operation of the system as it progresses through the run while at the same time utilizing the dual updating strategy for GSMMS from Chapter 4.

Figure 5.4 shows the behavior of the ideal predictions for a single run that would result from this methodology if everything behaved linearly with zero noise. On the left, the multiple step ahead prediction errors are shown for predicting a single run as if the current cycle is beginning at 1, 10 and 20 respectively. On the right, the same is shown but with the 95% confidence limits on the predictive distributions. From these plots, one can see that the prediction at each future time is worse and has larger confidence intervals as the future time increases. However, as the run progresses, the predictions get better and have higher confidence (smaller 95% confidence limits), along with having better one step ahead predictions. Also, looking at a single cycle being predicted within the run, the prediction of that cycle becomes more accurate as the current run progresses towards it. Obtaining a results along these lines will demonstrate that the predictive VM methodology works and will open the door for future investigation.

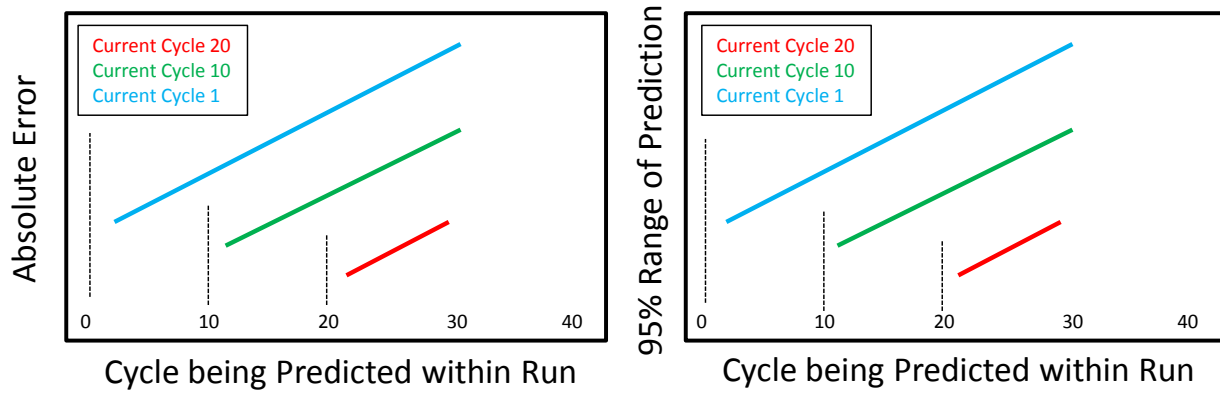


Figure 5.4: Illustration of the ideal linear behavior of the prediction error and range of the prediction distribution as the current run progresses. One can see, at a particular current cycle within the run, as it predicts further, the absolute error and range of the prediction become larger. Also, as the current run progresses from cycle 1 to 10 to 20, the absolute error and range both become smaller, demonstrating the capability of similarity based predictions along with an accurate VM model.

Beginning with the prediction of a single run, Figure 5.5 below shows the predictions of the mean wafer thicknesses coming from the PECVD dataset. On the left, there are three plots showing the predictions starting at cycles 1, 10 and 20 within the run respectively. On the right, is a plot of the one step ahead predictions of this run. One can see the predictions get slightly better as the run progresses and the confidence intervals drop. Also, notice that the majority of the actual metrology lies within the 95% confidence limits of the one step ahead predictions, which is good for applying process control on a wafer to wafer process. Figure 5.6 shows the absolute errors and 95% confidence limits for the predictions in this run in two ways. The top two plots are comparable to Figure 5.4. In reality, however there are only slight increases in the

accuracy of the predictions as the run progresses from cycle 1 to 10 to 20. The bottom two plots show the absolute prediction errors and 95% confidence limits by singling out particular cycles within the run as a function of what cycle the current run has progressed to. These plots show that for the cycles shown, there are consistent drops in the error and confidence limits as the system progresses toward these cycles. These results have a lot of noise but there is a consistent trend.

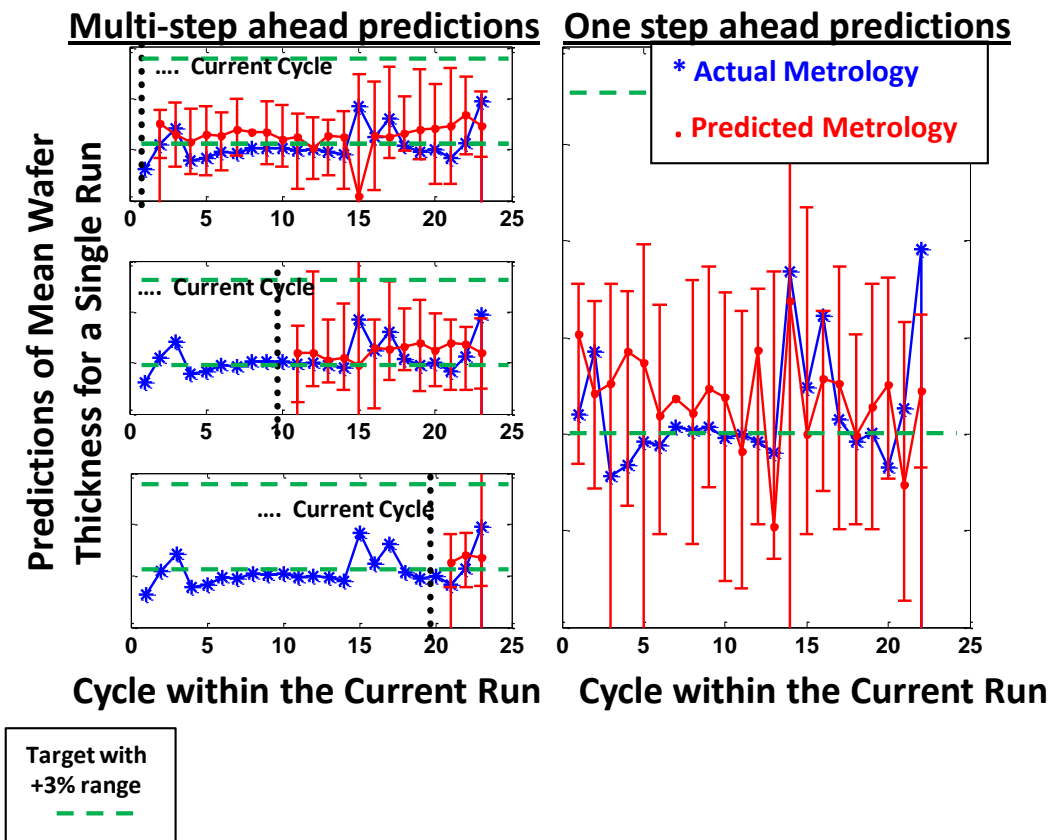


Figure 5.5: Predictions with 95% confidence limits of the mean wafer thicknesses for an entire run based on the integrated predictive VM approach. On the left are the multiple step ahead predictions starting from cycle 1, 10 and 20 respectively. On the right are the one step ahead predictions for the entire run, which is useful in wafer-to-wafer control.

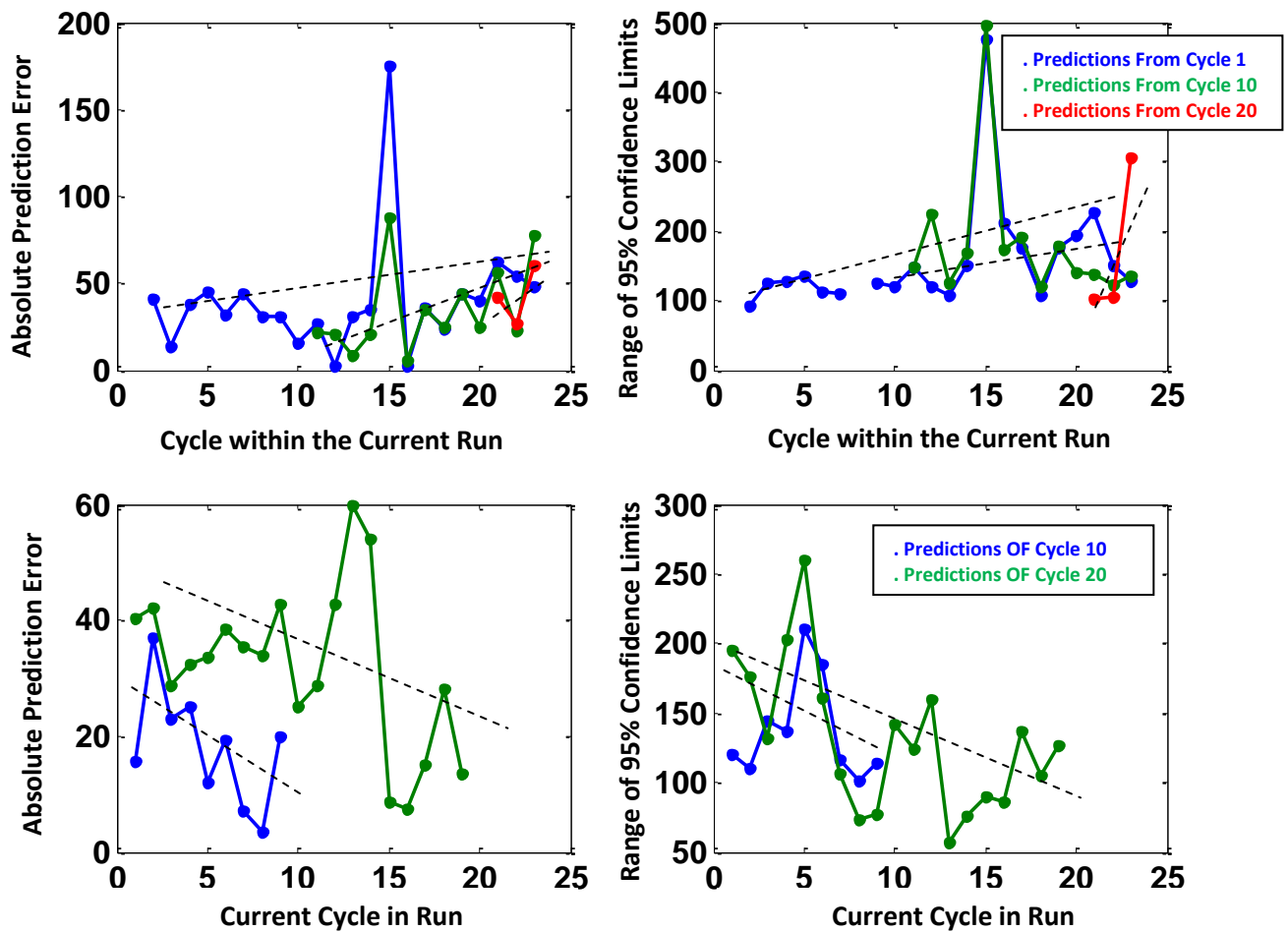


Figure 5.6: Absolute prediction error and 95% confidence limits of prediction distributions as the run in Figure 5.5 progresses. The top two plots show multistep ahead errors and ranges at various snapshots corresponding to Figure 5.5 and comparable to the ideal situation in Figure 5.4. These errors should ideally grow as one predicts forward, but drop as one gathers more information about the current run. The bottom two plots show the predictions and confidence limits as a function of the current cycle, for predicting select cycles within the run. These ideally should drop for every cycle as the run progresses.

Next, around 1000 runs were analyzed for prediction errors and averaged together to determine the overall behavior of the predictive model. Figure 5.7 shows the average predictions over all of the runs, along with the 95% confidence limits for various selected current cycles (starting at 1, 10, 20, and 30). One can see that the actual metrology and the predictions are very close on average and all are within the confidence limits of the predictions. Also, the confidence limits become narrower as the current cycle progresses forward, indicating an average increase in confidence when progressing. Figure 5.8 shows the range of the 95% confidence limits corresponding to Figure 5.7 for the same selected current cycles as before. This Figure clearly shows that the 95% confidence limits drop as the current cycle progresses through the run and the predictions become more confident.

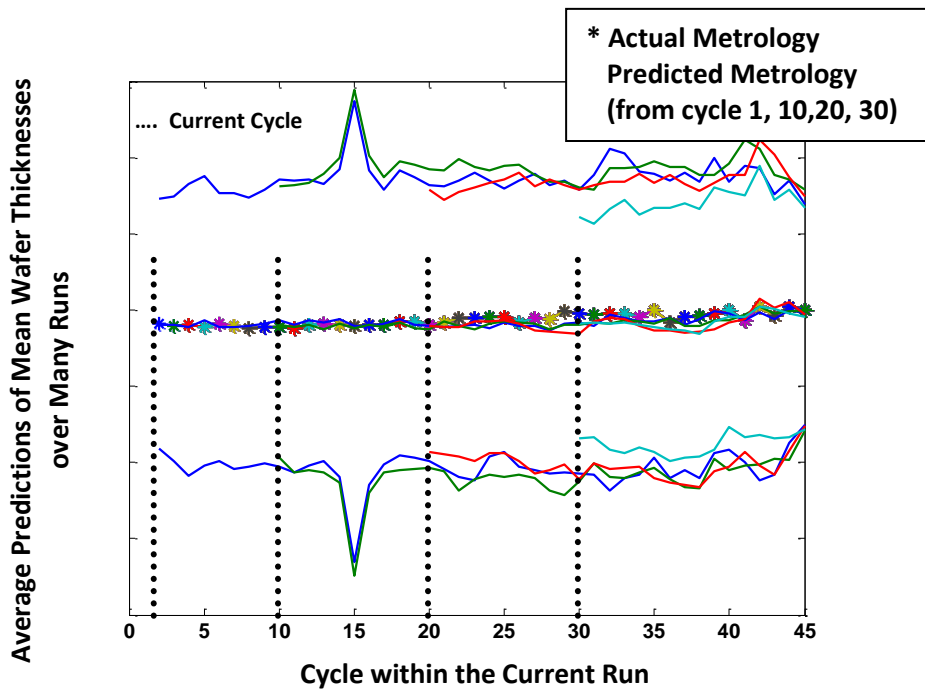


Figure 5.7: Averaged predictions and 95% confidence limits for multi-step ahead predictions starting at current cycle 1, 10, 20, and 30. One can see that on average, the predictions are close to the actual and there is increasing confidence as the run progresses.

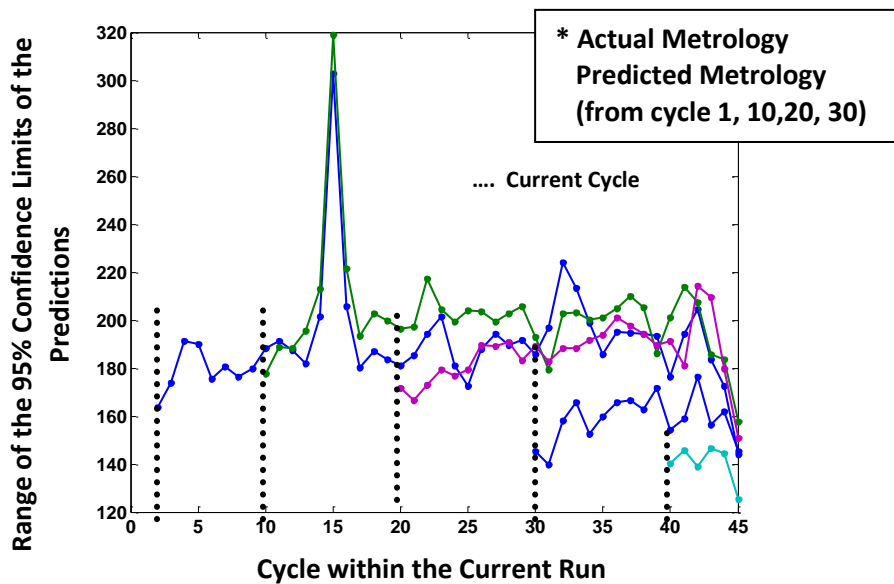


Figure 5.8: Ranges of the 95% confidence limits of multi-step ahead predictions starting from cycle 1, 10, 20 and 30 respectively of the averaged run. One can see the range dropping, indicating a more confidence prediction as the current run progresses.

As another result, we wanted to make sure that on average, the predictions of each cycle within the run should become more accurate as the current cycle moves toward the predicted cycle. This is equivalent to taking a slice out of Figure 5.4 at any predicted cycle and seeing that the prediction error consistently drops as the current cycle move forward. For the average run performed on the PECVD tool, Figure 5.9 below shows that if one is predicting cycles 11, 21, and 30 within the current run, then the prediction error drops as the current cycle moves toward these future cycles. The slopes of these lines are how much accuracy is gained as each new piece of information is gathered from the equipment signatures. One strange occurrence was the predicting cycle 11 was always less accurate than predicting cycle 21, which must have to do with a certain run having bad predictions around cycle 11 which could have been caused by a shift in metrology that GSMMS did not adapt to as of yet. As for all of the cycles being predicted within the run, Figure 5.10 shows the slopes of the prediction errors for every cycle. One can see that after cycle 7, almost every cycle has a negative slope of prediction error versus the current cycle. This indicates that there are consistent downward trends for almost every predicted cycle and that each one of the cycles on average becomes more accurate as the current run progresses towards it. Before cycle 7, there are not enough data points and too much noise to see any consistent trends in the prediction error. All of these results point to the predictive VM methodology seems to be working, even with the large amount of noise within the wafer thicknesses. Finally, Figure 5.11 shows what percentage of the wafers had the actual metrology fall within the predicted metrology 95% confidence limits. One can see that on average over 83% of wafers are falling within these limits,

which is slightly less than 95% and can be contributed to parameter uncertainty within the GSMMS model. Also, this percentage grows as one predicts further due to limited runs actually being that long.

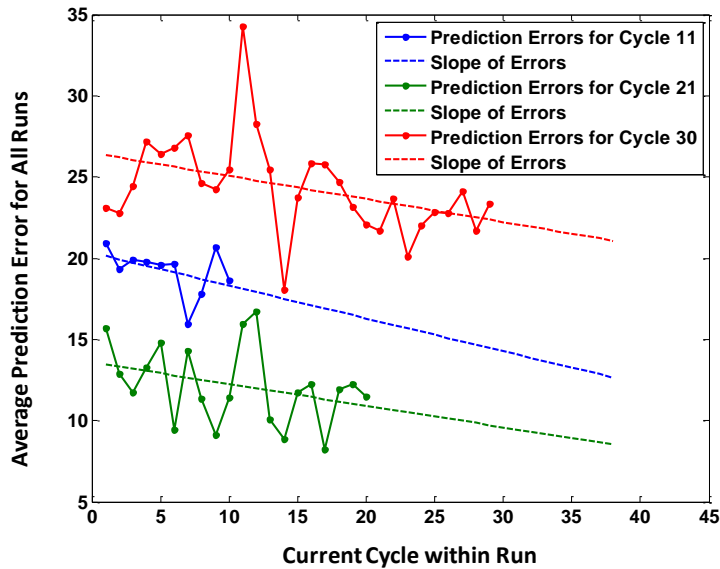


Figure 5.9: Averaged prediction errors of select cycles as a function of the current cycle. One can see the downward slopes, indicating better prediction of these cycles as the current cycle moves forward, on average.

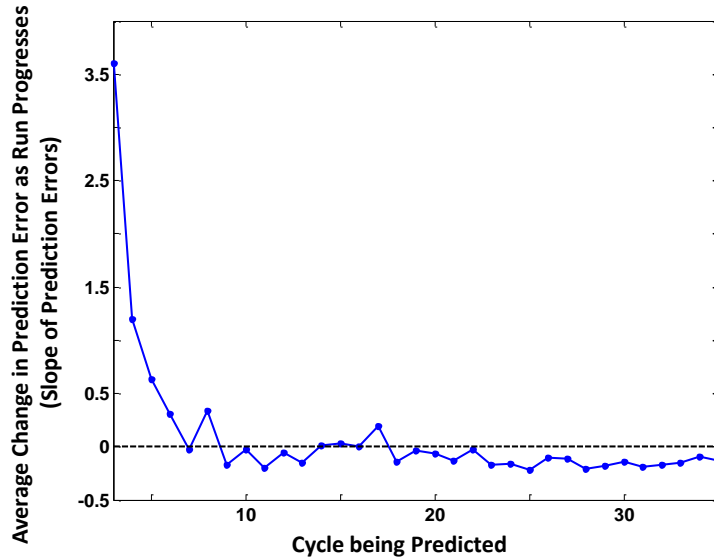


Figure 5.10: Slopes of the averaged prediction errors for all cycles being predicted within a run. The majority of the cycles being predicted have negative slopes, meaning most cycles being predicted within the run have increasing accuracy as the current run progresses on average.

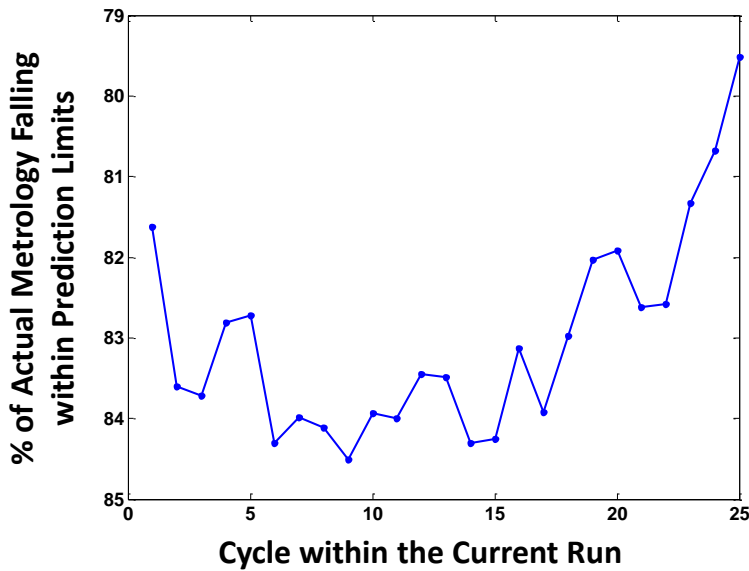


Figure 5.11: Percentage of actual metrology falling within the predicted metrology 95% confidence limits for each cycle being predicted within current runs. 83% of the wafers on average are predicted within the limits.

5.3.1 Conclusions for Integrated Performance Prediction and Quality Estimation in Manufacturing Systems

Chapter 5 presented the integrated predictive VM approach that combines the methodology of Chapters 3 and 4. Monte Carlo simulation was used to estimate predictive distributions of wafer thickness metrology given predictive equipment signature distributions. Results showed that prediction accuracy of the metrology gets better as the current operations progress forward in time. For certain runs, one step ahead predictions were falling within range of the actual metrology, giving good preliminary findings for a wafer to wafer control study. Also, it is shown that the predictive distributions become narrower as the operation progresses while most of the actual metrology on average fall within the prediction bounds. On average, it was finally shown that the predictions for most of the cycles within the run become more accurate as the run progresses forward.

Chapter 6

Summary and Future Work

This Chapter summarizes the specifics that were developed in each of Chapters 3-5 in the context of time-series prediction for predictive condition monitoring, Virtual Metrology (VM) and predictive VM using a combination of novel time series prediction and dynamic VM modeling methods. In Chapter 3, a novel time-series prediction algorithm capable of dealing with a long-term prediction of non-stationary multivariate time-series was presented. The method is based on the concept of similarity-weighted Gaussian mixture models (GMMs) obtained via comparisons of signatures describing the current degradation process with those observed on the same machine/process in the past. The new method was tested in predicting signatures extracted during the operation of an industrial Plasma Enhanced Chemical Vapor Deposition (PECVD) tool. The results showed that the newly developed prediction method yields noticeably smaller mean squared errors (MSE), compared with ARMA based prediction and comparable MSE to another recently introduced similarity-based matrix prediction model. However, its analytical character allows the new prediction method to compute the prediction distributions an order of magnitude faster. Chapter 4 introduced the concept of growing, locally linear dynamic models for VM applications. The newly developed VM method was evaluated for estimating the average wafer thicknesses of over 30,000 wafers from the same PECVD system. The results show that in terms of modeling errors and amount of necessary physical wafer measurements, the newly introduced GSMMS-based VM model outperforms the PLS and T-PLS regression based VM. Furthermore, when

GSMMS based VM model was updated using scheduled measurements and whenever unusual VM inputs appeared, it consistently outperformed PLS-based VM methods, regardless of the metric used to describe model accuracy and regardless of the updating scheme of the VM model. Chapter 5 presented the integrated predictive VM approach that combines the methodologies introduced in Chapters 3 and 4 to enable the prediction of quality characteristics of future products. Monte Carlo simulation was used to estimate predicted distributions of mean wafer thicknesses, given predicted equipment signature distributions. Results showed that prediction accuracy of the metrology gets better as the current operations progress forward in time with most of the actual metrology falling within the prediction confidence bounds.

An avenue for possible future work in time series prediction using similarity based methods is the potential for grouping of degradation trajectories (runs) that have similar evolution dynamics of the time-series of sensory features, in order to reduce the number of degradation trajectories that need to be kept in the historical database and used for predictions. In that context, a new run would be added to the library of past runs only after a degradation trajectory is observed that is “sufficiently different” from the ones seen in the past. Another avenue is to incorporate maintenances that do not bring the system back to a consistent state necessitating a proper alignment of the historical trajectories to accomplish prediction. Namely, if an operation does not come back to a normal condition, the condition drift must be modeled for proper alignment.

As for the GSMMS based VM framework, it can also be adapted in many ways in the future. One can experiment with different local models, such as local PLS with different components in each region. Another approach can be non-linear local models

that can be either derived from known physics within each region or can be a set of trained feed-forward type neural networks. In addition, there can be many modifications to the growth and deletion mechanisms for the SOM underlying the GSMMS VM model. Fuzzy boundaries can also be added to reduce the frequency of model switching, with the output becoming a weighted combination of local models within a neighborhood around the input.

VM model updating is another realm needing investigation within the local modeling framework in future work. For example, the number of instances of unusual inputs that are necessary for requiring a new local model can be optimized or made more sophisticated than what was done in this thesis.

As for the general dynamic modeling and data collection/experimentation within semiconductor manufacturing equipment or any other complex system, it is of highest importance that one captures the proper dynamic features to link the system performance to the quality of the products coming out of the system. This is the most challenging aspect, especially when dealing with such complex physics with many subsystems influencing each other, which can be thermo-fluidic and are not strictly well known physically as of the current state of technology. It is important to extract features from sensors that measure physical quantities that are as closely related to the condition surrounding the product and impacting its quality variables as possible. Therefore, in modern complex systems future work must be done in making these models more accurate in general through a combination of physical knowledge, engineering work, and machine learning.

In the realm of the integrated predictive VM approach demonstrated in Chapter 5, much work must still be done in comparing the methodology to other predictive models. The various types of uncertainty coming from the VM modeling parameters, VM model, and time-series prediction must all be taken into account and estimated correctly for this methodology to work. Another possible avenue for future work is to investigate analytical uncertainty propagation techniques through the GSMMS model rather than the MC based one, which was pursued in this thesis. This would greatly accelerate the computations and potentially improve their accuracy. Gaussian mixtures have a rich analytical background and it could be possible to obtain analytical results for propagating Gaussian mixtures through the local linear GSMMS model. Finally, the predictive VM methodology should be tested for one step ahead predictions and determined its feasibility for implementing run-to-run control (wafer to wafer control). Robust control methods can quantify performance within the predicted bounds of control and seem as a plausible way to attach this problem.

6.1 Scientific Contributions

First, the similarity based performance prediction methodology has the ability to increase the accuracy of any long-term prediction scheme for any system that can be placed into the similarity-based paradigm. Systems that undergo repeated and frequent maintenance between which production operations take place are obvious candidates for this methodology (e.g. thin film depositions with repeated in-situ cleaning, metal cutting operations with repeated tool replacements, etc...). Additionally, the newly introduced WLE approach to modifying the GMM likelihood function with the similarity weights,

which was adapted from [140] led to tremendous gains in terms of computation times for predicted distributions.

Second, the GSMMS VM model provides significant scientific contributions to the widely utilized quality estimation and virtual sensing literature. It simultaneously enables model validity checking and high model accuracy, while keeping a mathematically tractable and accurate framework. The result showing its advantages over global type modeling can potentially change the way VVM models are pursued in the future.

Finally, the integration of performance prediction and quality estimation provides many scientific contributions to manufacturing research and practice. Performance prediction and VM quality estimation have never been integrated before in literature, and their integration in this work enables long-term system performance predictions in manufacturing.

6.2 Publications

The publications already produced or anticipated to be produced based on this doctoral research are as follows:

- **Bleakie, A., Djurdjanovic, D., 2011**, “Dynamic Feature Monitoring Technique Applied to Thin Film Deposition Processes in an Industrial PECVD Tool,” Proceedings of the ASME 2011 International Manufacturing Science and Engineering Conference, June 13-17.
- **Bleakie, A., Djurdjanovic, D., 2013**, “Feature Extraction, Condition Monitoring, and Fault Modeling in Semiconductor Manufacturing Systems,” *Computers in Industry*, 64(3), pp. 203-213.
- **Bleakie, A., Djurdjanovic, D., 2013**, “Analytical Approach to Similarity Based Prediction of Manufacturing System Performance,” *Computers in Industry*, 64(6), pp. 625-633.

- **Bleakie, A., Djurdjanovic, D., 2014**, "Growing Structure Multiple Model System for Quality Estimation in Manufacturing Processes," submitted to IEEE Transactions on Semiconductor Manufacturing.
- **Bleakie, A., Djurdjanovic, D., 2014**, "Integrated Performance Prediction and Quality Control of Manufacturing Systems," to be submitted to IEEE Transactions on Semiconductor Manufacturing.

Appendix

The following tables show the results of the analysis described in this thesis and shown in Figures 4.9- 4.11.

Table A1: Results of entire analysis of PLS, T-PLS and GSMMS showing the frequency of scheduled measurements, total number of scheduled measurements, MSE, number of model updates based on unusual inputs, and number of model updates for both updating schemes analyzed in this thesis.

Schedule Frequency	Total Scheduled Measurements	PLS Output Updating Only			PLS Both Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
1/8	3985	81.0	0	630	85.7	648	704
1/25	1275	85.2	0	204	89.7	598	212
1/50	638	87.9	0	107	92.0	582	107
1/75	425	87.0	0	87	93.5	583	103
1/100	319	90.4	0	53	94.9	576	59
1/125	255	94.8	0	41	95.5	575	45
1/150	213	89.4	0	40	95.6	576	52
1/175	182	92.5	0	28	97.8	573	33
1/200	159	91.9	0	25	97.4	574	29

Schedule Frequency	Total Scheduled Measurements	T-PLS Output Updating Only			T-PLS Both Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
1/8	3985	83.0	0	667	84.6	187	693
1/25	1275	86.9	0	211	86.7	168	208
1/50	638	89.5	0	105	87.9	153	102
1/75	425	89.0	0	91	88.3	139	91
1/100	319	92.0	0	53	89.4	150	54
1/125	255	95.8	0	44	90.7	138	40
1/150	213	89.5	0	41	89.7	137	45
1/175	182	93.0	0	28	91.3	136	31
1/200	159	92.5	0	27	94.7	139	29

Schedule Frequency	Total Scheduled Measurements	GSMMS Output Updating Only			GSMMS Both Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates

1/8	3985	80.3	0	618	79.8	21	618
1/25	1275	84.0	0	193	82.9	23	184
1/50	638	86.7	0	101	83.5	23	100
1/75	425	87.4	0	87	84.6	21	82
1/100	319	90.0	0	50	86.7	23	52
1/125	255	93.3	0	42	88.0	22	42
1/150	213	89.1	0	41	87.9	24	42
1/175	182	93.2	0	28	90.1	20	30
1/200	159	91.9	0	24	88.3	23	21

Table A2: Percentage of wafers that had relative VM errors less than 1% and less than 3%, along with the maximal relative error.

PLS									
	Scheduled Updating Only			Abnormal Input Updating Only			Both Updating Types		
Period	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error
1/8	84.90	100.00	2.98	75.50	99.99	3.70	83.10	99.99	3.21
1/25	83.60	99.98	4.76	75.50	99.99	3.70	81.50	99.99	3.31
1/50	82.50	99.93	4.75	75.50	99.99	3.70	80.30	99.99	3.43
1/75	82.90	99.92	4.17	75.50	99.99	3.70	79.50	99.99	3.45
1/100	81.70	99.81	6.39	75.50	99.99	3.70	78.66	99.99	3.51
1/125	79.70	99.81	6.10	75.50	99.99	3.70	78.15	99.99	3.52
1/150	82.30	99.79	5.56	75.50	99.99	3.70	78.20	99.99	3.59
1/175	81.30	99.73	5.87	75.50	99.99	3.70	76.95	99.99	3.66
1/200	81.40	99.80	6.40	75.50	99.99	3.70	77.20	99.99	3.61

T-PLS									
	Scheduled Updating Only			Abnormal Input Updating Only			Both Updating Types		
Period	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error
1/8	84.20	99.99	3.14	80.22	99.99	3.59	83.35	99.99	3.27
1/25	82.80	99.98	4.83	80.22	99.99	3.59	82.66	99.99	3.29
1/50	81.52	99.93	4.69	80.22	99.99	3.59	82.18	99.98	3.24
1/75	82.14	99.91	4.31	80.22	99.99	3.59	81.77	99.99	3.24
1/100	81.18	99.82	6.37	80.22	99.99	3.59	81.64	99.99	3.44
1/125	79.12	99.81	6.09	80.22	99.99	3.59	80.88	99.99	3.45
1/150	82.18	99.80	5.41	80.22	99.99	3.59	81.12	99.99	3.48
1/175	81.10	99.71	5.86	80.22	99.99	3.59	80.52	99.98	3.75
1/200	81.00	99.81	6.41	80.22	99.99	3.59	79.19	99.92	3.84

Divide and Conquer (GSMMS)									
Period	Scheduled Updating Only			Abnormal Input Updating Only			Both Updating Types		
	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error	Less than 1%	Less than 3%	Max Error
1/8	85.40	100.00	2.94	74.90	99.99	3.53	85.30	100.00	2.97
1/25	84.10	99.98	4.76	74.90	99.99	3.53	84.30	100.00	2.86
1/50	82.90	99.94	4.71	74.90	99.99	3.53	83.80	99.99	3.01
1/75	83.00	99.93	4.16	74.90	99.99	3.53	83.50	100.00	2.99
1/100	81.20	99.82	6.42	74.90	99.99	3.53	82.70	99.99	3.16
1/125	80.70	99.82	6.09	74.90	99.99	3.53	81.90	99.99	3.19
1/150	82.50	99.80	5.56	74.90	99.99	3.53	82.90	99.99	3.26
1/175	81.20	99.69	5.86	74.90	99.99	3.53	81.98	99.99	3.39
1/200	81.10	99.81	6.40	74.90	99.99	3.53	82.40	99.99	3.23

Table A3: Results of entire analysis of PLS, T-PLS and GSMMS showing the frequency of scheduled measurements, total number of scheduled measurements, MSE, number of model updates based on unusual inputs, and number of model updates for both updating schemes analyzed in this thesis. This is for the *100 Wafer Moving window* result.

Schedule Freq	Total Scheduled Measurements	PLS Output Only			PLS Mixed Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
8	3985	83.56	0	671	83.93	329	674
25	1275	86.66	0	217	86.88	343	211
50	638	92.29	0	110	88.53	375	100
75	425	88.96	0	95	90.80	383	80
100	319	93.48	0	52	90.07	395	54
125	255	97.04	0	43	89.73	395	45
150	213	92.74	0	46	91.67	400	42
175	182	96.76	0	30	91.36	409	27
200	159	95.81	0	31	91.77	409	25
Schedule	Total Scheduled	TPLS Output Only			TPLS Mixed		

Freq	Measurements				Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
8	3985	84.09	0	679	83.28	204	661
25	1275	89.00	0	214	87.56	145	219
50	638	92.39	0	114	90.49	110	107
75	425	91.97	0	95	92.19	130	104
100	319	94.09	0	51	88.09	90	55
125	255	96.60	0	43	91.81	92	37
150	213	91.97	0	45	91.04	87	47
175	182	95.65	0	30	90.19	86	30
200	159	95.28	0	29	89.61	89	23
Separator							
Schedule Freq	Total Scheduled Measurements	GSMMS Output Only			GSMMS Mixed Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
8	3985	80.33	0	618	79.89	21	618
25	1275	84.05	0	193	83.00	23	184
50	638	86.70	0	101	83.58	23	100
75	425	87.48	0	87	84.65	21	82
100	319	90.09	0	50	86.72	23	52
125	255	93.32	0	42	88.05	22	42
150	213	89.16	0	41	87.93	24	42
175	182	93.24	0	28	90.11	20	30
200	159	91.97	0	24	88.39	23	21
Separator							
Schedule Freq	Total Scheduled Measurements	RBNN Output Only			RBNN Mixed Updating		
		MSE	# Input Updates	# Output Updates	MSE	# Input Updates	# Output Updates
8	3985	87.06	0	760	85.97	311	760
25	1275	89.12	0	231	92.57	344	253
50	638	89.91	0	107	95.97	368	126
75	425	90.35	0	93	95.40	377	110
100	319	92.60	0	58	98.74	383	81
125	255	94.96	0	47	95.52	399	56
150	213	98.19	0	52	97.38	397	56
175	182	99.71	0	41	99.87	398	39

200	159		99.51	0	33		103.5 3	402	37
-----	-----	--	-------	---	----	--	------------	-----	----

Table A4: Percentage of wafers that had relative VM errors less than 1% and less than 3%, along with the maximal relative error for the *100 Wafer Moving Window* results.

PLS									
	Scheduled Updating Only			Abnormal Input Updating Only			Both Updating Types		
Period	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error
8	84.21	99.99	3.27	79.52	99.99	3.19	84.02	100.00	2.89
25	82.79	99.96	5.07	79.52	99.99	3.19	82.20	99.99	3.13
50	80.45	99.88	5.58	79.52	99.99	3.19	81.60	99.99	3.21
75	82.09	99.94	3.91	79.52	99.99	3.19	80.35	99.99	3.18
100	80.34	99.80	6.68	79.52	99.99	3.19	80.81	99.99	3.24
125	78.57	99.84	5.30	79.52	99.99	3.19	81.03	99.99	3.02
150	80.66	99.78	5.60	79.52	99.99	3.19	80.18	99.99	3.16
175	79.45	99.73	5.48	79.52	99.99	3.19	80.58	99.99	3.10
200	79.15	99.86	5.96	79.52	99.99	3.19	80.34	99.99	3.33
Average:	80.86	99.86	5.21	79.52	99.99	3.19	80.89	99.99	3.17
TPLS									
	Scheduled Updating Only			Abnormal Input Updating Only			Both Updating Types		
Period	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error
8	83.79	99.99	3.28	79.86	99.99	3.47	84.25	99.99	3.83
25	81.62	99.96	5.10	79.86	99.99	3.47	81.58	99.99	3.46
50	80.40	99.89	5.43	79.86	99.99	3.47	80.54	100.00	2.98
75	80.72	99.89	4.08	79.86	99.99	3.47	79.34	99.99	3.52
100	80.35	99.78	6.71	79.86	99.99	3.47	81.77	99.98	3.34
125	78.77	99.85	5.34	79.86	99.99	3.47	80.17	99.98	3.20
150	80.81	99.86	5.08	79.86	99.99	3.47	80.37	99.98	3.56

175	79.52	99.77	5.05	79.86	99.99	3.47	80.96	99.98	3.54
200	79.29	99.86	5.91	79.86	99.99	3.47	81.56	99.97	3.44
Average:	80.59	99.87	5.11	79.86	99.99	3.47	81.17	99.98	3.43
Divide and Conquer (GSMMS)									
Scheduled Updating Only									
Abnormal Input Updating Only									
Both Updating Types									
Period	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error
8	85.40	100.00	2.94	74.90	99.99	3.53	85.30	100.00	2.97
25	84.10	99.98	4.76	74.90	99.99	3.53	84.30	100.00	2.86
50	82.90	99.94	4.71	74.90	99.99	3.53	83.80	99.99	3.01
75	83.00	99.93	4.16	74.90	99.99	3.53	83.50	100.00	2.99
100	81.20	99.82	6.42	74.90	99.99	3.53	82.70	99.99	3.16
125	80.70	99.82	6.09	74.90	99.99	3.53	81.90	99.99	3.19
150	82.50	99.80	5.56	74.90	99.99	3.53	82.90	99.99	3.26
175	81.20	99.69	5.86	74.90	99.99	3.53	81.98	99.99	3.39
200	81.10	99.81	6.40	74.90	99.99	3.53	82.40	99.99	3.23
Average:	82.46	99.87	5.21	74.90	99.99	3.53	83.20	99.99	3.12
RBNN									
Scheduled Updating Only									
Abnormal Input Updating Only									
Both Updating Types									
Period	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error	Percent Over 1%	Percent Over 3%	Max Error
8	83.43	99.97	3.57	75.39	100.00	2.97	84.10	99.98	3.45
25	81.99	99.98	5.26	75.39	100.00	2.97	80.00	99.99	3.53
50	81.60	99.98	4.45	75.39	100.00	2.97	78.44	99.97	3.48
75	81.67	99.86	5.24	75.39	100.00	2.97	78.28	99.97	3.48
100	80.65	99.91	4.20	75.39	100.00	2.97	76.99	99.97	3.48
125	80.40	99.80	5.37	75.39	100.00	2.97	78.26	99.99	3.45
150	78.03	99.82	4.55	75.39	100.00	2.97	77.09	99.98	3.17
175	77.59	99.84	4.68	75.39	100.00	2.97	75.97	99.96	3.47
200	77.73	99.88	4.07	75.39	100.00	2.97	75.44	99.88	3.45
Average:	80.34	99.89	4.60	75.39	100.00	2.97	78.29	99.97	3.44

Bibliography

- [1] Heng , A, S. Zhang, A. Tan, J. Mathew, 2009, “Rotating Machinery Prognostics: State of the Art, Challenges and Opportunities,” *Mech. Sys. and Sig. Proc.*, Vol. 23, pp. 724–739.
- [2] Ferret, 2009, “The Basics of Predictive/Preventive Maintenance,” <http://www.ferret.com.au/articles/ad/0c0259ad.asp>.
- [3] Hao, S., 2010, “I-35W Bridge Collapse,” *Journal of Bridge engineering*, 15, pp. 608-614.
- [4] Kerr, R., and Kintisch, E., 2010, “Will Deepwater Horizon Set a New Standard for Catastrophe?,” *Science Magazine*, 328, pp. 674-675.
- [5] Oien, K., Schjolberg, P., Merland, O., Leto, S., and Spilde, H., 2010, “Correct Maintenance Prevents Major Accidents,” *MaintWorld*, 2, pp. 26-28.
- [6] Lind, S., 2008, “Types and Sources of Fatal and Severe Non-fatal Accidents and Industrial Maintenance,” *International Journal of Industrial Ergonomics*, 38, pp. 927-933.
- [7] Gertler, J., 1998, *Fault Detection and Diagnosis in Engineering Systems*, Marcel Dekker, New York, Ch.1.
- [8] Djurdjanovic, D., Lee, J., and Ni, J., 2003, “Watchdog Agent -An Infotronics Based Prognostics Approach for Product Performance Assessment and Prediction,” *International Journal of Advanced Engineering Informatics, Special Issue on Intelligent Maintenance Systems*, (17)3-4, pp. 109-125.
- [9] Jardine, A., Lin, D., and Banjevic, D., 2006, “A Review on Machinery Diagnostics and Prognostics Implementing Condition Based Maintenance,” *Mechanical Systems and Signal Processing*, 20, pp. 1483-1510.
- [10] Djurdjanovic, D., and Liu, Y., 2006, "Survey of Predictive Maintenance Research and Industry Best Practice," University of Michigan, Ann Arbor, MI.
- [11] Hu, J., Zhang, L., Ma, L., and Liang, W., 2010, “An Integrated Method for Safety Pre-Warning of Complex System,” *Safety Science*, 48, pp. 580-597.
- [12] Hu, J., Zhang, L., and Liang, W., 2012, “An Adaptive Online Safety Assessment Method for Mechanical System with Pre-warning Function,” *Safety Science*, (3)50, pp. 385-399.
- [13] You, M., Li, L., Meng, G., and Ni, J., 2010, “Cost-Effective Updated Sequential Predictive Maintenance Policy for Continuously Monitored Degrading Systems,” *IEEE Transactions on Automation Science and Engineering*, (2) 7, pp. 257-265.
- [14] Liu, J., Djurdjanovic, D., Ni, J., Casotto, N., and Lee, J., 2007, “Similarity Based Method for Manufacturing Process Performance Prediction and Diagnosis,” *Computers in Industry*, (6)58, pp. 558-566.

- [15] Peggs, G., 2003, "Virtual Technologies for Advanced Manufacturing and Metrology," *International Journal of Computer Integrated Manufacturing*, Vol. 16, 7-8, pp. 485-490.
- [16] Fortuna, L., Graziani, S., and Xibilia, M., 2005, "Virtual Instruments in Refineries," *IEEE Instrumentation and Measurement Magazine*, Oct., pp. 26-34.
- [17] Khan, A., Moyne, J., Tilbury, D., 2007, "An Approach for Factory-Wide Control Utilizing Virtual Metrology," *IEEE Transaction on Semiconductor Manufacturing*, 20, pp. 364-375.
- [18] Su, Y., Hung, M., Cheng, F., and Chen, Y., 2006, "A Processing Quality Prognostics Scheme for Plasma Sputtering in TFT LCD Manufacturing," *IEEE Trans. on Semiconductor Manufacturing*, 19(2), pp. 183-194.
- [19] Khan, A., Moyne, J., Tilbury, D., 2008, "Virtual Metrology and Feedback Control for Semiconductor Manufacturing Processes using Recursive PLS," *Journal of Process Control*, 18, pp. 961-974.
- [20] Chang, Y.C., and Cheng, F.T., 2005, "Application Development of Virtual Metrology in Semiconductor Industry," *Proceedings of IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, pp. 50-60.
- [21] Liu, L., Kuo, S., and Zhou, M., 2009, "Virtual Sensing Techniques and their Applications," *IEEE Conference on Networking, Sensing and Control*, pp. 31-36.
- [22] Albertos, P., and Goodwin, G., 2002, "Virtual Sensors for Control Applications," *Annual Reviews in Control*, 26, pp. 101-112.
- [23] Bustillo, A., Correa, M., and Renones, A., 2011, "A Virtual Sensor for Online Fault Detection of Multitooth-Tools," *Journal of Sensors*, 11, pp. 2773-2795.
- [24] Bleakie, A., Djurdjanovic, D., 2012, "Analytical Approach to Similarity Based Prediction of Manufacturing System Performance," *Computers in Industry*, 64(6), pp. 625-633.
- [25] Bleakie, A., Djurdjanovic, D., 2014, "Growing Structure Multiple Model System for Quality Estimation in Manufacturing Processes," Submitted to *IEEE Journal of Semi Manufacturing*.
- [26] Serway, R., and Jewett, J., 2010, *Physics for Scientists and Engineers*, Brooks/Cole Publishers (8).
- [27] Harris, R., 1975, *A Primer of Multivariate Statistics*, New York: Academic Press.
- [28] Pandit, S. M. and Wu, S. M., 1983, *Time Series and System Analysis with Applications*, John Wiley.
- [29] Kalman, R., 1960, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME Journal of Basic Engineering*, 82, pp. 34-45.
- [30] Duda, R., Hart, P., and Stork, D., 2001, *Pattern Classification*, Wiley & Sons, NY, (2), pp. 117-124.

- [31] Geva, A., 1999, "Non-stationary Time-series Prediction using Fuzzy Clustering," Presented at Proceedings of NAFIPS-99: 18th International Conference of the North American Fuzzy Information Processing Society, New York.
- [32] Lee, C., Liu, A., and Chen, W., 2006, "Pattern Discovery of Fuzzy Time Series for Financial Prediction," IEEE Transactions on Knowledge and Data Engineering, (18), pp. 613–625.
- [33] Bishop, C., 2006, *Pattern Recognition and Machine Learning*, Springer Text (1).
- [34] Renaud, O., Starck, J., and Murtagh, F., 2005, "Wavelet-based Combined Signal Filtering and Prediction," IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, (35), pp. 1241–1251.
- [35] Cristianini, N., and Shawe-Taylor, J., 2000, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge.
- [36] Zivkovic, Z., and Heijden, F., 2004, "Recursive Unsupervised Learning of Finite Mixture Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, (5)26, pp. 651-656.
- [37] Lindsay, B., 1995, *Mixture Models: Theory, Geometry and Applications*, Institute of Mathematical Statistics, CA, pp. 1-6.
- [38] Haykin, S., 1999, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Upper Saddle River, NJ, (2).
- [39] Han, M., Xi, J., Xu, S., and Yin, F., 2004, "Prediction of Chaotic Time Series Based on the Recurrent Predictor Neural network," IEEE Transactions on Signal Processing, (52), pp. 3409-3416.
- [40] Connor, J., Martin, R., Atlas, L., 1994, "Recurrent Neural Networks and Robust Time Series Prediction," IEEE Transactions on Neural Networks, (5), pp. 240-253.
- [41] Lynn, S., Ringwood, J., and MacGearailt, N., 2012, "Global and Local Virtual Metrology Models for a Plasma Etch Process," IEEE Transactions on Semiconductor Manufacturing, Vol. 25, 1, pp. 94-103.
- [42] Yan, J., Guo, C., and Wang, X., 2011, "A Dynamic Multi-scale Markov Model Based Methodology for Remaining Life Prediction," Mechanical Systems and Signal Processing, 25, pp. 1364-1376.
- [43] Sun, J., Zuo, H., Wang, W., and Pecht, M., 2012, "Application of a State Space Modeling Technique to System Prognostics Based on a Health Index for Condition Based Maintenance," Mechanical Systems and Signal Processing, pp.1-12.
- [44] Ferreira, E., 1995, "Recurrent Neural network Models of Multivariable Dynamic Systems: Numerical Methods and Experimental Evaluation," Masters Thesis, Carnegie Melon University Department of Electrical and Computer Engineering.

- [45] Bushman, S., Edgar, T., and Trachtenberg, I., 1997, "Modeling of Plasma Etch Systems Using Ordinary Least Squares, Recurrent Neural Network, and Projection Latent Structure Models," *Journal of Electrochemical Society*, 155(4), pp. 1379-1389.
- [46] Wen, G., and Zhang, X., 2004, "Prediction Method of Machinery Condition Based on Recurrent Neural Networks Models," *Journal of Applied Sciences*, 4(4), pp. 675-679.
- [47] Xu, Q., Krishnamurthy, K., Mcmillin, B., and Lu, W., 1994, "Identification of Cutting Force in End Milling Operation using Recurrent Neural Networks," *IEEE*, pp. 3828-3833.
- [48] Yu, G., Qiu, H., Djurdjanovic, D., and Lee, J., 2005, "Feature Signature Prediction of a Boring Process using Neural Network Modeling with Confidence Bounds," *International Journal of Advanced Manufacturing Technology*.
- [49] Tian, Z., and Zuo, M., 2009, "Health Condition Prognostics of Gears Using a Recurrent Neural Network Approach," *IEEE conference*, pp. 1-6.
- [50] Tian, Z., and Zuo, M., 2010, "Health Condition Prediction of Gears Using a Recurrent Neural Network Approach," *IEEE Transactions on Reliability*, 59(4), pp. 700-705.
- [51] Yang, H., and Ni, J., 2005, "Dynamic Neural Network Modeling for Nonlinear, Nonstationary, Machine Tool Thermally Induced Error," *International Journal of Machine Tools and Manufacture*, 45, pp. 455-465.
- [52] Huang, Y., Cheng, F., and Chen, Y., 2006, "Importance of Data Quality in Virtual Metrology," *IEEE*, pp. 3727-3732
- [53] Yu, J., 2011, "Bearing Performance Degradation Assessment using Locality Preserving Projections," *Expert Systems with Applications*, 38, pp. 7440-7450.
- [54] Kim, H. E., Tan, A.C.C, and Mathew, J., 2011, "New Machine Prognostics Approach Base on Health State Probability Estimation," *Australian Journal of Mechanical Engineering*, (2)8, pp. 79-89.
- [55] Yang, Y., Liao, Y., Meng, G., and Lee, J., 2011 "A Hybrid Feature Selection Scheme for Unsupervised Learning and its Application in Bearing Fault Diagnosis," *Expert Systems with Applications*, 38, pp. 11311-11320.
- [56] Wang, T. Yu, J., Siegel, D., and Lee, J., 2008, "A Similarity-Based Prognostics Approach for Remaining Useful Life Estimation of Engineered Systems," *International Conference on Prognostics and Health Management*, pp. 1-6.
- [57] Ming-Yi, Y., and Meng, G., 2011, "Updated Proportional Hazards Model for Equipment Residual Life Prediction," *International Journal of Quality and Reliability Management*, (7)28, pp. 781-795.
- [58] Li, J. C., 2006, "Signal processing in manufacturing monitoring," *Condition Monitoring and Control for Intelligent Manufacturing*, pp. 245- 265.

- [59] Yu, J., 2011, "Bearing Performance Degradation Assessment using Locality Preserving Projections and Gaussian Mixture Models," *Mechanical Systems and Signal Processing*, pp. 1-16.
- [60] You, M., and Meng, G., 2011, "Updated Proportional Hazard Model for Equipment Residual Life Prediction," *Journal of Quality and Reliability Management*, Vol. 28, 7, pp. 781-795.
- [61] Hughes, E., Wilson, A., and Peggs, G., 2000, "Design of a High Accuracy CMM Based on Multi-Lateration Techniques," *CIRP Annals-manufacturing technology*, 49 (1), pp. 391-394.
- [62] Chanal, H., Duc, E., Ray, P., and Hascoet, J., 2007, "A New Approach for the Geometrical Calibration of Parallel Kinematics Machines Tools based on the Machining of a Dedicated Part," *International Journal of Machine Tools and Manufacture*, 47 (7-8), pp. 1151-1163.
- [63] Albiero, D., Maciel, A., 2012, "Proposal of Using Exponentially Weighted Moving Average for Studies about Longitudinal Distribution of Seeds," *Revista Ciencia Agronomica, Artigo Cientifico*, 43, pp. 86-95.
- [64] Suh, C., Glynn, S., Scharf, J., Contreras, M., Noufi, R., and Jones, W., 2011, "Exploring High Dimensional Data Space: Identifying Optimal Process Conditions in Photovoltaics," *IEEE Photovoltaic Specialists Conference*, pp. 1-6.
- [65] Yang, Y., Chou, J., Juang, W., Fu, T., and Li, G., 2012, "An Artificial Neural Network for Predicting the Friction Coefficient of Deposited CrAlC Films," *Applied Soft Computing*, 13, pp. 109-115.
- [66] Wilson, E., 1997, "Virtual Sensor Technology for Process Optimization," *Presentation in Symposium on Computers and Controls in the Metals Industry in Iron and Steel Society*.
- [67] Moreau, D., Cazzaloto, B., Zander, A., and Peterson, C., 2008, "A Review of Virtual Sensing Algorithms for Active Noise Control," *Algorithms*, pp. 69-99.
- [68] Peterson, C., Fraanje, R., Cazzoloto, B., Zander, A., and Hansen, C., 2008, "A Kalman Filter Approach to Virtual Sensing for Active Noise Control," *Mechanical Systems and Signal Processing*, 22, pp. 490-508.
- [69] Kestell, C., Hansen, C., and Cazzoloto, B., 2001, "Virtual Sensors in Active Noise Control," *Acoustics Australia*, Vol. 29, 2, pp. 57-61.
- [70] Petersen, C., Cazzoloto, B., Zander, A., Hansen, C., 2006, "Active Noise Control at a Moving Location using Virtual Sensing," *13th Annual Congress of Sound and Vibration*, pp. 1-8.
- [71] Bellas, F., Becerra, J., Santos, J. and Duro, R., 2000, "Applying Synaptic Delays for Virtual Sensing and Actuation in Mobile Robots," in *Proc. of the IEEE International Joint Conference on Neural Networks*, Vol. 6, pp.144-149.
- [72] Cheng, J., Chao, T., Chang, L., and Huang, B., 2004, "A Model-Based Virtual Sensing Approach for the Injection Molding Process," *Polymer Engineering and Science*, Vol. 44, 9, pp. 1605-1614.

- [73] Guerrero, L., Gutierrez, J., and Jalisco, C., 2007, "Simulation Study of the SAR Imaging with the Virtual Remote Sensing Laboratory," in Proc. of Electronics, Robotics and Automotive Mechanics Conf, pp. 563-567.
- [74] Srivastava, A., Oza, N., Stroeve, J., 2005, "Virtual Sensors: Using Data Mining Techniques to Efficiently Estimate Remote Sensing Spectra," IEEE Transactions on Geoscience and Remote Sensing, Vol. 43, 3, pp. 590-600.
- [75] Prokhorov, D., 2005, "Virtual Sensors and their Automotive Applications," IEEE Conference on Intelligent Systems, pp. 411-416.
- [76] Maggiore, M., Ordonez, R., Passino, K., Adibhatla, S., 2003, "Estimator Design in Jet Engine Applications," Engineering Applications of Artificial Intelligence, 16, pp. 579-593.
- [77] Yu, Y., 2012, *Model Based Multivariate Control of Conditioning Systems for Office Buildings*, PhD Dissertation, Carnegie Mellon.
- [78] Fortuna, L., Graziani, S., Xibilia, M., 2005, "Soft Sensors for Product Quality Monitoring in Debutanizer Distillation Columns," Control Engineering Practice, 13, pp. 411-508.
- [79] Gola, G., Nybo, R., Sui, D., and Roverso, D., "Improving Management and Control of Drilling Operations with Artificial Intelligence," Society of Petroleum Engineers Intelligent Energy, pp. 1-7.
- [80] Diebold, A., 1995, "Overview of metrology requirements based on the 1994 national technology roadmap for semiconductor," in Proc. IEEE/SEMI Adv. Semicond. Manuf. Conf. Workshop, pp. 50-60.
- [81] Chang, Y., and Cheng, F., 2005, "Application Development of Virtual Metrology in Semiconductor Industry," in Proc. 31st Annu. Conf. IEEE Ind. Electron., pp. 124-129.
- [82] Geng, Hwaiyu, 2005, *Semiconductor Manufacturing Handbook*, McGraw-Hill Handbooks, NY, (14).
- [83] Kang, P., Lee, H., Cho, S., Kim, D., Park, J., Park, C., and Doh, S., 2009, "A Virtual Metrology System for Semiconductor Manufacturing," Expert Systems with Applications, 36, pp. 12554-12561.
- [84] Kang, P., Kim, D., Lee, H., Doh, S., and Cho, S., 2011, "Virtual Metrology for Run to Run Control in Semiconductor Manufacturing" Expert Systems with Applications, 38, pp. 2508-2522.
- [85] Lee, S., and Spanos, C., 1995, "Prediction of Wafer State after Plasma Processing Using Real Time Tool Data," IEEE Trans. Semiconductor Manufacturing, 8(3), pp. 252-261.
- [86] Nadi, F., Agogino, A., and Hodges, D., 1991, "Use of Influence Diagrams and Neural Networks in Modeling Semiconductor Manufacturing Processes," IEEE Trans. Semiconductor Manufacturing, 4, pp. 52-58.

- [87] Wold, S., Sjoström, M., and Eriksson, L., 2001, "PLS-Regression: A Basic Tool of Chemometrics," *Chemometrics and Intelligent Laboratory Systems*, 58, pp. 109-130.
- [88] Himmel, C., and May, G., 1993, "Advantages of Plasma Etch Modeling Using Neural Networks Over Statistical Techniques," *IEEE Trans. Semiconductor Manufacturing*, 6(2).
- [89] Rietman, E., and Low, E., 1993, "Use of Neural Networks in Modeling Semiconductor Manufacturing Processes: An Example for Plasma Etch Modeling," *IEEE Trans. Semiconductor Manufacturing*, 6(4), pp. 343-347.
- [90] Cheng, F., Huang, H., and Kao, C., 2007, "Dual Phase Virtual Metrology Scheme," *IEEE Trans. on Semiconductor Manufacturing*, 20(4), pp. 566-571.
- [91] Cheng, F., Chen, Y., Su, Y., and Zeng, D., 2008, "Evaluating Reliance Level of a Virtual Metrology System," *IEEE Trans. on Semiconductor Manufacturing*, 21(1), pp. 92-103.
- [92] Su, Y., Lin, T., Cheng, F., and Wu, W., 2007, "Implementation Considerations of Various Virtual Metrology Algorithms," *IEEE Conference on Automation Science and Engineering*, pp. 277-281.
- [93] Su, Y., Lin, T., Cheng, F., and Wu, W., 2008, "Accuracy and Real Time Considerations for Implementing Various Virtual Metrology Algorithms," *IEEE Trans. on Semiconductor Manufacturing*, 21(3), pp. 426-434.
- [94] Wu, W., Cheng, F., and Kong, F., 2012, "Dynamic Moving Window Scheme for Virtual Metrology Model Refreshing," *IEEE Trans. on Semiconductor Manufacturing*, 25(2), pp. 238-246.
- [95] Wu, W., Cheng, F., Lin, T., Zeng, D., and Chen, J., 2011, "Selection Schemes of Dual Virtual Metrology Outputs for Enhancing Prediction Accuracy," *IEEE Trans. on Automation Science and Engineering*, 8(2), pp. 311-318.
- [96] Hung, M., Lin, T., Cheng, F., and Lin, R., 2007, "A Novel Virtual Metrology Scheme for Predicting CVD Thickness in Semiconductor Manufacturing," *IEEE Transactions on Mechatronics*, Vol. 12, 3, pp. 308-316.
- [97] Hung, M., Chen, C., Lin, Y., Chou, M., and Cheng, F., 2012, "Refinement of Kernel and Functional Mechanisms for Automatic Virtual Metrology System," *IEEE Conference on Advanced Intelligent Mechatronics*, pp. 472-477.
- [98] Hung, M., Huang, H., Yang, H., and Cheng, F., 2010, "Development of an Automatic Virtual Metrology Framework for TFT-LCD Industry," *IEEE Conference on Automation Science and Engineering*, pp. 879-884.
- [99] Ferreira, A., Roussy, A., Kernaflen, C., Gleispach, D., and Hayderer, G., 2011, "Virtual Metrology Models for Predicting Average PECVD Oxide Film Thickness," *IEEE ASMC*.
- [100] Ferreira, A., Roussy, A., and Conde, L., 2009, "Virtual Metrology Models for Predicting Physical Measurement in Semiconductor Manufacturing," *ASMC*, pp. 149-154.

- [101] Besnard, J., Gleispach, D., Gris, H., Ferreira, A., Roussy, A., Kernaflen, C., and Hayderer, G., 2012, "Virtual Metrology Modeling for CVD Film Thickness," *International Journal of Control Science and Engineering*, 2(3), pp. 26-33.
- [102] Olsen, K., and Moyne, J., 2010. "Adaptive Virtual Metrology Applied to a CVD Process," *IEEE ASMC*, pp. 1-6.
- [103] Chou, P., Wu, M., and Chen, K., 2010, "Integrated Support Vector Machine and Genetic Algorithm to Implement Dynamic Wafer Quality Prediction System," *Expert Systems with Applications*, 37, pp. 4413-4424.
- [104] Purwins, H., Barak, B., Jagi, A., Engel, R., Hockele, U., Kyek, A., Cherla, S., Lenz, B., Pfeifer, G., and Weinzierl, K., 2012, "Regression Methods for Virtual Metrology of Layer Thickness in Chemical Vapor Deposition," *IEEE Transactions on Mechatronics*, pp. 1-8.
- [105] Zeng, D. and Spanos, C., 2009, "Virtual Metrology Modeling for Plasma Etch Operations," *IEEE Trans. on Semiconductor Manufacturing*, 22(4), pp. 419-431.
- [106] Lynn, S., Ringwood, J., and MacGearailt, N., 2010, "Gaussian Process regression for Virtual Metrology of Plasma Etch," *ISSC, UCC*, pp. 1-6.
- [107] Lynn, S., MacGearailt, N., and Ringwood, J., 2012, "Real Time Virtual Metrology and Control for Plasma Etch," *Journal of Process Control*, 22, pp. 666-676.
- [108] Lu, B., Stuber, J., and Edgar, T., 2014, "Integrated Online Virtual Metrology and Fault Detection in Plasma Etch Tools," *I&EC Research*.
- [109] Gill, B., 2011, "Development of Virtual Metrology in Semiconductor Manufacturing," *The University of Texas at Austin PhD Dissertation*.
- [110] Gill, B., Edgar, T., Stuber, J., 2010, "A Novel Approach to Virtual Metrology Using Kalman Filtering," *Future Fab International*, 35. .
- [111] Banks, H., and Hu, S., 2011, "Propagation of Uncertainty in Dynamical Systems," *NCSU technical paper, Center for Research and Scientific Computation*, pp. 1-6.
- [112] Ratliff, L., 2010, *Conservation Based Uncertainty Propagation in Dynamic Systems*, UNLV Theses.
- [113] Giza, D., Singla, P., and Jah, M., 2009, "An Approach for Nonlinear Uncertainty Propagation: Application to Orbital Mechanics," *Proceedings of the American Institute of Aeronautics and Astronautics*, pp. 1-19.
- [114] Mari, L., and Petri, D., 2006, "Propagating Uncertainty through Discrete Time Dynamic Systems," *IEEE Workshop on Advanced Methods for Uncertainty Estimation in Measurement*, pp. 23-26.

- [115] Worden, K., Manson, G., Lord, T., and Friswell, M., 2005, "Some Observations on Uncertainty Propagation through a Simple Nonlinear System," *Journal of Sound and Vibration*, 288, pp. 601-621.
- [116] Lee, S., and Chen, W., 2009, "A Comparative Study of Uncertainty Propagation Methods for Black Box Type Problems," *Structured Multidisciplinary Optimization*, 37, pp. 239-253.
- [117] Terejanu, G., Singla, P., Singh, T., and Scott, P., 2008, "Uncertainty Propagation for Nonlinear Dynamic Systems Using Gaussian Mixture Models," *Journal of Guidance, Control, and Dynamics*, Vol. 31, 6, pp. 1623-1633.
- [118] Terejanu, G., Singla, P., Singh, T., and Scott, P., 2011, "Approximate Propagation of both Epistemic and Aleatory Uncertainty through Dynamic Systems," *FUSION*, pp. 1-8.
- [119] Shinozuka, M., 1972, "Monte Carlo Solution of Structural Dynamics," *Computers and Structures*, Vol. 2, pp. 855-874.
- [120] Hosder, S., Walters, R., and Perez, R., 2006, "A Non-Intrusive Polynomial Chaos Method for Uncertainty Propagation in CFD Simulations," *AIAA Aerospace Sciences Meeting*, pp. 1-19.
- [121] Knio, O., and LeMaitre, O., 2006, "Uncertainty Propagation in CFD using polynomial Chaos Decomposition," *Fluid Dynamics Research*, 38, pp.616-640.
- [122] Patra, A., Bursik, M., Dehn, J., and Jones, M., 2012, "A DDDAS Framework for Volcanic Ash Propagation and Hazard Analysis," *International Conference on Computational Science*, 9, pp. 1090-1099.
- [123] Terejanu, G., Cheng, Y., Singh, T., and Scott, P., 2011, "Comparison of SCIPUFF Plume Prediction with Particle Filter Assimilated Prediction for Dipole Pride 26 Data," graduate research conference University of Buffalo, NY, pp. 1-5.
- [124] Salih, Y., Malik, A., 2011, "Comparison of Stochastic Filtering Methods for 3D Tracking," *Pattern Recognition*, 44, pp. 2711-2737.
- [125] Rosenbluth, M., 1957, "Fokker Plank Equation for an Inverse Square Force," *The Physical Review*, Vol, 107, 1, pp. 1-6.
- [126] Risken, H., 1989, *The Fokker Plank Equation*, Springer-Verlag, New York.
- [127] Oksendal, B., 1998, *Stochastic Differential Equations*, Springer, Berlin.
- [128] Terejanu, G., 2009, "Tutorial on Monte Carlo Techniques," University of Buffalo, NY, Technical Paper.
- [129] Godsill, S., and Djuric, P., 2002, "Monte Carlo methods for statistical signal processing," *IEEE Trans. Signal Processing (Special Issue)*, Vol. 50, 2, pp. 173-499.
- [130] Doucet, A., de Freitas, N., and Gordon, N., 2001, *Sequential Monte-Carlo Methods in Practice*, Springer-Verlag, pp. 6-14.

- [131] Haykin, S., and Freitas, N., 2004, "From Kalman filters to particle filters," Proc. IEEE (Special Issue on Sequential State Estimation), Vol. 92, 3, pp. 399–574.
- [132] Candy, J., 2007, "Bootstrap Particle Filtering," IEEE Signal Processing magazine, July, pp. 73-85.
- [133] Iyengar, R. N. and Dash, P. K., 1978, "Study of the Random Vibration of Nonlinear Systems by the Gaussian Closure Technique," Journal of Applied Mechanics, Vol. 45, pp. 393-399.
- [134] Roberts, J. B. and Spanos, P. D., 1990, *Random Vibration and Statistical Linearization*, Wiley, pp. 122-176.
- [135] Lefebvre, T., Bruyninckx, H., and Schutter, J. D., 2004, "Kalman Filters of Non-Linear Systems: A Comparison of Performance," International Journal of Control, Vol. 77, 7, pp. 639-653.
- [136] Lefebvre, T., Bruyninckx, H., and Schutter, J. D., 2002, "Comment on A New Method for the Nonlinear Transformations of Means and Covariances in Filters and Estimators," IEEE Transactions on Automatic Control, Vol. 47, 8.
- [137] Enszer, J., Lin, Y., Ferson, S., Corliss, G., and Stadtherr, M., 2008, "Propagating Uncertainties in Modeling Nonlinear Dynamic Systems," University of Notre Dame REC Technical Conference.
- [138] Chen, Y., and Gupta, M., 2010, "EM Demystified: An Expectation Maximization Tutorial," UWEE Technical Report, University of Washington Department of EE.
- [139] Montgomery, D. C., 2001, *Introduction to statistical quality control*, New York, John Wiley (4).
- [140] Hu, F., and Zidek, J., 2002, "The Weighted Likelihood," The Canadian Journal of Statistics, (30)3, pp. 347-371.
- [141] Hu, F., Rosenberger, W., and Zidek, J., 2000, "Relevance Weighted Likelihood for Dependent Data," Journal of Metrika, (51)3, pp. 223-243.
- [142] Hu, F., and Zidek, J., 2004, "Forecasting NBA Basketball Playoff Outcomes Using the Weighted Likelihood," Institute of Mathematical Statistics, Lecture Notes-Monograph Series, (45), pp. 385-395.
- [143] Liu, J., Djurdjanovic, D., Marko, K., and Ni, J., 2009, "Growing Structure Multiple Model Systems for Anomaly Detection and Fault Diagnosis," ASME Journal of Dynamic Systems, Measurement, and Control, 131, pp. 1-13.
- [144] Liu, J., Sun, P., Djurdjanovic, D., Marko, K., and Ni, J., 2006, "Growing Structure Multiple Model System Based Anomaly Detection for Crankshaft Monitoring," Advances in Neural Networks, Lecture Notes in Computer Sciences ISNN, 3973, pp. 396-405.
- [145] Liu, J., and Djurdjanovic, D., 2008, "Topology Preservation in Cooperative Learning in Identification of Multiple Model Systems," IEEE Transactions on Neural Networks, 19(12), pp. 2065-2072.

- [146] Cholette, M., and Djurdjanovic, D., 2012, "Precedent Free Fault Isolation in a Diesel Engine Exhaust Gas Recirculation System," *ASME Journal of Dynamic Systems and Control*, 134 (3), pp. 1-11.
- [147] Cholette, M., Liu, J., Djurdjanovic, D., and Marko, K., 2012, "Monitoring of Complex Systems of Interacting Dynamic Systems," *Applied Intelligence*, 37(1), pp. 60-79.
- [148] Cho, K. B., and Wang, B. H., 1996, "Radial Basis Function Based Adaptive Fuzzy Systems and Their Applications to System Identification and Prediction," *Fuzzy Sets Syst.*, 83(3), pp. 325-339.
- [149] Leung, H., Lo, T., and Wang, S., 2001, "Prediction of Noisy Chaotic Time Series Using an Optimal Radial Basis Function Neural Network," *IEEE Trans. Neural Netw.*, 12(5), pp. 1163-1172.
- [150] Principe, J. C., Wang, L., and Motter, M. A., 1998, "Local Dynamic Modeling With Self-Organizing Maps and Applications to Nonlinear System Identification and Control," *Proc. IEEE*, 86(11), pp. 2240-2258.
- [151] Patton, R., Frank, P. M., and Clark, R. N., 1996, *Issues Of Fault Diagnosis For Dynamic Systems*, Springer-Verlag, New York.
- [152] Takagi, T., and Sugeno, M., 1985, "Fuzzy Identification of Systems and Its Applications to Modeling and Control," *IEEE Trans. Syst. Man Cybern.*, SMC-15(1), pp. 116-132.
- [153] Tzafestas, S., and Zikidis, K., 2001, "NeuroFAST: On-Line Neuro-Fuzzy Art- Based Structure and Parameter Learning TSK Model," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, 31(5), pp. 797-802.
- [154] Johansen, T., and Foss, B., 1993, "Constructing NARMAX Models Using ARMAX Models," *Int. J. Control*, 58(5), pp. 1125-53.
- [155] Johansen, T., and Foss, B., 1995, "Identification of Nonlinear System Structure and Parameters Using Regime Decomposition," *Automatica*, 31(2), pp. 321-6.
- [156] Kohonen, T., 1995, *Self-Organizing Maps*, Springer-Verlag, Berlin, NY.
- [157] Barreto, G., and Araujo, A., 2004, "Identification and Control of Dynamical Systems Using the Self-Organizing Map," *IEEE Trans. Neural Netw.*, 15(5), pp. 1244-59.
- [158] Ge, M., Chin, M.-S., and Wang, Q.-G., 1999, "An Extended Self-Organizing Map for Nonlinear System Identification," *Proceedings of the 1999 Conference on Decision and Control*, IEEE, Phoenix, AZ, Dec. 7-10, 1, pp. 1065-70.
- [159] Fritzke, B., 1995, "A Growing Neural Gas Network Learns Topologies," *Advances in Neural Information Processing Systems*, MIT, Cambridge, MA, 7, pp. 625-632.
- [160] Fritzke, B., 1994, "Growing Cell Structures—A Self-Organizing Network for Unsupervised and Supervised Learning," *Neural Networks*, 7(9), pp. 1441-1460.

- [161] Alahakoon, D., Halgamuge, S. K., and Srinivasan, B., 2000, "Dynamic Self- Organizing Maps With Controlled Growth for Knowledge Discovery," *IEEE Trans. Neural Netw.*, 11(3), pp. 601–614.
- [162] Fritzke, B., 1995, "Incremental Learning of Local Linear Mappings," *Proceedings of the International Conference on Artificial Neural Networks, ICANN '95*, Oct. 9–13, EC2 & Cie, Paris, France, 1, pp. 217–22.
- [163] Martinetz, T. M., Berkovich, S. G., and Schulten, K. J., 1993, "'Neural-Gas' Network for Vector Quantization and Its Application to Time-Series Prediction," *IEEE Trans. Neural Netw.*, 4(4), pp. 558–569.
- [164] Sedgewick, R., 1995, *Algorithms in C++, Part 5: Graph Algorithms*, Addison-Wesley, London.
- [165] Ikonen, E., and Najim, K., 2002, *Advanced Process Identification and Control*, Marcel Dekker, New York.
- [166] Martinetz, T., and Schulten, K., 1991, "A 'Neural-Gas' Network Learns Topologies," *Artificial Neural Networks, Proceedings of the 1991 International Conference, ICANN-91*, Jun. 24–28, North-Holland, Espoo, Finland, pp. 397–402.
- [167] Martinetz, T., and Schulten, K., 1994, "Topology Representing Networks," *Neural Networks*, 7(3), pp. 507–522.
- [168] Bleakie, A., Djurdjanovic, D., 2013, "Feature Extraction, Condition Monitoring, and Fault Modeling in Semiconductor Manufacturing Systems," *Computers in Industry*, 64(3), pp. 203–213.
- [169] Zhou, D., Gang, L., and Qin, S.J., 2010, "Total Projection to Latent Structures for Process Monitoring," *American Institute for Chemical Engineering Journal*, 56(1), pp. 168-178.

Vita

Alexander Q. Bleakie was born in Houston, Texas, USA. After graduating in 2003 from Clear Lake High School in Houston, he attended Stephen F. Austin State University in Nacogdoches, Texas where he studied Physics and Engineering. In 2005, he decided to attend Texas A&M University in College Station, Texas to study Mechanical Engineering and received a Bachelor of Science degree in 2008 with Summa Cum Laude Honors. In September 2008, he enrolled in The University of Texas at Austin Ph.D. Program in Mechanical Engineering with emphasis on Dynamic Systems and Control. His work at the University of Texas for the last six years has been focused on predictive health modeling and control of manufacturing systems and machines.

Permanent address: 1218 W. Flamingo Dr.

Seabrook, TX 77586

E-mail address: bleakie_8@hotmail.com

This thesis was typed by the author.