I, Lei Yang, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Mechanical Engineering.

It is entitled:
Methodology of Prognostics Evaluation for Multiprocess Manufacturing Systems

Student's name: Lei Yang

This work and its defense approved by:

Committee chair: Jay Lee, PhD
Committee member: Dragan Djurdjanovic, PhD
Committee member: Gregory A. Cherry, PhD
Committee member: Hongdao Huang
Committee member: Mark Schulz
Methodology of Prognostics Evaluation for Multiprocess Manufacturing Systems

A dissertation submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

in the School of Dynamic Systems
of the College of Engineering and Applied Science
by

Lei Yang

University of Cincinnati
January 2011

Committee Chair: Jay Lee, Ph.D

© 2011 Lei Yang. All Rights Reserved.
ABSTRACT

Methodology of Prognostics Evaluation for Multiprocess Manufacturing Systems

by

Lei Yang

Chair: Dr. Jay Lee

This dissertation presents a unified methodology of prognostics evaluation for multiple product and process manufacturing system. Under the proposed framework, several prognostics tasks, including failure predictability evaluation, component Remaining Useful Life (RUL) estimation, and root cause analysis, are systematically addressed. While the methods introduced in this work are applicable for general manufacturing system with multiple operational conditions, it places a particular emphasis on the semiconductor environment and addresses some unique challenges in wafer fabrication equipment prognostics. All methods developed in this work are tested with use cases from the fabrication facilities in GLOBALFOUNDRIES.

A Partial Least Square (PLS) Regression-based method is developed for failure predictability evaluation, RUL estimation and fault diagnosis. A group of so-called Generalized Remaining Useful Life (GRUL) curves, which are defined to reflect various degradation patterns, are combined as multivariate output of PLS regression model. The PLS model selects the variations from the inputs that have the best correlation with the output, based on which an optimal RUL target curve is obtained to
represent the degradation pattern of the equipment. The predictability of the failure mode can be evaluated by comparing the RUL estimation provided by the model and the lead time requirement of actual maintenance practice. Furthermore, a variable blocking contribution strategy is proposed to enable a user to hierarchically drill down to the input variables and identify the root cause of the failure.

As today’s equipment is constantly operating under different regimes for various products’ specification while its condition keeps degrading, it is critical to monitor the condition consistently such that the schedule of operation and maintenance can be optimized. The PLS modeling strategy is further extended to address this issue with separate models built for individual operational regimes. Assuming that data from each process spread widely over the whole degradation process, individual PLS models are built to capture the impact of different processes on the equipment. Given a tool is running on a particular process, the RUL for the current process as well as the equivalent RUL for other processes can be obtained. With this information, a recommendation can be made on future production such that equipment availability is maximized and maintenance can be scheduled in a timely manner.
To my parents,

whose love and support help me find my success
ACKNOWLEDGEMENTS

First, I want to thank my advisor Prof. Jay Lee, who has not only taught me to be a good researcher but more importantly a good person. Dr. Lee also puts me in an environment where I could find my success, and generously granted me the flexibility that allows me to pursue my educational goals and to advance my professional career simultaneously, which is instrumental to the completion of this work. I am grateful to all my committee members for their in-depth feedback and guidance. Special thanks goes to Dragan, who led me into the area of CBM when I started my graduate study in US; and to Greg, who has been my mentor since day one when I joined AMD.

I am also very fortunate to be able to work with many fellow graduate students, visiting scholars and friends who have inspired me through the years. In particular, Linxia Liao, Tianyi Wang, Patrick Brown, Edzel Lapira, Yan Chen, Masoud Gaffari, Wenyu Zhao, David Siegel, Ashwin Balasubramanya, Mohamed AbuAli, Shijin Wang, Bob Yu and Li He all lent their hands in both good days and in bad.

I must also thank many past and present colleagues for their assistance. The entire Process Control Application team lent their support for this work. In particular, Elfido Coss, Robert Chong, Niels Rackwitz, Matthew Purdy, Matthew Ryskoski, Thomas Sonderman, Christopher Bode and Gregory Cherry offered me guidance and advice on how to complete this research as well as continue my career at GLOBAL-FOUNDRIES.

Finally, and most importantly, I want to thank my parents, for their constant love and tolerance that ensure me to maintain enough sanity to complete this journey.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>DEDICATION</td>
<td>v</td>
</tr>
<tr>
<td>ACKNOWLEDGEMENTS</td>
<td>vi</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>xiii</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>xiv</td>
</tr>
<tr>
<td><strong>CHAPTER</strong></td>
<td></td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Condition-based Monitoring for Multiprocess Manufacturing Equipments</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Multiprocess Manufacturing in Semiconductor Fabrication</td>
<td></td>
</tr>
<tr>
<td>1.2.1 Manufacturing Process</td>
<td>4</td>
</tr>
<tr>
<td>1.2.2 Manufacturing Equipment</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Research Challenges</td>
<td>9</td>
</tr>
<tr>
<td>1.3.1 Predictability quantification</td>
<td>9</td>
</tr>
<tr>
<td>1.3.2 Data quantity</td>
<td>11</td>
</tr>
<tr>
<td>1.3.3 Process variability</td>
<td>12</td>
</tr>
<tr>
<td>1.4 Research Objectives and Broader Impacts</td>
<td>12</td>
</tr>
<tr>
<td>1.5 Organization of the Dissertation</td>
<td>14</td>
</tr>
<tr>
<td>II. State-of-the-art of CBM Techniques in Semiconductor Manu-</td>
<td></td>
</tr>
<tr>
<td>facturing</td>
<td>16</td>
</tr>
<tr>
<td>2.1 Data Pre-processing and Feature Extraction</td>
<td>16</td>
</tr>
<tr>
<td>2.2 Performance Assessment</td>
<td>21</td>
</tr>
<tr>
<td>2.3 Fault Diagnostics</td>
<td>27</td>
</tr>
<tr>
<td>2.4 Performance Prediction</td>
<td>29</td>
</tr>
</tbody>
</table>
5.3.3 Process Similarity .......................... 117

BIBLIOGRAPHY ........................................ 119

APPENDICES ............................................. 127
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Example of Recipe Step Definition from a Chamber</td>
<td>6</td>
</tr>
<tr>
<td>1.2</td>
<td>Example of Voltage Readings from a Chamber</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>A Typical Mainframe Configuration of Semiconductor Tool</td>
<td>7</td>
</tr>
<tr>
<td>1.4</td>
<td>Example of Recipe Changes in a Production Chamber</td>
<td>9</td>
</tr>
<tr>
<td>1.5</td>
<td>Predictability and Maintenance Strategy</td>
<td>10</td>
</tr>
<tr>
<td>1.6</td>
<td>Development Process of a Prognostic Solution</td>
<td>13</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of target GRUL group defined by truncated quadratic function</td>
<td>40</td>
</tr>
<tr>
<td>3.2</td>
<td>Waveforms of Input Variables</td>
<td>47</td>
</tr>
<tr>
<td>3.3</td>
<td>Input Variables for PLS Simulation</td>
<td>48</td>
</tr>
<tr>
<td>3.4</td>
<td>Target Group and Optimal RUL</td>
<td>49</td>
</tr>
<tr>
<td>3.5</td>
<td>Variable Individual Contribution for PLS Regression</td>
<td>50</td>
</tr>
<tr>
<td>3.6</td>
<td>Variable Block Contribution for PLS Regression</td>
<td>51</td>
</tr>
<tr>
<td>3.7</td>
<td>Loading Scores of PLS Model</td>
<td>53</td>
</tr>
<tr>
<td>3.8</td>
<td>RUL Estimation of Simulated Data</td>
<td>54</td>
</tr>
<tr>
<td>3.9</td>
<td>Simulation Input Function Types and Training Samples (In red)</td>
<td>55</td>
</tr>
<tr>
<td>3.10</td>
<td>PLS Block Contribution with Variable Selection Strategy</td>
<td>55</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.11</td>
<td>PLS Loading Scores with Variable Selection Strategy</td>
<td>56</td>
</tr>
<tr>
<td>3.12</td>
<td>RUL Estimation with Variable Selection Strategy</td>
<td>57</td>
</tr>
<tr>
<td>3.13</td>
<td>PLS Target Group and Optimal RUL for Random Noise Input</td>
<td>58</td>
</tr>
<tr>
<td>3.14</td>
<td>PLS Loading Scores for Random Noise Input</td>
<td>59</td>
</tr>
<tr>
<td>3.15</td>
<td>Variable Block Contribution with Noise Target Presence</td>
<td>59</td>
</tr>
<tr>
<td>3.16</td>
<td>Example of Recipe Switch on the Tool</td>
<td>66</td>
</tr>
<tr>
<td>3.17</td>
<td>Example of Target RUL Group for Individual Recipes</td>
<td>68</td>
</tr>
<tr>
<td>3.18</td>
<td>Example of Optimal Equivalent RUL Individual Recipes</td>
<td>68</td>
</tr>
<tr>
<td>3.19</td>
<td>Simulated Example of Semiconductor Recipe Switch in a Chamber</td>
<td>70</td>
</tr>
<tr>
<td>3.20</td>
<td>Simulated Example of Process Switch</td>
<td>74</td>
</tr>
<tr>
<td>3.21</td>
<td>Relationship Between Process Switch and Model Trainability</td>
<td>74</td>
</tr>
<tr>
<td>3.22</td>
<td>Input Function Types for Two Different Training Sets</td>
<td>76</td>
</tr>
<tr>
<td>3.23</td>
<td>Variable Blocking Contribution of Combined Training Data Sets</td>
<td>77</td>
</tr>
<tr>
<td>4.1</td>
<td>PLS Loading Scores without Variable Selection</td>
<td>81</td>
</tr>
<tr>
<td>4.2</td>
<td>Variable Blocking Contribution without Variable Selection</td>
<td>82</td>
</tr>
<tr>
<td>4.3</td>
<td>PLS Loading Score with Selected High Contribution Variable</td>
<td>83</td>
</tr>
<tr>
<td>4.4</td>
<td>Variable Blocking Contribution with Selected High Contribution Variable</td>
<td>83</td>
</tr>
<tr>
<td>4.5</td>
<td>Mean Values from Sensor 19 for Model Building</td>
<td>84</td>
</tr>
<tr>
<td>4.6</td>
<td>Mean Values from Sensor 2 for Model Building</td>
<td>85</td>
</tr>
<tr>
<td>4.7</td>
<td>RUL Prediction by Selected Variables</td>
<td>86</td>
</tr>
<tr>
<td>4.8</td>
<td>Validation of RUL Prediction</td>
<td>87</td>
</tr>
<tr>
<td>4.9</td>
<td>Optimal RUL Target of Cryo Pump</td>
<td>88</td>
</tr>
</tbody>
</table>
4.10 Loading Score of a Detectable Failure Mode 89
4.11 Contribution of the Pressure Variables 90
4.12 Summary Statistic Variables of the Pressure 90
4.13 Optimal RUL Target of Spin Motor 92
4.14 Loading Score of Unpredictable Failure Mode 92
4.15 Variable Blocking Contribution 93
4.16 Feature Variables from Sensor 5 for Model Building 93
4.17 Recipe Switch in Two Etch Chambers 95
4.18 Example of Recipe Step and Sensor Readings 95
4.19 Sensor Blocking Contribution by Different Training Data Set of Recipe 97
4.20 Mean Values of Recipe 5, Step 6 from Sensors with Contradictive (11 & 18) and Consistent (2 & 29) Blocking Contribution 97
4.21 Optimal Target RUL for the Etch Recipes 99
4.22 Loading Scores on Selected Principal Component of Different Recipes 100
4.23 RUL Estimation of Different Recipes 101
4.24 RUL Validation for Recipes 3 and 5 102
4.25 Schematic of a Polisher Configuration 103
4.26 Example of Reflexion Normalized Scan Head Measurement Data 104
4.27 Production Recipes Before a Pad Change 105
4.28 Optimal Target RUL for the Polish Recipes 106
4.29 Loading Scores on Selected Principal Components of Different Recipes 107
4.30 Variable Block Contribution of Different Recipes 109
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Statistics for Recipe Switch Simulation</td>
<td>70</td>
</tr>
<tr>
<td>4.1</td>
<td>Process Summary for High Frequency Recipes in Fig. 4.17</td>
<td>96</td>
</tr>
<tr>
<td>4.2</td>
<td>Process Switch Index for Recipes in Fig. 4.27</td>
<td>106</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------</td>
<td></td>
</tr>
<tr>
<td>AMHS</td>
<td>Automatic Material Handling System</td>
<td></td>
</tr>
<tr>
<td>CBM</td>
<td>Condition Based Monitoring</td>
<td></td>
</tr>
<tr>
<td>CMP</td>
<td>Chemical Mechanical Planarization</td>
<td></td>
</tr>
<tr>
<td>DPLS</td>
<td>Discriminant Partial Least Square</td>
<td></td>
</tr>
<tr>
<td>ESC</td>
<td>Electrostatic Chuck</td>
<td></td>
</tr>
<tr>
<td>FDA</td>
<td>Fisher Discriminant Analysis</td>
<td></td>
</tr>
<tr>
<td>GMM</td>
<td>Gaussian Mixture Model</td>
<td></td>
</tr>
<tr>
<td>GRUL</td>
<td>Generalized Remaining Useful Life</td>
<td></td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Model</td>
<td></td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Component Analysis</td>
<td></td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbors</td>
<td></td>
</tr>
<tr>
<td>PLS</td>
<td>Partial Least Square</td>
<td></td>
</tr>
<tr>
<td>PSI</td>
<td>Process Switch Index</td>
<td></td>
</tr>
<tr>
<td>PVD</td>
<td>Physical Vapor Deposition</td>
<td></td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
<td></td>
</tr>
<tr>
<td>RTA</td>
<td>Rapid Thermal Annealing</td>
<td></td>
</tr>
<tr>
<td>RUL</td>
<td>Remaining Useful Life</td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machines</td>
<td></td>
</tr>
</tbody>
</table>
CHAPTER I

Introduction

1.1 Condition-based Monitoring for Multiprocess Manufacturing Equipments

Condition Based Monitoring (CBM) can be seen as an integral process of seamless transformation of sensory data that are collected during operations into information about equipment health, and further into decisions that need to be made with respect to that equipment. In general, a typical CBM solution includes the following functionalities:

- **Feature Extraction**: pre-process raw data to eliminate possible contamination and extract indicative features for equipment condition monitoring.

- **Performance Assessment**: quantitatively evaluate equipment health status based on extracted features, provide warnings as any imminent fault is detected.

- **Performance Prediction**: forecast the equipment performance into the future and provide estimation of component RUL for specific failure mode.

- **Fault Diagnostics**: identify root cause when fault is detected and suggest appropriate correction items based on engineering experience.
The concept of CBM received significant attention, especially in the case of sophisticated, expensive and safety critical systems, such as manufacturing equipment [46, 70], automotive [37, 22, 67] and aircraft engines [29, 6].

In the manufacturing industry, a workpiece usually goes through multiple stages of manufacturing process before the product is finally made. On each individual stage, a particular type of equipment is deemed to perform a distinct operation according to the specification. As the product category varies, the requirements for different operation cycles can change dramatically. In order to reduce production cost while maintaining the robustness of manufacturing capability, each type of equipment becomes more sophisticated to satisfy various process requirements. This inevitably causes tremendous complexity of the manufacturing system, which often results in unexpected downtime and troubleshooting difficulties. Thus, equipment condition monitoring and failure prevention are becoming important factors for manufacturing productivity and product quality.

In this dissertation, the definition of a process is given as

**Definition I.1** (Process). A time period in which a certain type of operation is performed on a workpiece by the equipment according to the specifications.

And thus a multiprocess equipment can be defined as

**Definition I.2** (Multiprocess Equipment). Manufacturing equipment running under different conditions to perform the operation with various specifications.

For example, a drilling process, during which a hole is made on the workpiece, can be performed by a machining tool based on the specifications such as hole diameter, depth, surface smoothness etc. The operating condition of the tool can be described by parameters like feed rate, motor rotation speed, spindle load, etc.

Due to the increasing product diversity, today’s manufacturing equipments are usually required to constantly operate under distinct conditions to deliver different
specifications for various products. As a significant amount of data is collected during the production nowadays, it is critical to convert the data into useful information to assist production planning and equipment monitoring. The work presented in this dissertation aims to develop a data-driven method that addresses the challenges in prognostics evaluation for the multiprocess manufacturing equipment. While the method introduced here is applicable for general multiprocess equipment, it places a particular emphasis to address some unique challenges in semiconductor manufacturing.

In the remainder of this chapter, Section 1.2 briefly introduces the semiconductor fabrication process and equipment. Section 1.3 identifies the research challenges and requirements of a CBM solution in a general multiprocess manufacturing environment. Section 1.4 highlights the objectives and broader impacts of the conducted research. Finally, Section 1.5 outlines the content that will be elaborated in the following chapters of the dissertation.

1.2 Multiprocess Manufacturing in Semiconductor Fabrication

Semiconductor fabrication has a typical multiprocess environment, in which numerous manufacturing processes are designed to fabricate a variety of products. In addition, it is characterized by high equipment cost and rich data collection. The price for an ordinary 300 millimeter equipment ranges from hundreds of thousands to several million dollars each. For a regular fabrication facility that contains hundreds of tool sets, the cost can easily reach as much as several billion dollars. A standard tool set usually provides hundreds of sensors to measure process condition for the purpose of process and equipment control. Thus, CBM has a broad horizon in reducing manufacturing cost and ensuring product quality. This section first intro-
duces the semiconductor manufacturing process, followed by the review of a typical semiconductor equipment, detailing component characteristics and process flow.

1.2.1 Manufacturing Process

Semiconductor process refers to a number of chemical-mechanical operations, such as lithography, deposition, etch, diffusion, polishing, etc., that the workpiece goes through before the product is made. The silicon workpiece on which the process is performed is called a wafer. During a particular process, a single layer of the wafer is created or altered to form a specific piece of the circuitry. In general, the semiconductor process starts from lithography, in which the masking pattern of the circuitry is formed on the wafer. Successive processes then act upon the wafer uniformly to alter the portion that is exposed by the mask. When the process is complete with the current pattern, the mask is removed from the wafer surface and it is ready for the next process cycle. In this manner the electrical structures of the device are built.

An individual semiconductor process is controlled by a recipe, in which a sequence of steps are defined to prepare process conditions and perform the operation on the wafer surface. Based on the step characters, a recipe can be largely divided into three stages. Steps during the first stage are responsible for creating the required operation conditions; followed by steps that perform the actual operation on the wafer surface; the recipe is then concluded by steps that resume the conditions to ambient environment. For example, in a typical deposition process, preparation steps include heat, pump up, chuck, and pressure stabilizing steps. Then several deposition steps, depending on the specifications, are followed. Then, after steps of pump down and de-chuck, the wafer can be evacuated from the processing area. For different product types, the number of recipes that need to be performed on a wafer varies dramatically, which can take several weeks to months to complete.

Usually the steps in the second stage are considered as critical or major steps and
they take up the most amount of time of a recipe. During each step, a unique operation condition, specified by sensor set points such as temperature, pressure, liquid flow rate, etc., are maintained by the equipment. Theoretically, the major recipe step should continue until it reaches the target specification, such as the film thickness of a deposition process, or the amount of material that needs to be removed in an etch process. However, because the in-situ measurement, also known as metrology, is hard to achieve during the process, there is no real-time feedback and the process time of current operation is given by the controller based on the measurements on previous wafers before the process actually starts. For example, with the actual film thickness and deposition time of previous wafer, the controller estimates the average deposition rate of the equipment, and according to the target thickness of the current wafer, it gives a deposition time. This control scheme is termed as run-to-run control, whose goal is to optimize various inputs such that the variation of target specification is minimized.

With different process specification, the variance that incurs in a recipe comes in two forms. First, the steps’ definition as well as their durations can vary. Fig. 1.1 shows an example of three different recipes from a Physical Vapor Deposition (PVD) process. We notice that the total processing time varies from one to another, and the processing time for each step of the recipe is also not the same. For example, step 5 in recipe 3 takes much longer than that in recipe 1 and 2. Additionally, we can see that recipe 2 has two steps (i.e. 8 and 9) that are not defined in recipe 1 and 3, which indicates different operation types are required for recipe 2.

The second form of recipe variation is observed as different sensor readings due to distinct process conditions. Correspondingly, Fig. 1.2 shows the normalized voltage signals of the recipes in Fig.1.1. It can be seen that although similarities are found in some local patterns between recipe 1 and 3, their relative scales are different. For example, the voltage level of step 5 in recipe 3 is higher than that in recipe 1.
Furthermore, step 5 in recipe 2 shows a spiking trend instead of maintaining at a steady level, which implies different process conditions.
1.2.2 Manufacturing Equipment

Semiconductor equipment is commonly referred to as *tool*, and the isolated processing area in which the recipe takes place is called *chamber*. The tools are docked in the clean room of a fabrication facility, also known as *Fab*, according to their operation types. An Automatic Material Handling System (AMHS), which consists of wafer transfer vehicles and handling robots, is built to connect different tool sets and transfer wafers in between. Together they formulate a highly automated manufacturing environment.

As the processing technology advances continuously, the minimum feature dimension keeps decreasing, hence enabling more and more micro-transistors being built on the wafer surface. Consequently the complexity of semiconductor tools is increasing in order to satisfy the precision requirement as well as diversifying product specification. Figure 1.3 shows a typical configuration of a semiconductor tool, which usually consists of the following systems:

1. Main processing chambers: perform the major deposition process on the wafer.

2. Assisting chambers: perform cleaning, de-gas, cooling and buffering for the

Figure 1.3: A Typical Mainframe Configuration of Semiconductor Tool
wafer before and after the major process.

3. Wafer handling system: transfers wafer from one place to another as it is being processed by the tool.

4. Pumping system: includes cryo, turbo, and rotary vane pumps to provide vacuum condition and gas flow for the chambers during the process, as well as to evacuate exhaust from the chamber after the process.

5. Power system: provides power supply for the subsystems on the tool.

Wafer usually arrives the tool in batch, which also termed as lot and contains no more than 25 wafers that require the same type of process. A typical wafer process starts when a wafer arrives at the load port of a tool, it first goes through those preprocessing chambers for cleaning procedures, and then is transferred to the main chamber for the major processing. Then it will go through the cool down chamber before it leaves the tool for the next process. Within each chamber, the process is controlled by the corresponding recipe.

With mixed groups of products being processed on the same piece of equipment, the recipes change frequently on the tool. Fig. 1.4 shows a continuous production recipe index chart during a time frame of over 500 wafer processes from a PVD chamber. We can see that six different recipes are running alternatively on a lot basis, among which recipe 3 and 4 have higher frequency than others and recipe 1 only appears in one lot. With different requirements for each recipe, one can expect that the operation condition in the chamber is constantly changing. In addition to single-wafer processing, there are other operations, such as Rapid Thermal Annealing (RTA) process, that are performed in batch mode, in which several wafers are processed simultaneously in the chamber.
1.3 Research Challenges

The goal of prognostics for multiprocess manufacturing equipment is two-fold. First, one needs to evaluate the predictability of a particular failure mode, which is determined by the existence of any indicator for component condition in the collected data. Secondly, for a predictable failure mode, information regarding how the component is able to properly operate in the future needs to be provided. Finally, the effect from process variation needs to be addressed such that equipment degradation can be evaluated consistently under different operation conditions. The challenges can be summarized in the following perspectives.

1.3.1 Predictability quantification

The complex tool configuration leads to a variety of failure modes for different components, which prompts the need to quantitatively evaluate the predictability of each one of them based on the collected data and adopt an optimal maintenance
strategy. The relationship between failure predictability, data characteristics, and maintenance strategy is depicted in Fig. 1.5, with predictability changes from high to low going down each row.

In general, the predictability of a particular failure mode depends on how specific one can, based on available data, describe the degradation process. The more specific one can distinguish the degradation status before failure, the more predictable the failure is. And the degradation is indicated by the variation patterns in data, which also determines the type of suitable model that gives the health indicator of the component.

![Figure 1.5: Predictability and Maintenance Strategy](image)

For instance, if there are any trends from the input variables that coincide with the degradation, one might use them as indicators for equipment health status and build a regression model to either predict its future value and decide equipment condition based on some failure criteria, or use that as dependent variable and correlate it with some continuous health index. One example is the RUL, which indicates how long the equipment is good to run before an imminent failure happens. On the other hand, if the input data only demonstrates cluster patterns rather than trends over the course of degradation, a classification model is more suitable to process the data. In this case the health indication becomes discrete instead of continuous, which is less specific than a continuous indicator and thus yields lower predictability. One example is vehicle engine oil life, for which discrete percentages are given. Similarly, if the data only shows some abnormal pattern before failure, such as sudden shifts, and one is...
not able to tell the status change at early age of the degradation, the failure mode would be only detectable. Example of this type of failure pattern are usually found for electrical components, which fail rapidly once any abnormal behavior is observed. Finally, the worst case is that data does not have any meaningful variation at all to indicate component health and the failure is thus totally unpredictable.

It is important to point out that, since the predictability is evaluated based on the variation with data collected during the process, the conclusion is relative to the data and is not absolute for the failure mode, so to speak. If no good patterns were found and we consider the failure unpredictable, one should use engineering analysis to evaluate if other sensors are available from the tool which measure some other related properties that might be helpful to indicate component condition.

1.3.2 Data quantity

Due to the increasing equipment complexity and diversity of product requirements, numerous sensor data are collected during semiconductor processes [35]. Although the quantity of data can be effectively reduced after preprocessing, in which feature variables are extracted from each sensor signal; the resulting number of variables is usually much bigger than the number of sensors, due to the fact that various features are extracted from different steps of each sensor.

Traditional method evaluates each of the feature variables to look for any indicative pattern that coincides with the component life-cycle. This univariate approach suffers from huge difficulty in that a significant familiarity with the system is essential to eliminate irrelevant parameters from consideration at an early stage. Additional analysis would then be necessary to identify predictable patterns from among the remaining signal features. For these reasons, the identification of a single, useful tool condition indicator that is independent from process specifications is very difficult to discover by engineering analysis alone.
1.3.3 Process variability

As described in Section 1.2.1, different processes with distinct recipe definitions are often running on the same piece of equipment, thus the equipment degradation patterns are expected to be distinctive under different operation conditions. In addition, sensor readings are also affected by recipe set-points, maintenance orders, chamber matching activities etc., all of which can cause shifts in the signal. These practices will not only cause non-normally distributed variables, but they can also introduce different feature variables across recipes, which makes it difficult to evaluate the condition with a global model. Thus, equipment condition assessment becomes context-specific, and yet the prediction of RUL needs to be consistent with multiple processes being considered. For example, at any given point, in order to optimize production, one needs to know the RUL estimation for each recipe, which is affected by the operation of all the previous recipes. Those challenges need to be properly addressed in order to ensure the applicability of the proposed method.

1.4 Research Objectives and Broader Impacts

Taken collectively, Fig. 1.6 shows the procedure of the development of a prognostic solution. Generally, it can be divided into two analytical areas, namely offline analysis for predictability evaluation and model identification, and online analysis for RUL estimation and maintenance scheduling. After the feature variables are extracted from sensor data and historical records are cumulated, the developer first goes through an offline cycle to assess failure predictability and identify the degradation model if it is predictable. Once the model is valid for the failure model, online features can be fed in to estimate the RUL and determine maintenance.

In this work, we develop a unified methodology based on PLS regression method to systematically address the aforementioned challenges. The methodology developed
here will be focusing on the areas of predictability evaluation, root-cause diagnosis and RUL estimation, while the methods for feature extraction and maintenance scheduling are beyond the scope of this work and prior art will be reviewed in details in Chapter II. Under the proposed framework, the research objectives are:

- Establish a framework to evaluate the predictability of a failure mode and pro-
vide component RUL estimation.

- Develop an integrated diagnosis capability to assist trouble-shooting and root cause analysis.

- Define a strategy to consistently evaluate the impact of multiple processes on the equipment.

In addition, the multiprocess characteristics in the manufacturing industry can be generalized into a multi-regime engineering system, where multiple distinctive operations are observed during the life cycle. For example, an aircraft engine usually operates under different working conditions during various stages of the flight, thus the degradation pattern varies in these stages. At any time in the life cycle, the component condition is affected by the impacts from all previous periods with distinctive operations. Therefore, the method developed in this work is also applicable to the RUL estimation of those multi-regime engineering system as well.

1.5 Organization of the Dissertation

The rest of the dissertation is organized as follows:

Chapter II reviews the state-of-the-art CBM techniques with emphasis on semiconductor manufacturing. Methodologies from both academic research and industry use cases are reviewed for the four perspectives that were highlighted at the beginning of this chapter.

Chapter III presents the framework of the proposed method, detailing the theoretical background of the PLS algorithm and elaborating the procedure of using it for prognostic evaluation. The key procedures are first introduced for a single process problem. Then the characteristics and requirements of the multiprocess problem are highlighted, based on which, a strategy is defined to extend the PLS method for
multiprocess RUL modeling. Illustrative examples are given based on simulated data to validate the effectiveness for both cases.

Chapter IV presents semiconductor use cases with different predictability scenarios to evaluate the performance of the proposed method. Examples of both single and multiprocess are given, and the limitations of the method are identified.

Chapter V summarizes the contributions of the research work, identifies areas that need further improvements and makes recommendations for future work.
CHAPTER II

State-of-the-art of CBM Techniques in Semiconductor Manufacturing

As stated before, a complete prognostics solution consists of a work-flow of several standard steps. In this chapter, we provide a comprehensive review on methodologies in each area. Based on the survey results, we will highlight the gaps as well as challenges in current solutions, for which opportunities are identified for further improvement. Although data preparation techniques are beyond the scope of this research, as it is the essential step for various prognostic tasks, a review is provided to help understand the challenges.

2.1 Data Pre-processing and Feature Extraction

A review of common practices for data preprocessing can be found in [18, 23]. Cherry [18] has outlined a systematic way for data preprocessing. First the critical recipe steps need to be properly extracted based on the relevance of various process steps to various failure modes being considered. Since the transient period is often included in the beginning or end of a step, one needs to keep the stable part of the signal by removing samples from the step based on number or time such that consistent features can be obtained. Instead of using step ID as data separator,
Deprost and Ansquer [23] propose a windowing strategy using parameter conditions as the trigger for data selection. A signal is only selected when the condition imposed on a sensor is satisfied. In this way, transient data can be eliminated by specifying a range as the selection condition. The advantage of this approach is to implement engineering knowledge into the analysis and to avoid unnecessary processing given that expertise is available.

After critical adequate portions of signals are selected, they need to be further summarized into features. For most sensors with a specified target value during the process, simple statistics such as mean, standard deviation, maximum, minimum etc. of the signal are used as features. These static features are used in numerous publications, including [50, 41, 74, 60, 54, 58, 13, 2]. They are also most frequently used in industrial practice, simply because currently prevalent sampling rates of up to several Hertz for a great majority of sensors do not reveal highly dynamic phenomena in signals. Thus, the possibilities and needs for the use of more advanced feature extraction techniques, such as frequency or wavelet-based methods [28] are limited. An example of the use of a dynamic signal feature can be found in [60], where the slope describing the rate of the MFC flow increase between steps is calculated to indicate the pump health status on a PVD tool. It was discovered that a nonlinear trend of the slope values occurs as the pump starts degrading.

Another feature extraction example that differs from the aforementioned trend can be found in [24], where the coefficient of friction (CoF) during a Chemical Mechanical Planarization (CMP) process is estimated from the sensor readings and is directly used as the feature CMP process monitoring. Selective samples from some crucial trace points of a sensor reading can also be selected as feature variables. In [19], batch data from a metrology tool are unfolded so that the measurements of critical dimensions at different stage of the process are considered as features that reflect the process condition. Although this approach increases the number of input variable, the
temporal information within a dynamic process is preserved by creating time-specific variables.

One challenge in feature extraction is that most of the sensors available to the users of semiconductor manufacturing tools are used to indicate chemical process stability and serve the purpose of control. Therefore, even though the equipment condition is deteriorating, it is often impossible to detect the abnormality by monitoring that sensor alone, since the readings of those sensors are controlled to maintain their target. Instead, a system-based approach is needed, where both the controlled sensor reading as well as the relevant control inputs are available in order to detect abnormalities in the dynamic relations between them.

Another challenge related to feature extraction for CBM in semiconductor manufacturing is the fact that one same piece of equipment is often used to execute several different recipes and/or operations. As shown in Fig. 1.1 and Fig. 1.2, while similar recipes often have common steps and comparable ranges of sensor readings, with just the time durations of various steps being different, most of them have distinct steps and quite different ranges of sensor readings. During the execution of these variable recipes, the stochastic relations between sensor readings and the equipment condition change too. Furthermore, the degradation dynamics with which the tool condition deteriorates also change. Thus, the relations between features extracted out of sensor readings and the tool condition are usually inconsistent during the operation of a tool. For example, in [74] it was shown that when recipe changes on a CMP tool, the feature variable shows two distinct magnitudes with large variances. In this case, if the tool behavior model is built on the data from both recipes, the model will allow large feature variations and result in a large probability of missed fault detection ($\beta$ error). On the other hand, if the model is built only on one recipe, a false alarm will occur each time the other recipe is used, thus leading to a high probability of false alarms (large $\alpha$ error). Another example of the influence of the equipment operation
mode on the way sensor readings depict degradation can be found in [41], where the trend of the maximum of a critical sensor has been identified as the equipment condition indicator.

An intuitive solution to this problem is to build recipe specific condition degradation models for each recipe and to apply the appropriate degradation model as the tool operates. However, this strategy requires identification of a large number of degradation models. As each recipe only consumes a fraction of the total operation time of the tool, availability of data from identical recipes is limited, with numerous discontinuities because of recipe changes. An alternative solution is to find features that are independent of recipe changes so that a consistent indicator is available throughout the entire equipment operation time. In [87], the resistance is identified to have a monotonically increasing trend as the implanter source assembly degrades. A control limit is then imposed on the trend for PM scheduling. A similar strategy is employed by Carter et al. [13], who showed that die temperature is proportionally increasing when the vacuum capacitor wears out in a plasma power deliver system.

Unfortunately, the aforementioned strategy requires extensive engineering knowledge to identify such recipe independent features, especially when the number of failure modes and process parameters is large. Furthermore, the existence of such a variable for each type of failure that can be encountered on the tool usually cannot be guaranteed.

Several other alternatives were proposed to mitigate the recipe effect and to use the data in an integrated fashion. Roussy et al. [71] developed a normalization approach that uses various regression models to de-trend the data from a dry etch chamber. Even though results from [71] show that the normalized residuals obtained after removing regression trends from the raw data do not improve significantly the prediction accuracy of the etch metrology results compared with the non-normalized data, this strategy represents a potential way to deal with the variability of operating
conditions on the tool. In [41], 5 statistical features were calculated from the so called Key Sensitive Time-lot (KST, which is equivalent to critical step aforementioned in this review) of 3 process parameters, and one feature was identified as representative of the tool degradation regardless of the recipe. A product index, encoding product types processed on the tool, was used to mitigate the outliers associated with different product types. A moving average was further used to produce a predictable trend for the time series model.

Juge and Youlton [44] create an Enhanced Composite Variable (ECV) to eliminate the drift effects in extracted features caused by a tool maintenance or a lot change. First, a linear trend model is fitted to each critical parameter within each maintenance or product cycle, then the difference between actual and predicted value from regression model is used as the ECV for tool monitoring. By eliminating effects of maintenance operations or product changes, the ECV was shown to be a more consistent performance indicator, with smaller variance that led to a reduced false alarm rate. Chen [16] recently proposed a calculation of the so-called moving variance and moving covariance, which are obtained from the moving average of the time series formed by the conventional variance and covariance between sensors’ traces. It is shown that the results of moving variance and covariance are independent of the ranges of the sensor readings from different recipes, and it only depends on the patterns of sensor readings. Furthermore a generalized variance, defined as the determinant of the moving covariance matrix is introduced as the overall tool health indicator. Results of monitoring a PECVD tool showed that this method can effectively reduce the variance within similar recipe group, but its usefulness for modeling the degradation under arbitrary recipes is yet to be validated. Sakai et al. [74] highlight a procedure to extract recipe independent features by taking the difference between sensor readings and their corresponding set points, instead of the original sensor readings. Applications of the model to the Cu-CMP process with multiple recipes eliminates
unscheduled maintenance by predicting appropriate timing of predictive maintenance. This led to a reduction of 35% of scrapped wafers.

In summary, the characteristics of semiconductor manufacturing almost always necessitate that raw sensor and production data be preprocessed for further analysis and interpretation. As the wafer is processed through various multi-step recipes, data need to be segmented into corresponding data sections. In addition, the signal properties vary from step to step, and not all steps are related to the problem being considered. Therefore it is necessary to choose related signals and abstract the raw data from each step into some summary information for further analysis. Another fact in data that prompts for preprocessing is the transient period between steps. Although they may contain useful information, the low sampling rate of wafer process (normally 1 - 2 Hz) would limit our option to apply many advanced signal processing algorithms such as time-frequency, wavelet analysis etc. Thus, one can only expect to extract features in low frequency domain such as data changing rate, range etc. Data preprocessing is a critical step for the whole application in that it not only eliminates noise that contaminates data, but also reduces the amount of data for processing by extracting summary features.

2.2 Performance Assessment

At the performance assessment stage of CBM, historical records of extracted signal features and operational records of the equipment are used to build a model that describes the normal equipment behavior, based on which, one can flag abnormalities and failures if significant departures away from normal behavior are observed [66]. If models of various faulty behavior modes exist, one can then identify the root causes of the abnormal behavior and thus accomplish the fault diagnosis, which is highly important for subsequent activities to correct the fault.

Traditional methods combine detection and diagnostic into one step by monitoring
a critical sensor reading or a critical feature that is identified as the indicator of a
particular fault. As abnormalities are seen in a given indicator, the corresponding root
cause is then immediately found. Typical examples of this approach can be found in [9,
40, 17, 2, 25]. Brggemann and Lindner [9] proposed an approach based on monitoring
of DI temperature in a wet-cleaning process. A sudden drop of the DI-temperature
is related to a boiler problem and is being monitored by the corresponding SPC
chart for fault detection. In [2], a high performance VI probe was evaluated in a
plasma-enhanced Titanium deposition process. Its utilization was focused on finding
any reasonable differences between good and bad chambers classified in terms of end
of line yield data. Gaudet and Gilchrist [25] give another good example on fault
detection for a vacuum system on the semiconductor tool, in which the pressure
abnormalities in ion gauge can be correlated with below leak in wafer heater bellows
or life bellows. In [40], an interconnection failure in a copper damascene process
caused by the plating bath degradation in the copper electroplating equipment was
discovered using an SPC-based method. It is clarified that the root cause of the failure
is the plating bath degradation, caused by a byproduct, whose existence is confirmed
by analyzing the bath plating using high performance liquid-chromatograph (HPLC).
In [17], a top quartz window of an etch tool was the object of maintenance. A dry clean
process is used to remove the byproduct on the window surface and keep chamber
condition stable. The dry clean time is correlated with roughness and can be used as
an indicator for PM. Instead of fitting regression model to predict the value of dry
clean time in future, a SPC chart is used to monitor the dry clean time and to provide
early alarm. When out of control status is detected, the quartz window is replaced.
In [13], the life cycle of a plasma power delivery system is divided into 3 periods and
monitoring strategies are defined for each stage. Engineering analysis is applied to
identify critical variables for performance monitoring in each of the 3 stages.

Although the above SPC approach is validated as an effective fault detection
method, it can be further improved by incorporating engineering knowledge or expertise. In [15], Chen et al. develop an alternative mean estimator for early detection of process shift. A Lag-Window is defined for the observation, using Bayes’ principle, the probability that the shift starts at each of the previous sample can be estimated. Then the new mean estimation for the latest observation is obtained by taking the weighted summation of previous samples in the lag-window. In this way, engineering expertise can be taken into account by specifying different types of prior distribution for the Bayes estimation. An example from $SiO_2$ oxidation process demonstrates the propose approach outperforms the traditional CUSUM chart [65] in that the actual process shift is detected earlier. Bousetta and Cross [8] examined different SPC techniques used to cope with excursion data and help maintain an optimal balance between alpha and beta risks by applying them to real data from a production line. These techniques are compared using metrics such as practicality, effectiveness and robustness. Both Type I and Type II errors of control charts with Shewhart limits, Poisson limits, Tukey limits [65] and empirical limits are evaluated with data from a production process, based on which recommendations for improvements of each of the tested strategy are made by the author.

Due to the characteristics of the semiconductor manufacturing process, the SPC-based approaches mentioned above encountered many challenges in processing data effectively as well as presenting results informatively such that one can rapidly detect and diagnose numerous faults that could occur in semiconductor manufacturing. First, multivariable analysis capability is essential when the number of sensors increases, which is very common now that the semiconductor manufacturing tools have become highly sophisticated. The majority of approaches addressing this issue are based on multivariate statistics. Within the family of statistical approaches, parametric models such as $T^2$ statistics, PCA [43], or Markov chain [26] have received wide applicability because of their capability to simultaneously deal with multiple
random variables. In [50], the authors detect abnormalities using the $T^2$ statistic calculated from the variance-covariance matrix of the most recently emitted sensor features and the variance-covariance matrix of the sensor features emitted during normal equipment operation. Statistical limits on the $T^2$ statistic can be theoretically set and abnormalities are detected when the $T^2$ statistic violates those limits. The same anomaly detection methodology applied to variance-covariance matrix sub-blocks corresponding to feature sets indicative of a particular fault is suggested as the tool for the corresponding fault diagnosis. Such an approach to diagnosis requires the existence of a knowledge-base associating feature sets with various equipment fault and failures and such a database can be, for example, built from historical operational records of the tool. Cherry [18] also provides a method for PCA-based fault detection. Instead of directly applying $T^2$ statistics, the covariance matrix of signal features extracted during normal system behavior is decomposed and reconstructed in its eigen-space using PCA. The use of PCA is intended to capture the principal correlations within variables, while neglecting insignificant relations. A combined fault detection index, taking into account both squared prediction errors and the $T^2$ statistics, is developed and is known to follow the well-known $\chi^2$ distribution during normal process behavior. Thus, abnormality detection can be accomplished based on control limits set with statistical rigor.

Unfortunately, a strong assumption underlying the use of the $T^2$ statistic or PCA is that data under investigation follow a Gaussian distribution. This condition cannot always be guaranteed, especially when equipment is operating under different operation conditions. Furthermore, the very degradation of equipment introduces departures away from Gaussianity since the feature distributions shift over time. Depending on the dynamics of the change, there seem to be two main approaches to address this issue. The work reported in [78, 18] deals with the situation when distributions of sensory features are locally Gaussian, and depart from Gaussianity.
over longer time-scales. On the other hand, [58, 24, 84, 20, 81, 30] consider situations when even local Gaussianity of features cannot be attained.

Local feature Gaussianity with continuous updates of distribution parameters is pursued in [78] where work on condition monitoring of an etching process is reported. One adaptation strategy tested in [78] updates only the mean and standard deviation coefficients used for centering and scaling of the new data, thus accommodating for the drift of the process. The second strategy tested in [78] is based on frequently rebuilding the covariance matrix from the most recent data to update the statistical dependency. An experimental test with known failure modes is conducted to validate these strategies. More recently, a recursive PCA method is developed in [18] where the update of parameters is done through a recursive formula with forgetting factor. This approach updates parameter rapidly and has advantage with large number of variables.

A more challenging situation arises when quasi-Gaussianity over short time frames cannot be assumed. Mao and Holland [58] use a modified PCA for an implanter fault detection. A kernel density estimator is proposed to estimate a non-Gaussian distribution, so that the proper control limit for fault detection can be defined. In addition, selective samples were used to update the model in order to take into account potential process drift or variance changes. A similar approach is also proposed by Wang and Chen [84] for a plasma etcher application, in which two transformation schemes are introduced to convert the non-Gaussian features into normality conforming forms. Then, the Johnson’s transformation system described in [42] was directly applied to the input variables or on the principal components from PCA. In [20], the authors propose mixture Gaussian models [63] for non-Gaussian density estimation in order to reduce the complexity and computational effort of the kernel methods. In [81], it shows that regular control limits with assumed Gaussian distribution exclude a substantial amount of training samples that are collected during normal operation.
condition of the equipment, and hence yield high false alarm rate. The authors propose a so-called knowledge-based control limits, obtained by gradually expanding the elliptical Gaussian control limits, with the constraints from a rectangular engineering limits, until the new limits include all training samples from normal operation condition. A case study showed that false alarm rate was dramatically reduced with the new knowledge-based control limits. Harel and Adam [30] use the Monte-Carlo method to estimate failure probability distribution, with a Markov state machine employed as the tool for modeling of the condition status. In [24], a multi-scale Bayesian sequential probability ratio test (MBSPRT) is developed, and was shown to be efficient in monitoring processes with signal features that do not follow a Gaussian distribution. The efficacy of this method was tested via detection of the end point occurrence in a chemical-mechanical planarization (CMP) process, using the coefficient of friction (CoF) estimates as process features. Test results from both oxide and copper metal CMP are presented, showing that MBSPRT is capable of identifying the start and finish of the end point event.

He and Wang [31, 32] developed a K-Nearest Neighbors (KNN)-based fault detection approach. The idea of this approach is that sample data from normal operation condition should have certain concentration while data from faulty condition should demonstrate deviation from this concentration; therefore the sum square distance of each one of normal sample with its k nearest neighbors can be characterized by a non-central $\chi^2$ distribution. And by imposing a threshold with some confidence level on the distribution, an unknown sample can be considered normal if the sum square distance with its k nearest neighbors is below the threshold; otherwise it is detected as a fault. Since the KNN classification rule does not make any assumption of normal distributed data, and examples showed that it is able to detect fault that from a non-Gaussian process.

Independent Component Analysis (ICA) is also reported for non-Gaussian process
Unlike PCA approach, ICA seeks a different data representation by expressing them as a linear combination of some independent components, in the meantime the variables of reconstructed data would also be as independent as possible. Lee et al. [48] proposed three different indexes for fault detection. As these indexes depart from Gaussian assumption, it can be used to monitor non-Gaussian process and kernel density estimation is adopted to evaluate the distributions of the fault detection index. Similarly with PCA approach, a threshold with associated confidence value can be specified as fault criterion for detection. Based on this philosophy, several other approaches are proposed to enhance the performance of ICA-based method. A dynamic ICA (DICA) is developed in [48] to augment the observed data matrix by adding time-lagged observations. In [39] Adjusted Outlyingness metric (AO) [10] is utilized for rejecting outliers and online process monitoring. And in [38] Support Vector Machines (SVM) classification [76] is used instead of the aforementioned fault detection indexes to address the issue of kernel density estimation which yields poor performance as index is usually autocorrelated over the process.

Finally, instead of detecting fault for the process, the FDC methods are also used for equipment setup or chamber matching. In [49], calibrated tools which are known in good operation conditions are used to characterize the ideal status based on which the fault detection model is built. The model is then used to evaluate data from new tools or those that just underwent a PM to compare their conditions with the "golden" dataset baseline. The tools are ultimately qualified for production only if the model does not indicate any abnormality compared to the baseline dataset.

2.3 Fault Diagnostics

Fault diagnosis refers to the process of characterizing and recognizing the root cause once the presence of a fault is detected. Depending on how fault detection is accomplished, the diagnosis function can be accomplished in various ways. For
univariate SPC-based fault detection, the diagnosis stage is straightforward if the monitored variable is directly related to a fault in the system. Such integrated fault detection and diagnosis schemes are described in [9, 15, 8, 40, 45, 25, 80].

For multivariable techniques, the fault diagnosis is more complicated. One approach to the multivariate root cause identification is based on decomposing some overall multivariate fault index used for fault detections into contributions from different sources that are indicative of various faults. For PCA-based fault detection methods, contributions are identified by calculating the statistics from sub-matrices of the covariance matrix related to feature subsets deemed indicative of a particular fault. Based on such covariance matrix sub-blocks analysis, a reasoning strategy is described in [18], where the variables are grouped into a hierarchical structure, that enables the user to drill down through the results from the overall fault index to individual sensors and recipe steps to identifying the source of the problem. For $T^2$ based fault detection [50], by applying regression adjustment method that described in [59], the total $T^2$ statistic can be decomposed into a series of conditional contributors, in which each one has a corresponding failure mode; thus the major contributor of the decomposition indicates the failure type whenever an alarm is given.

In addition to statistical methods, artificial intelligence based algorithms like neural networks, decision trees etc. have been reported as methods accomplishing condition diagnosis in semiconductor manufacturing. May and Spanos [61] use evidential reasoning for automated equipment diagnosis. This technique combines the quantitative algorithmic diagnosis and qualitative knowledge-based approach, integrating evidence from equipment maintenance history, real-time tool data and in-line measurements. A plasma etching tool was used as an example for validation. Heavlin and Koslov [34] applied a Mantel-Haenszel-based approach [57] for reliability analysis of reparable systems. In [82], decision tree is used for equipment commonality analysis. When defects are found on a wafer, one needs to determine which tools during the
whole multi-step process are responsible. A defect classification tree whose leaves correspond to each equipment that was used during the whole process is built to help engineers diagnose the problem. It is shown that the tools from the optimal tree whose defect classification rate is closest to real defect rate are the most probable cause of the problem. In [83], a disturbance detection and classification method is developed using Bayesian statistics, which is applied to detect and classify various disturbance types. The type of disturbance is identified by matching the pattern from posterior probabilities of the system states to the pre-defined patterns that associated with different known disturbance types.

2.4 Performance Prediction

Equipment condition monitoring has received significant attention in the semiconductor industry in recent years. Different algorithms, such as univariate SPC [27], multivariate principal component analysis (PCA) [19, 89, 77, 47], and k-nearest neighbor (KNN) [31, 14], are some of the more popular methods for fault detection and diagnosis.

But while the methods of equipment condition monitoring have been around for some time, efforts on equipment fault prediction are only now gaining momentum. In the fault prediction realm, most of the recently reported methods are based on a univariate approach. The procedure starts by identifying a feature variable with a consistent, predictable pattern (e.g. a monotonic trend) that relates to the component condition. Then a regression model, such as a time series model, is built to capture the dynamics of the indicator and provide prediction of its future value. Finally, given a failure criterion, such as a threshold of the condition indicator, one is able to predict the likelihood of the failure in the future. Since this univariate approach is straightforward and relatively easy to implement, it is commonly used in many fault prediction applications [79, 60, 87].
Sugimoto [79] predictively scheduled chamber cleaning events based on the relations between the number of wafers processed in the chamber and particle density on the chamber wall under different chamber initial conditions. With experimental wafers running in the chamber, a system of differential equations is obtained, based on which the particle density could be predicted for production wafers. An empirical threshold is then imposed on the predicted particle density to trigger chamber cleaning in order to avoid undesired particle outbreaks. Matsuhashi [60] gives another example of condition prediction for a deposition tool. The film deposition process on a wafer was divided into several windows with summary statistics calculated as features. Critical indicators are identified by engineering experience and the Auto-regressive Moving Average time series model was used to predict their values. In [36], the illumination power is used as an indicator of optical element performance. Since the power degradation is a function of several factors, 3 different patterns of power decrease were identified and thus, its decay over time was modeled for the purpose of prediction by 3 distinct trend lines using polynomial fitting. Wang et al. [87] present a case study of predictive condition monitoring of an implanter source assembly. An increasing trend is identified in the resistance signal before the source filament needs to be changed. Therefore, the regression model was built to predict the future value of filament resistance and determine when the source needs replacement.

Due to the increasing equipment complexity and diversity of product requirements, numerous sensor data are collected during semiconductor processes [35]. This creates a disadvantage for the univariate approach, in that a significant familiarity with the system is essential to eliminate irrelevant parameters from consideration at an early stage. Additional analysis would then be necessary to identify predictable patterns from among the remaining signal features. For these reasons, the identification of a single, useful tool condition indicator that is independent from process specifications is very difficult to discover by engineering analysis alone.
Instead of simultaneously predicting the future value of multiple variables and deciding if there will be an imminent failure, the concept of remaining useful life (RUL) is developed to facilitate prognostics by converting multivariate historical data into a scalar indicator of future usefulness of the equipment. RUL is originally defined as a random variable $T$ that represents the random time to failure of an item [3]. At a given time $t$, $X_t = T - t$ is the RUL of $T$, then a mean residual life, $\mu(t) = E(T - t | T > t)$, is used as the estimation of RUL. Under this scheme, several statistical models are applied to evaluate RUL, such as proportional hazard model (PHM) [52, 51], Markov model [4, 11], particle filtering [68], and Bayesian statistics [72]. Over the years, in addition to probabilistic estimation, different approaches are also developed to directly calculate system RUL. Examples include instance-based methods such as fuzzy logic [90], neural networks [62], similarity-based data fusion [86]; as well as regression-based method such as logistic regression [91], Bayesian regression [73] and Cox regression [88].

Although these RUL estimation approaches are actively applied in various areas and have demonstrated their success in different applications, difficulties remain in adopting them for fault prediction of semiconductor equipment. For example, the presence of various mechanical and electrical components within a single semiconductor tool makes it difficult to determine the appropriate type of statistical function. For instance-based methods, the main goal is to determine the similarity between various samples such that the relevant ones can be identified. But this is not really a challenge for semiconductor process since the operational context is strictly defined and known to the engineers. Therefore, their applicability is questionable. As for regression-based methods, the target RUL of logistic regression will have poor resolution at the beginning and the end of the life-cycle. This makes it hard to distinguish the tool condition because the RUL can be asymptotic to 0 and 1. Finally, the application of Bayesian regression is infeasible because it requires certain knowledge of
the system in order to determine proper priori.

2.5 Summary

In this chapter, we reviewed the CBM methodologies and recognized its opportunity in semiconductor manufacturing. The CBM practice is outlined as a standard procedure with 4 steps, for each step we surveyed and summarized industrial applications as well as research activities. In the area of data pre-processing, dominant applications extract information based on summarized features of recipe steps. It has been recently brought into attention the large variation of data due to recipe or product type changes, several strategies are proposed from the research community to normalize data and have a consistent representation across different operation conditions. In addition, the transient periods between recipe steps are mainly neglected in most applications, but for equipment degradation those data may provide valuable information. Due to the low sampling rate of data collection, advanced signal processing algorithms are not applicable. These two challenges should be the focus of future research.

For fault detection, although traditional SPC approaches can still be found in industry, the majority of the applications have adopted multivariable analysis such as PCA, $\chi^2$ or Markov methods. For non-Gaussian data, methods such as the kernel algorithm, mixture model, and Bayesian analysis for probability density estimation are proposed but the application is limited due to their requirement of large calculation efforts.

For fault diagnosis, while variable contribution evaluation is mainly used for diagnosis in multivariable analysis, other algorithms like neural networks, decision tree, and Bayesian analysis are also tested as research topics.

For prediction, most applications today are still heavily relying on univariate methods with expertise to identify indicative failure variables. Time series models
are used for prediction once the ‘golden trace’ is found. As for nonlinear degradation pattern, instead of predicting the value of indicative signals, early warning is given for maintenance scheduling by statistically comparing the proximity between historical failure and current operation. Future works are essential in this area to address the need of multivariable analysis.
CHAPTER III

Methodology: Partial Least Square Analysis for Predictability Evaluation, Remaining Useful Life Estimation and Root-cause Analysis

In this chapter, we develop a method for equipment RUL estimation based on Partial Least Square (PLS) regression [76]. Unlike other regression methods, PLS selects regressors from directions in the latent space of covariance that bear the largest correlation between inputs and outputs. Under this framework, a group of generalized RUL curves are defined to represent different degradation profiles during the component life-cycle. With run-to-failure training data, the PLS model is built and an optimal RUL, which has the smallest validation error, is chosen from the group. Based on the shape of the optimal RUL, one is able to evaluate variations that exist in the input space and determine the predictability of the failure. In addition, by evaluating the blocking contribution of subsets of the variables, one can quickly identify the source of the variations that cause the degradation. Finally, with a selection procedure to eliminate variables with small contribution to RUL change, the noise level in the model is dramatically reduced, thus improving prediction accuracy.

As the product diversity keeps increasing, various manufacturing processes with distinctive conditions are frequently running on the same piece of equipment due to
production capacity limitation or manufacturing cost control. Thus, it is important to have consistent evaluation of equipment health condition with data collected from different processes.

The remainder of the chapter is organized as follows. Section 3.1 briefly reviews the theoretical background of PLS algorithm. Section 3.2 elaborates on the details of using it for prognostic evaluation for a single process scenario, and several modeling issues are addressed with various statistical techniques. Section 3.3 conducts simulation to the proposed strategy and evaluates its performance. Then in section 3.4, we highlight the challenges and requirements for multiprocess equipment prognostics, and review some current solutions in the literature. Finally, section 3.5 illustrates the extended PLS strategy with simulation in section 3.6 to validate the performance.

3.1 PLS Regression Algorithm

In a least squares regression problem, we seek a matrix $W \in \mathbb{R}^{m \times n}$ that solves

$$
\min_W \|XW - Y\|_F^2,
$$

(3.1)

where $X \in \mathbb{R}^{l \times m}$ contains $l$ sample feature vectors as rows and $m$ variables as columns. The corresponding desired outputs are stored in $Y \in \mathbb{R}^{l \times n}$ with a column dimension of $n$ response variables. The covariance between $X$ and $Y$ can be obtained as:

$$
C_{xy} = \frac{1}{l} X'Y.
$$

(3.2)
The directions that solve the maximal covariance optimization

\[
\max_{w_x, w_y} w'_x C_{xy} w_y = \frac{1}{l} w'_x X' Y w_y, \tag{3.3}
\]

subject to \( \|w_x\|_2 = \|w_y\|_2 = 1, \) \( \tag{3.4} \)

are the first singular vectors \( w_x = u_1 \) and \( w_y = v_1 \) of the singular value decomposition of \( C_{xy} = U \Sigma V' \). Reader please refer to Appendix A for proof of this conclusion.

Therefore we can project input \( X \) onto \( k \) directions which have the biggest covariance and performs a least squares regression between the latent features \( XU_k \) and unchanged \( Y \), we seek a matrix of coefficients \( B \in \mathbb{R}^{k \times n} \) that solves the optimization

\[
\min_B \|XU_k B - Y\|_F^2, \tag{3.5}
\]

where \( U_k \) is the matrix formed of the first \( k \) columns of \( U \). However, there is an implicit restriction as there are only \( \min(m, n) \) non-zero singular values of \( C_{xy} \), therefore \( k < \min(m, n) \). The PLS algorithm utilizes the following procedure to overcome this constraint by selecting feature directions iteratively from the covariance matrix formed by deflation of input \( X \) and unchanged output \( Y \):

1: input: \( X = X_1, Y \)
2: for \( i = 1 \) to \( k \) do
3: \quad let \( u_j, v_j, \sigma_j \) be the first singular vectors and value of \( X'_j Y \)
4: \quad \( X_{j+1} = (I - \frac{X_j u_j u'_j X'_j}{u'_j X'_j X_j u_j}) X_j \)
5: end for
6: output: Feature directions \( U_k \) with columns of \( u_j \) for \( j = 1 \ldots k \).

The reason that \( Y \) is unchanged is because even if it is deflated, the fact that we are only removing the explained covariance means it will have no effect on the extraction of subsequent features. Therefore, one can continue to extract hidden
features as long as there continues to be explainable variance in $Y$, typically when $k > \min(m, n)$. To calculate the regression coefficients for the test point with feature vector $\phi(x)$, the transformations that are performed at each step of extracting $X$ should also be applied to $\phi(x)$ to create a series of feature vectors

$$\phi_{j+1}(x)' = \phi_j(x)'(I - u_jp_j')$$  \hspace{1cm} (3.6)

where $p_j = \frac{x_j'x_ju_j}{u_jx_j'x_ju_j}$. Similarly we have

$$\phi(x)' = \phi_{k+1}(x)' + \sum_{j=1}^{k} \phi_j'u_jp_j'. \hspace{1cm} (3.7)$$

The feature vector that for regression $\hat{\phi}(x)$ has components $\hat{\phi}(x) = (\phi_j(x)'u_j)_{j=1}^k$, consider

$$\phi(x)'U = \phi_{k+1}(x)'U + \sum_{j=1}^{k} \phi_j(x)'u_jp_j'U$$

$$= \phi_{k+1}(x)'U + \hat{\phi}(x)'P'U, \hspace{1cm} (3.8)$$

where $P$ is the matrix with columns of $p_j$ for $j=1\ldots k$. Since for $s > j$, $(I - u_s p_s')u_j = u_j$, we have $\phi_{k+1}(x)'u_j = \phi_k(x)'(I - u_k p_k')u_j = 0$. Then, the new feature vector for regression is

$$\hat{\phi}(x) = \phi(x)'U(P'U)^{-1} \hspace{1cm} (3.9)$$

since the regression coefficients for dimension $j$ of the new feature vector is $\frac{\sigma_j}{u_jx_j'x_ju_j}v_j'$, where $v_j$ is the complementary singular vector corresponding with $u_j$ so that $\sigma_j v_j = Y'Xu_j$. Thus, the overall regression coefficient can be calculated as

$$W = U(P'U)^{-1}C', \hspace{1cm} (3.10)$$

37
where $C$ is the matrix with columns $c_j = \frac{Y'X_ju_j}{u_j'X_jX_ju_j}$.

### 3.2 Prognostics Evaluation for Single Process Operation

This section elaborates the details of using PLS regression RUL modeling, issues include RUL definition, optimal model and parameter selection, predictability evaluation, and degradation root-cause analysis.

#### 3.2.1 Definition of Generalized RUL

Assuming $m$ feature variables extracted from sensor readings are used as inputs for the PLS algorithm, and the RUL of a component is specified as the output. To build the model, a set of $l$ training samples, $X \in \mathbb{R}^{l \times m}$, and their corresponding target RUL values, $Y \in \mathbb{R}^{l \times 1}$, are used to estimate the model parameters, $W$, based on Equation (3.10). In order to facilitate decision-making for maintenance scheduling, we define $Y$ as the number of wafers the tool will be able to process before the component needs replacement. Based on this scheme, $Y$ is a linearly decreasing function that counts the number of wafers from $l - 1$ to 0. It is important that the training samples be collected from the end of a run-to-failure process so that the 0-valued RUL corresponds to the worst condition before the component is replaced.

However, when setting the number of training samples, $l$, it is not necessary to include data from the whole component life-cycle. Furthermore, the $l - 1$-valued RUL does not need to indicate a perfect condition. Once the model gets built, it is used to identify the variations within input that best correlate with the component degradation indicated by the target training RUL. This allows it to be used for the RUL prediction when feature variables are extracted from the process that are being predicted. Based on the RUL prediction, one can adjust the production plan such that tool usage is maximized while avoiding unscheduled downtime.

Although in the long-term a monotonically decreasing trend does reflect a con-
tinuously deteriorating process, the linear target RUL used above may not always accurately represent the real degradation profile of the equipment. Because semiconductor tools have various components, each with distinct characteristics, their degrading rates can vary throughout the degradation process.

In order to discover the best degradation pattern that can be predicted by the variations in the input, a group of \( n \) generalized RUL (GRUL) curves \( \mathbf{Y} \in \mathbb{R}^{l \times n} \) between \( l - 1 \) and 0 is defined, instead of a single linear trend. For simplicity, the target GRUL group in this work is characterized by a truncated quadratic function. Without losing generality, the training sample size is normalized to a range from 0 to 1, and the target GRUL group is given by \( GRUL = ax^2 + bx + c \). Considering the constraints \( GRUL_{x=0} = 1 \) and \( GRUL_{x=1} = 0 \), the shape of target GRUL is determined by one free parameter \( a \) with definition

\[
GRUL = ax^2 - (1 + a)x + 1 \tag{3.11}
\]

Furthermore, in order to ensure that all GRUL curves are monotonically decreasing, values of the curves that are less than 0 or greater than 1 are truncated to fixed values of 0 or 1, respectively.

Fig. 3.1 shows an example of a target GRUL group of 30 curves with \( a \) evenly spread through the interval of \([-4, 4]\). With a smaller value of \( a \), the GRUL curves towards the upper right, which corresponds to a process where the condition is stable for most of the modeling period but has a sudden deterioration at the end. Alternatively, a larger \( a \) defines a process in which the condition changes rapidly at the beginning and before flattening out as the failure approaches. The linear case mentioned above is achieved when \( a = 0 \), giving a target RUL of \( 1 - x \).

The target GRUL curves are intended to express a realistic library of patterns that can be correlated to a wide variety of component degradation processes. An
optimal GRUL can be identified by the PLS algorithm as the one with smallest root mean square errors of the prediction, according to

\[
GRUL_o = \arg\min_{o \in [1-n]} \sqrt{\sum_{i=1}^l (GRUL_{i,o} - x_i w_o)^2 / l}
\] (3.12)

where \(x_i\) is row \(i\) of \(X\) and \(w_o\) is column \(o\) of \(W\). The optimal GRUL indicates the best degradation characteristic that can be represented by the variations within the input.

### 3.2.2 Selection of Optimal PLS Order

A free parameter of the PLS algorithm that needs to be optimized is the order of the model, namely number of input deflation, \(k_o\). In general, according to the parsimony principle, a simpler model representation is preferred given that all other things are equal.
One of the most common methods to select PLS model order is based upon the percentage of variance that explained by the model, which is indicated by the cumulative summation of the normalized eigenvalue sequence, \( \lambda_i \), for \( i = 1, 2, \ldots, k \), that given by each deflation of the input. It is assumed that the majority variance captured by the model is coherent with the physical process it deems to describe, while the rest are due to noise. With this philosophy, a threshold \( T \) is imposed on the percentage, and the optimal \( k \) is determined as

\[
k_o = \arg \max_k \left[ \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{k} \lambda_i} \leq T \right].
\]  

(3.13)

Another way to determine \( k_o \) is to evaluate the significance of model improvement, which is measured by the reduction of prediction error due to the increment of \( k \). In the process, \( k \) starts from 1 and increases until the improvement is insignificant. For the univariate output regression, the significance of improvement can be statistically determined by a \( F \)-test. However, as we define a group of target RULs, the distribution of prediction error from a multivariate output regression model is not trivial.

In this work, we apply likelihood ratio criterion to test the linear hypothesis about regression coefficients in the latent space spanned by PLS principal components. Suppose \( t_\alpha, \alpha = 1, 2, \ldots, l \) are the projections of original input \( X \) onto the latent directions given by PLS, and \( y_i, \ i = 1, 2, \ldots, l \) is the corresponding RUL vector for each \( t_\alpha \). For a least square regression problem, it is equivalent to consider that \( y_i \) are a set of \( N \) independent observations being drawn from \( N(Bt_\alpha, \Sigma) \), where \( B \) is the regression coefficient matrix of \( n \times k \), with \( n \) being the number of RULs in the target group and \( k \) the order of PLS model, and \( \Sigma \) is a \( n \times n \) covariance matrix of input variables. According to [1], the maximum likelihood estimation of \( B \) and \( \Sigma \) are given
by

$$\hat{B} = CA^{-1} \quad (3.14)$$

$$\hat{\Sigma} = \frac{1}{l} \sum_{\alpha=1}^{l} (y_\alpha - \hat{B}t_\alpha)(y_\alpha - \hat{B}t_\alpha)^\prime, \quad (3.15)$$

where matrices $C$ and $A$ are given by

$$C = \sum_{\alpha=1}^{l} y_\alpha t_\alpha^\prime \quad (3.16)$$

$$A = \sum_{\alpha=1}^{l} t_\alpha t_\alpha^\prime. \quad (3.17)$$

As the PLS model order increases by 1, an additional column is added to $B$ matrix. Suppose we partition a $B$ matrix with $k$ columns into ($B_1, B_2$) such that $B_1$ has $k - 1$ columns and $B_2$ is the last column of $B$. The hypothesis for a sufficient model order as $k - 1$ is

$$H : B_2 = 0. \quad (3.18)$$

The maximum of the likelihood function $L$ for the sample $y_1, \ldots, y_l$ based on a model with order $k$ is

$$\max_{B, \Sigma} L = (2\pi)^{-\frac{l}{2}ml} \left| \hat{\Sigma}_{\Omega} \right|^{\frac{1}{2}l} e^{-\frac{1}{2}ml}, \quad (3.19)$$

where $\hat{\Sigma}_{\Omega}$ is the unrestricted sample covariance matrix given by Eq. (3.15). Similarly, when the model takes order as $k - 1$, the maximum of the likelihood function would take the same form as Eq. (3.19) with the covariance matrix $\Sigma$ estimated based on $B_1$ and the corresponding partition input $t_1^\alpha$, which is the first $k - 1$ elements of $t_\alpha$. Thus, the likelihood ratio criterion for testing $H$ is

$$\lambda_k = \frac{\max_{B_1, \Sigma} L}{\max_{B, \Sigma} L} = \left| \frac{\hat{\Sigma}_{\Omega}}{\hat{\Sigma}_{\omega}} \right|^{\frac{1}{2}l}, \quad (3.20)$$
where \( \hat{\Sigma}_\omega \) is the covariance matrix estimator for model of order \( k - 1 \). In testing \( H \), one rejects the hypothesis if \( \lambda < \lambda_\alpha \), which follows a Wishart distribution with significant level \( \alpha \). Readers are encouraged to review reference [1] for the details of the Wishart distribution.

To this end, the procedure to select the optimal PLS model order is as follows:

1. Set \( k \) from 2, estimate sample covariance matrix for order \( k \) and \( k - 1 \), respectively, according to Eq. (3.15).

2. Evaluate the likelihood ratio \( \lambda_k \), and compare with \( \lambda_\alpha \).

3. If \( \lambda_k \) is in the reject zone, reject \( H \) and increase \( k \) by 1, repeat the process until \( \lambda_k \geq \lambda_\alpha \), then the optimal PLS order is \( k \).

### 3.2.3 Selection of Optimal Target GRUL

The optimal target GRUL is chosen by cross-validation [75]. For a training sample with size \( l \), the so-called \( N \)-fold cross-validation randomly divides it into \( N \) partitions. Then the training process repeats \( N \) times with each one of the sample partition being left out as validation records while the rest \( N - 1 \) partitions are used to build the model. When \( N = l \), it is also called leave-one-out cross validation as there is only one point being left for validation during each training. In our testing a 10-fold cross-validation is adopted and the optimal target GRUL curve is chosen based on Eq. (3.12) with minimum aggregated prediction error from the 10 training cycles.

### 3.2.4 Predictability Evaluation

Once the PLS model is built based on training data, the shape of the optimal GRUL will assist to determine the predictability of the problem and to define a suitable maintenance strategy. For example, consider the case where training data from \( l \) wafers are used to find an optimal GRUL with \( a = -0.7 \) (denoted by o’s
in Fig. 3.1). The gradual trend indicates that the model is able to distinguish the component condition very well for the entire training period. This indicates that we are able to predict RUL at least \( l \) wafers before the component fails. Therefore, as long as \( l \) is larger than the lead time window for maintenance scheduling, the failure is predictable and a predictive maintenance strategy should be adopted. Otherwise the number of training samples should be increased to check if the prediction model is able to provide longer RUL estimation before failure.

On the other hand, if the optimal GRUL is chosen with \( a = -3.4 \) (denoted by \( \triangle \)'s in Fig. 3.1), the biggest lead time before failure that the model is able to provide is only about \( 0.25 \times l \). If this is sufficient for maintenance scheduling, a predictive strategy could be implemented. If not, the RUL estimation can only be used as a fault detection solution, and condition-based maintenance should be considered instead.

Finally, if the optimal GRUL is chosen with \( a = 2.3 \) (denoted by \( * \)'s in Fig. 3.1), one is only able to observe condition changes early in the life-cycle, and would be unable to recognize any degradation immediately prior to the failure. In this case, neither prediction nor detection are possible and a preventive maintenance strategy should be implemented. This type of degradation pattern usually can be found in electrical components whose performance may be unreliable at its initial usage, after which it becomes stable for a length of time before the failure.

A special consideration should be made for how the model would react when there is no significant variation or trends in the input. In this case, the optimal GRUL would be chosen to be either of the two groups at the far ends of the chart, because those are the targets that fit the random noise the best. When either of these groups is chosen, it will be important to identify if correlation actually exists between the inputs and the RUL. Specifically, it should be evaluated how well the decreasing pattern in RUL correlates with the loading scores of the model. The scores, calculated as \( T = XU \), are the projections of original input on the principal directions in the latent space.
which have the best covariance with target RUL. If both the RUL and the loading score exhibit similar trends, then variation exists in the inputs to correlate with the condition change. If not, then the pattern indicated in the RUL is not the true representation of the tool condition. A use case in section 4.1.4 will further illustrate this scenario.

3.2.5 Degradation Root-cause Analysis

As mentioned in Section 3.1, PLS projects variables from inputs $\mathbf{X} \in \mathbb{R}^{l \times m}$ onto the direction on which the latent variables $\mathbf{T} \in \mathbb{R}^{l \times k}$ have the biggest correlation with output $\mathbf{Y} \in \mathbb{R}^{l \times m}$. Therefore, the influence of the input variables in defining the principal directions of the latent space is an indication of the correlation between the input and the component degradation. Suppose the normalized loading vectors of a PLS model with order $k$ are stored in the matrix $\mathbf{U} \in \mathbb{R}^{m \times k}$, by evaluating the absolute value of elements $u_{ij}$ one can identify the contribution of variable $i$ in component $j$ to the RUL of the equipment. In other words, if $u_{ij}$ is relatively larger than the rest element $u_{.j}$, it means the latent direction $\mathbf{u}_j$ is leaning more towards the subcomponent of the vector and thus the variation from variable $i$ will dominant in the latent variable $\mathbf{t}_j$. On the other hand, if $u_{ij}$ is very small in $\mathbf{u}_j$ then the principal direction is more perpendicular to this subcomponent, whose variation will be in turn suppressed in $\mathbf{t}_j$.

Furthermore, for each principal direction $\mathbf{u}_j$, the variation it explains in the model is given by the corresponding eigenvalue $d_j$ from SVD. Thus we can aggregate the contributions from all principal directions based on the weights of their eigenvalues, and the total contribution $\mathbf{c}$ of each variable is given by:

$$c = \sqrt{\frac{1}{\sum_{j=1}^{k} d_j} \sum_{j=1}^{k} u_{ij}^2 \times d_j}$$

(3.21)
However, when the number of variables is large, it is time consuming to evaluate the contribution of each variable and the trouble-shooting process will take long. In this case, it is desired to define hierarchy groups of the variables so that the contribution of similar variables will be combined together and evaluated first in a block fashion. And then the root cause can be located by drilling down the hierarchy groups. In this manner, the number of contributions one needs to evaluate is dramatically reduced. The contribution of each hierarchy block $c_b$ can be evaluated by taking the norm of the elements that corresponding to the variables in that block, namely:

$$c_b = \sqrt{B_k c^2},$$  \hspace{1cm} (3.22)

where the individual variable contribution vector $c$ is given by Eq. (3.21) and $B_k$ is a blocking strategy matrix with element $b_{ij}$ taking value 1 when variable $j$ is in block $i$ and value 0 otherwise.

As a final note, we need to emphasize that the contributors identified by the model are only statistically significant to the degradation. Engineering analysis is essential to further validate that the physical properties that measured by those contributors are indeed the root cause of the problem.

### 3.3 Simulation Study for Single Process Operation

In this section, we use simulated challenge data to evaluate the performance of the proposed PLS modeling strategy. Section 3.3.1 defines the variables used in the simulation, and section 3.3.2 presents results and discussions.

#### 3.3.1 Definition of Input and Variable Blocks

We first start with a group of different signals that may or may not be useful to indicate a degradation process. Fig. 3.2 shows 8 waveforms, among which the
linear, logistic and exponential functions (i.e. linear, logistic, exp1, and exp2) can be considered as indicative variables for a degradation process. Relatively speaking, the

linear functions are the best degradation indicators in those signal, while logistic curve can imply some condition difference in the middle portion, and the exponential curves can only distinguish changes at the beginning or the end of the signal. On the other hand, the const, sine and wave signals are not representative signals for condition monitoring. The purpose of defining waveforms in such a way is to demonstrate the capability of the PLS strategy in comparing signal’s usefulness for predictability evaluation and selecting the optimal one for RUL estimation.

To define input variables for the simulation, we first add white noise to the waveforms defined in Figure. 3.2. Four different noise levels, which defined by different fractions of the range of the signal, are added to the waveforms. In addition, for illustration purpose, 3 different constants are added to the noisy signal to put the input variables on different scales. Thus the total number of input variables is the
A combination of waveform types, noise levels and constant levels, which in this simulation is $8 \times 4 \times 3 = 96$. Figure 3.3 shows the input variables defined under this scheme.

![Figure 3.3: Input Variables for PLS Simulation](image)

Given the above definition, the input variables can be grouped into a hierarchy of blocks, thus enabling the evaluation of the contributions of those blocks to the final regression result. For example, by adding the normalized weights of the variables from each waveform type for all noise and constant levels, one is able to evaluate the effect of a given waveform type on the results. Similar block contributions can be calculated for each noise level, each constant level, or any combination of waveform type, noise and constant levels as well. This strategy facilitates a diagnostic capability by enabling the user to drill down the variable hierarchy and locate the critical contributors to
3.3.2 Result and Discussion

3.3.2.1 General PLS Modeling Performance

We first define a group of 50 target RULs based on Eq. (3.11), with the free parameters $a$ evenly distributed in the interval of $[-7, 3]$. With the input variables defined in Section 3.3.1, we build a PLS model and choose the optimal RUL based on 10-fold cross validation. Fig. 3.4 shows the target group and optimal RUL (highlighted in red) chosen by the algorithm. Notice that the optimal RUL is approximately linear, which indicates there are indicators from input variables that can be used to represent a degradation process. Furthermore, if it is in a production environment, the optimal RUL is able to tell the condition change at least 1000 units before failure, and if 1000 gives enough lead time for maintenance scheduling, it would be a predictable failure.

Figure 3.4: Target Group and Optimal RUL
Fig. 3.5 shows the variable contributions to the regression, by a close look, one can see the major contributions are from the variables generated based on the linear function. In addition, we can see that variables generated by logistic function also have large contribution. These observations are consistent with our simulation setup in that both linear and logistic functions are the best among the functions to indicate a degradation process.

![Variable Individual Contribution for PLS Regression](image)

**Figure 3.5: Variable Individual Contribution for PLS Regression**

Obviously as the number of variables increases, it is difficult to look at the contribution chart of individual variable to find out what causes the degradation. Based on the variable blocking strategy described in section 3.3.1, we can group the variables according to function type, noise level and constant level to assist us evaluate the variable effect on the RUL. Fig. 3.6 shows the contribution from block variables.
From the function type standpoint, linear and logistic functions have dominant influence on the target RUL. Meanwhile, the white noise, wave1 and wave2 functions have very little effect on the regression. Due to the deterministic linear trend in the sine wave, it has some contribution to the RUL regression, but the effect is largely disguised by the periodic property of the signal. This observation prompts the fact that although the output target RUL are defined as monotonic function, the variables that selected by the model as useful for degradation representation are not necessarily monotonic. In reality the signal shown by the sine signal might be seen in some mechanical component failure with period movement. A more indicative feature from this particular variable would be the deterministic trend value of the signal.

From the noise-level perspective, the more noisy the variable, the lower the contribution for the regression. This is intuitive as the data is more contaminated, it is less adequate to be a good condition indicator. Finally, since the absolute value of each input variable is irrelevant to the model as it standardizes them to eliminate
numerical effects, the effects of constants that we add to the signal should be the same and are not relevant to the regression. This is evident in Figure. 3.6 as they all have equal contributions.

This drill down mechanism enables us to quickly evaluate all variables and effectively identify degradation contributors from the input. In addition to blocking the variables with a single criterion, one can also group the variable with multiple criteria if necessary. For example if the variables are grouped by function type and noise level, then a block will exist for each combination of function type and noise level, which will result in $4 \times 8 = 32$ variable blocks. The lower hierarchy blocking strategy is particularly useful when the block on the higher level contains too many variables to evaluate. Thus, a further segregation will make it easier to evaluate variable contribution in subgroups.

Moreover, one can evaluate the loading scores, $T$, on the principal components. These scores give us an idea about the variations in input space that are captured by the model for degradation monitoring. Fig. 3.7 shows the loading scores chosen by the model. In this case two principal components are chosen. The score on first component demonstrates a near monotonic decreasing trend, which can be attributed to the major contributors. However, the score on second component shows a periodic pattern, that resembles the variations in the sine curve as well as the wave curves. Although the increasing magnitude coincides with the degradation, but this feature is not explicitly usable by the model for RUL estimation.

Finally, Fig. 3.8 gives the RUL estimation of the training data. In general, it coincides with the degradation process while small fluctuations exist. This observation is consistent with the scores which also possess small noise. The results in Fig. 3.7 and 3.8 imply that although PLS model select regressors on the direction of maximum correlation, and while the majority of noise and irrelevant variables are suppressed in the model, noise contamination still exists in the regression results. This is due
to the variety of input signals as well as various target RUL curves that the model needs to handle. The individual contribution (Fig. 3.5) shows that contributions from several variables are significantly small compared with others. And those functions that produce these variables are indeed incapable of indicating degradation.

### 3.3.2.2 PLS Modeling Performance with Variable Selection

In order to further improve the estimation accuracy and reduce the number of irrelevant variables from the model, we apply a variable elimination process to remove those variables with insignificant contributions from the model. In this process, variables whose contributions are below a certain threshold limit will be removed from the model, until all remaining variables have weights that are larger than the threshold.

We use the same input variables defined in Fig. 3.3 for the test. Instead of using all 1000 samples for training, only the last 200 points are used to fit the model, which
are highlighted in red in Fig. 3.9. In the test we specify $t = 0.3$ as the regression weights threshold for variable elimination, and the rest of testing configurations are the same as previous. Fig. 3.10 shows the block contribution results. It can be seen that only variables from exp1, linear and wave2 are selected for the regression while the rest of them are not included in the model. It is obvious from the training data that these three functions have good trend and are expected to be good degradation indicators. On the other hand, logistic and exp2 functions become constant and hence are not selected by the algorithm. Additionally, wave1 and sine functions both have up-and-down dynamic trends, which are not helpful for degradation indication. From the noise level perspective, variables with higher noise are further suppressed in the model compare with Fig. 3.6.

Fig. 3.11 gives the loading scores from the PLS model, which selects only the first principal component. The score shows a monotonic trend that are largely contributed from the training portion of exp1, linear and wave2 functions. The variation
Figure 3.9: Simulation Input Function Types and Training Samples (In red)

Figure 3.10: PLS Block Contribution with Variable Selection Strategy
that captured by the model is optimal to describe a degradation process, and those irrelevant or insignificant components are not considered.

![Loading Scores](image)

Figure 3.11: PLS Loading Scores with Variable Selection Strategy

Finally Fig. 3.12 gives the RUL estimation for all input samples. As the model is built upon the last 200 samples, the first 800 points are used for validation. Although the RUL monotonically decreases for the last 200 sample, the decreasing trend is not consistent compare with the first 800 points. This is understandable as our testing input do not have consistent patterns that can be associated with degradation. For example, the wave2 function is actually periodic and it cannot be counted as degradation indicator considering its global pattern. Similarly for the exp1 function, it largely stays at constant level for the first 500 points. However, the RUL, in general, still keeps decreasing during first 800 points, this is due to the fact that the linear function, as one of the major contributors in the model, has consistent pattern to express degradation over the whole process.
3.3.2.3 PLS Modeling Performance with Random Noise Variables

In this section, we address the special case where all the input variables are random noise. And this scenario can be generalized to the case where no useful variation exists in the data for degradation assessment. As the target for PLS regression is defined with a group of useful RUL’s that can indicate a degradation process, it is realistic to ask what if the input data does not have good variation at all? Our deduction is that the data will fit either RUL curve on the boundary of the group as they are the best fit for noise.

We choose the variables generated by the constant (first column in Fig. 3.3) as input of PLS model. Fig. 3.13 shows the optimal RUL selected by the model. As we expected the first from right is chosen, and the reason is that this curve has the biggest portion of itself stays at constant 1, which fits the noise best. Although the last part where RUL drops dramatically from 1 to 0 does not really reflect the real
variation within the data, the error induced by fitting it is smaller compared with other curves. Further checking the loading score in Fig. 3.14, we can confirm that there is no variation from the data that can be used to define degradation.

![Target Group and Optimal RUL](image)

**Figure 3.13: PLS Target Group and Optimal RUL for Random Noise Input**

Based on this fact, one may argue why not define a RUL, say a random RUL, in the target group that can reflect random noise input. The reason we do not define it is that as the signals usually contain random noise input, and quite often the noisy variables are the majority in the input. So, if a random RUL exists in the target group, the algorithm can always find the principal direction that bears the noise which has even smaller fitting error than that of useful signals. Thus, the model will perform regression on the noise rather than useful information.

Fig. 3.15 shows the variable block contribution for a PLS model when we add a random noise RUL in the target group. Variables from noise contaminated constant function are dominant in the model, and some other variables that are unhelpful for degradation evaluation, such as sine, wave1 and wave2, also have heavy influence.
Figure 3.14: PLS Loading Scores for Random Noise Input

Figure 3.15: Variable Block Contribution with Noise Target Presence
on the result. On the other hand, useful variables like linear, logistic functions are suppressed in the model. Moreover, from a noise level standpoint, variables with larger noise have advantages over those with small noise. These are all conclusive evidence that the model is leaning to fit the noise rather than useful information.

In summary, the simulation results validate the performance and effectiveness of proposed PLS modeling strategy. In general, the GRUL target group describes the variation that we would like to see from the input variables that are useful for degradation assessment. And with PLS mechanism, the variables that persist the close correlation with that variation will be identified. Physically, they are the root cause of the degradation.

3.4 Characteristics and Requirements of Multiprocess Prognostics

Process diversity in semiconductor manufacturing can be attributed to both product type and technology node. Due to manufacturing cost and capacity limitations in a semiconductor fabrication facility, different types of products that require the same kind of process during the manufacturing period are inevitably being processed on the same piece of equipment given that the tool is available. This multi-process/product manufacturing environment has imposed challenges for both product quality assurance and equipment health monitoring.

From a process control standpoint, the goal is to deliver different specifications while keeping each process as stable as possible. Since the in-situ measurements during the process are difficult to obtain, run-to-run controllers evaluate the metrology data taken after the current process and make necessary modifications on the recipe to compensate any variations that might incur in the next run. Due to the fact that a wafer might have gone through multiple tools with multiple processes before the
current operation, there are various sources of deterministic variation that need to be removed from the control signal in order to correctly estimate the state, and hence adjust the inputs. In general, there are two ways of approaching this problem. One is to use the historical records, known as control threads, that come from the similar operation context with the current run, to approximate the possible variation in current run [7]. The other approach is to build a global model to consider all process contexts simultaneously and hence realize unified control of the total system [69].

From an equipment monitoring perspective, on the other hand, we focus on the impact on the equipment that is caused by running various processes. As the set-points of the control variables change dramatically for different recipes, the operating condition on the tool also changes, therefore the degradation rate of the tool is not expected to be constant under different operation. Consider at any given time, the tool condition is affected by all previous operations and therefore is the effect of the cumulative impact of the series of distinctive processes. The variation that is brought by different process is two-fold. First, as control variables have their own setpoints for each recipe, the scales of the variables are not the same. Statistically, the data would have a multi-modal distribution in the high dimension feature space. Secondly, as the step definition may vary depending on recipes, the feature variables may also change from process to process. For example, if two deposition recipes have different number of major steps, the dimensions of feature variables are not the same for these processes, therefore it is difficult to use a single model to evaluate the degradation indicated by different sets of variables.

Another challenge for equipment condition monitoring in multiprocess environment is that any prognostics information needs to be context-specific, meaning that at any given time the RUL estimation for a given component should specify under which operation condition the prediction is made. As the impact on the tool from each process is not equivalent, the scope for which a tool is able to operate before a
replacement is needed is not the same for different processes. Furthermore, from a scheduling standpoint, it is essential to know that different processes have their own scope for future operation such that the production plan can be optimized. Thus it is necessary to have a unique evaluation for each process for their impact on the tool, based on limited sample data that are obtained during the production.

Solutions reported in literature for multiprocess equipment condition monitoring can be largely divided into two types. The first type is to build a global model that takes into account all the variations from all possible operational modes. Due to the high nonlinearity of the problem by including all operation modes, various data preprocessing techniques are proposed to first normalize the feature variables with respect to each process so that the input would become independent to the process, and hopefully will reduce or even remove the nonlinearity from the input. Examples of such treatment can be found in [16, 41, 44, 71]. The advantage of using normalized data is that it enables the model to have a consistent evaluation of the impact on the tool from different kinds of operations. However, the problem of this approach is that as it transforms the input data to make them independent from the recipe it also disguises the impact of different processes. The prediction based on the normalized data is for some “normalized” process rather than for any specific operation condition. Therefore the prognostic information is not context-specific and it is difficult for decision making as to which process has better scope on the tool.

The other type of solution employs building separate models for individual processes. Under this philosophy, models are built solely based on data from each individual process. Although these methods require less data preparation, challenges stem from the area where the model needs to evaluate the overall equipment condition by considering the effects from all other processes. Liu [55] has proposed a Hidden Markov Model (HMM)-based approach to tackle this issue. Under this scheme, equipment condition is discretized into several status and is treated as Markovian states
in the model, while feature variables are considered as observations. As separate
HMMs are built for each process, the method uses the probability distribution from
the previous model, which is evaluated for a different process, as the initial condition
of the current model. In this way, the model is able to incorporate the degradation
status that resulted from previous operations. However, the risk of doing so is that
as the models are in chain action, any errors that occur at any particular stage will
be propagated through downstream.

Furthermore, the information that a model has regarding the degradation process
is limited due to the presence of each individual process on the tool. For a process that
runs more frequently on the tool, the model has higher chance in obtaining sample
data from various periods of the degradation. On the other hand, with data from the
process that only sporadically appears on the tool, it is more difficult for the model
to capture the overall degradation. Finally, similar with the problem that suffered
by global modeling approach mentioned above, as an individual model only focuses
on the degradation caused by a single process, it is unable to provide prognostics for
other process at any given time.

In summary, the challenges for multiprocess equipment prognostics are both degra-
dation assessment with limited observability of each process, as well as RUL estima-
tion for a different processes at any given time. The difficulty underneath is the lack
of ability to evaluate the equivalent impact on the tool between different process. For
example, during any time period, if one is able to know what are the tool degradation
patterns for a different process, prediction can be made with respect to each process.
In this work, we extend the PLS modeling strategy to tackle multiprocess problems.
Section 3.5 will detail the assumption as well as modeling procedures of the PLS
algorithm. Section 3.6 provides simulation studies to validate the assumption and
algorithm performance.
3.5 Prognostics Evaluation for Multiple Process Operation

We first start with making some realistic assumptions to simplify the problem such that the PLS strategy is still feasible. The constraints imposed by the assumptions are then identified and quantified as well. Finally, we will evaluate those assumptions against the semiconductor environment to validate their practicalities.

3.5.1 Practical Consideration and Modeling Assumption

For a single process operation, the equipment degradation process can be expressed by the variations in the data given that the failure mode is predictable. The degradation pattern is indicated by the shape of the optimal RUL that is chosen from the target group. Given two different recipes, if any one of them is running continuously without interruption on a tool, the number of wafers that the tool is able to process and the shape of the optimal RUL are expected to be different from each other as the processes have distinctive impacts on the tool. If these two processes are running on the same tool alternately, the impact on the tool condition would be the combination of that from both processes. Theoretically speaking, the total number of wafers that the tool can process would be in the interval of that from the two processes.

Assuming there are two recipes \((A & B)\) and we know their optimal RULs on a tool under single processing, let us consider a simple recipe switch scenario. Suppose recipe \(A\) first starts running on the tool for some wafers, during which time the impact on the tool would follow the pattern indicated by the beginning part of its optimal RUL. After that, recipe \(B\) starts running for some wafers, and a different influence incurs on the tool during this time frame. But the impact pattern that recipe \(B\) has on the tool would not start from the beginning of its optimal RUL because the tool condition already degrades to some point after running recipe \(A\). Hypothetically, if we can measure the impact from the previous operation of recipe \(A\) by the equivalent impact of recipe \(B\), we would know exactly the impact pattern that recipe \(B\) is going
to follow on its optimal RUL. Similarly, if recipe $A$ resumes running after recipe $B$, the impact pattern would pick up at the end of the summation of its first run and the equivalent run of recipe $B$.

In general, when an arbitrary number of recipes are running on a tool, if we can measure the equivalent impact of the currently running process by its counterpart from all other recipes on the tool, we are able to continuously evaluate the tool condition degradation under the multiprocess operation. Furthermore, when a prediction is given for any running process on the tool, we are also able to obtain the equivalent prediction for other processes based on their equivalent impact pattern.

Practically, the problem of the above discussion is that the single process optimal GRUL of an individual process is unknown. Due to the diversity of processes as well as the constraints of the production environment, it is impossible to conduct test runs to obtain the single-process RUL for each recipe. Instead we can only approximately estimate it based on the sample data that are collected from the multiple process production. Considering the optimal GRUL as an indicator that shows the way a particular process is affecting the tool, the instantaneous impact should be solely determined by the process itself and is irrelevant to other processes that share the tool. The influence of previous recipes is that they would cause the tool to degrade to some point after the process and change the initial position of degradation for the current recipe. Intuitively, as multiple processes are running on the tool, if a recipe runs frequently enough, the data that are collected during the process is expected to be representative to reflect the impact on the tool.

Therefore, we make the following assumption, and the validity will be evaluated by simulation as well as semiconductor case studies in the following sections.

**Assumption III.1.** *The instantaneous impact of an individual process on the tool is independent from other processes and is solely determined by the ongoing process.*

Based on this assumption, the impact pattern of an individual process on the
tool degradation can be approximated from the sample data that are observed in different stages of the degradation, given that the process spreads widely enough over the whole degradation. Furthermore, the equivalent impact from other processes in the time frame of the currently running process can be approximated by the impact pattern from the same time frame of other processes. The details of the modeling procedure are described in the following section.

3.5.2 Modeling Procedure

Figure 3.16: Example of Recipe Switch on the Tool

Based on assumption III.1 we extend the PLS modeling strategy described in section 3.2 into the following procedure for multiprocess tool condition modeling:

1. Fig. 3.16 shows an example of multiple recipes alternately running on a tool. We first define a group of RULs for all the recipes based on Eq. (3.11).

2. In order to estimate the impact of each individual process on the tool degrada-
tion, we divide the target GRUL into groups based on recipe index. As shown in Fig. 3.16, each recipe has the corresponding slices of the target RUL group.

3. Build individual PLS models for each process with its own target GRUL specified in step 2, and identify an optimal RUL based on Eq. (3.12). As can be seen in Fig. 3.18, the bold portions of the selected optimal RUL indicate the actual target values that have been fitted to the inputs, while the remaining parts in the slice are the equivalent impact of currently running process that are measured by other processes.

4. Evaluate the predictability of each recipe by comparing the number of wafers that are distinguishable before failure and the requirement for maintenance lead time.

5. For a predicted RUL of a running recipe, find the equivalent RUL for other recipes based on the equivalent RUL curves. For instance, if recipe 3 is currently running on the tool and an estimated RUL 0.2 is given by the model, the wafer index that corresponds to 0.2 on the green curve (Fig. 3.18) intercepting on the blue and red lines will yield the equivalent RUL of recipe 1 and 2, which are approximately 0.5 and 0.85, respectively. The relative position of optimal RULs indicates impact magnitude on the tool. In this example, recipe 3 has the heaviest impact on the tool and hence its RUL decreases faster than that of the other two.

The primary motivation of building individual model is that for semiconductor processes, step definition often varies a lot between recipes. Therefore it is not unusual that the feature variables are different for recipes. An individual model only deals with variables from a single process, which will remain unchanged. Secondly, individual models yield optimal RUL for each process, and based on the assumption we made, those RUL over the component life cycle span can be considered as their equivalent
Figure 3.17: Example of Target RUL Group for Individual Recipes

Figure 3.18: Example of Optimal Equivalent RUL Individual Recipes
single process RUL. Then, these equivalent RUL curves from the switching process can be used to estimate the RUL for other processes when a RUL prediction is made for the current process. Finally, data normalization is avoided by building individual models, and the variation the model describes is more consistent than if the inputs are combined.

### 3.5.3 Process Switch Index

In this section, we quantitatively evaluate the spread of an individual process over the whole degradation process. Based on assumption III.1, the individual process impact on the tool can be estimated from the the partially observed samples that are collected from the process, given that the samples are representative of the model. The question one would ask here is how widely the process has to spread, in order for the model to have a good approximation? Intuitively, the bigger the gap is between two consecutive runs of the same process, the more information we lose from that period of equipment degradation. On the other hand, the higher frequency a process has in the overall degradation process, the more representative the data are.

Based on these considerations, we first define a *gap* as the interrupted period between two consecutive runs of an identical process, and then a Process Switch Index (PSI) is defined as the normalized mean gap width to indicate the spread of the individual process, namely:

\[
PSI = \frac{\sum_{i=1}^{N} n_i / N}{M}
\]  

(3.23)

in which \(N\) is the number of gaps of a individual recipe during the whole degradation process, \(n_i\) is the width of the gap measured by the number of wafers in the time frame, and \(M\) is the total number of wafers in the training data.

It can be seen that the value of PSI is between 0 and 1, with 0 indicates no
interruption of a single process operation and 1 corresponds to case that a process has never appeared on the tool. For regular mixed production, if each gap in the process is small, then the nominator of Eq. (3.23) is small, which indicates less interruption and better continuity of the process. Therefore, the smaller the PSI, the wider it spreads the overall degradation and the better continuity it has, which also indicates a closer resemblance to the single-process operation with a zero-valued PSI.

Fig. 3.19 shows a simulated recipe switch process on a semiconductor tool. Table 3.1 shows the statistics and PSI results of the recipes. There are four recipes running alternately and they are randomly chosen to run based on their repeating

<table>
<thead>
<tr>
<th>Recipe index</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeating probability</td>
<td>10%</td>
<td>20%</td>
<td>30%</td>
<td>40%</td>
</tr>
<tr>
<td>Total lots</td>
<td>11</td>
<td>22</td>
<td>26</td>
<td>41</td>
</tr>
<tr>
<td>PSI</td>
<td>0.0804</td>
<td>0.0388</td>
<td>0.0390</td>
<td>0.0221</td>
</tr>
</tbody>
</table>

Table 3.1: Statistics for Recipe Switch Simulation

Figure 3.19: Simulated Example of Semiconductor Recipe Switch in a Chamber
probability of appearance. The recipes are running on a lot basis, with each lot containing about 20 wafers. From the recipe index plot (Fig. 3.19) we can see that recipe #1 only has sporadic appearance during the whole process, while the others have better coverage of the time span. This is quantified by the PSI value according to Eq. (3.23), as recipe #1 has the biggest value, while the rest are significantly smaller. The physical meaning of PSI can be interpreted as the average percentage of training data of the current process that was interrupted before it resumes. For example, recipe #4 has a PSI of 0.0221, which implies that after the process stops, it will resume, on average, after about 2.2% of the total training wafers with other type of processes, which in this case is about 44 wafers. Considering actual semiconductor manufacturing, the interruption is only about one to two lots, which is a high frequency runner. Furthermore, in a multiprocess manufacturing environment, it is expected that processes switch back and forth to accommodate the variety of production requests, as long as any individual appears at various stages of the tool degradation process, it has a better chance to reflect the impact on the tool.

One way to mitigate the influence of process switch on the degradation assessment is to combine training data from multiple incidents of identical failures. Since semiconductor scheduling usually takes a heuristic approach, and wafers will be processed once the tool becomes available, the appearance of a particular recipe would randomly occupies different stages of the degradation process. Therefore, the addition of multiple training runs will reduce the gaps in the training data, and thus enhance the visibility of the recipe during the degradation process.

Furthermore, the combination of multiple training sources will also reduce the chance of identifying some coincidental patterns from the data while neglecting the true root cause. As different components are all in a degradation process when the tool is running, various variables may indicate variations in the training data. Although the PLS algorithm chooses the most consistent variation from the input, it is still
possible that the identified variation is merely a statistical coincidence. By combining training data from multiple sources, the variation corresponding to root cause should remain consistent in those different data sets while the variables with inconsistent variations would demonstrate different patterns, and hence reduce their correlation with the degradation.

3.6 Simulation Study for Multiple Process Operation

So far, we have described the modeling strategy for multiprocess degradation. For the assumption we made in III.1, although it is based upon practical considerations, its validity needs to be quantitatively evaluated. In this section we will conduct simulations to quantify the influence of process switch on PLS modeling and the effect of combining training data from different failures on root-cause identification.

3.6.1 Model Trainability with Partially Observed Data Set

We define the trainability ($\tau$) of a process-specific model as the similarity between the optimal RUL obtained from the partially observed training sections and the optimal RUL that would have been obtained if the process occupied the entire time frame. In this work, the similarity is quantified by the normalized Root Mean Square (RMS) of the difference between the two optimal RULs,

$$\tau = 1 - \frac{d_i}{d_{\text{max}}},$$

$$d_i = \sqrt{\frac{\sum_{i=1}^{M} (y_i - \hat{y}_i)^2}{M}},$$

$$d_{\text{max}} = \max_{i \in [1 \rightarrow K]} (d_i).$$

(3.24)

Here, $\hat{y}_i$ is the element of the optimal RUL $\hat{y}$ obtained from partially observed data, while $y_i$ is the element of the optimal RUL $y$ identified by a complete training set.
Equation (3.24) implies that the trainability is in the interval of \([0, 1]\), where 1 indicates an identical optimal RUL from partially observed data while 0 indicates the biggest discrepancy in the RUL group.

If the two RULs are close enough, we can consider that the model is trainable based on the available wafer samples. As discussed before, the spread of a particular process can be described by its PSI defined previously. Therefore, the trainability of a model is influenced by the PSI of a process. Here, we evaluate this relationship by simulation. In Section 3.3.1, we have created a series of input variables, and according to Fig. 3.4, an optimal RUL is identified as the best indicator for the variations in the data. Here, we keep using the same challenge data as input and randomly segregate it into groups with different PSI.

Fig. 3.20 shows a random segregation of a process of about 5000 wafers, where the process is divided into 20 recipes with different PSI values. In the simulation, the optimal RUL \((y)\) is first obtained with data from all wafers. Then data corresponding to each recipe index are used to fit a PLS model and get the optimal RUL \((\hat{y})\). We train the model 20 times for each data partition and identify the mean and standard deviation of its trainability. Fig. 3.21 shows the relationship between PSI and trainability. The trainability is plotted as the mean with \(\pm 1\) standard deviation on Y-axis. We can see that, in general, as PSI increases, the model trainability drops, and the uncertainty of the training results increases. Specifically, recipes 11 to 20 are frequent runners, and their appearance at different stages of the degradation process ensure a representative population for model building, and hence guarantee good trainability.

On the other hand, recipes 1 to 10 only have sporadic appearances during the process, which result in larger discrepancies in optimal RUL. For example, recipe 5 has a PSI of 0.25, which indicates the average interruption in this recipe is 25% of the total training period. In other words, there are at most three runs during the
Figure 3.20: Simulated Example of Process Switch

Figure 3.21: Relationship Between Process Switch and Model Trainability
whole degradation. Therefore it is no surprise that the data from this recipe yields the poorest RUL estimation. Special cases that needs to be pointed out are recipes 7 and 8, both of which have similar PSI but different trainabilities. A close look at Fig. 3.20 reveals that for recipe 8, the process only runs until about wafer #3000 and never repeats again. For recipe 7, although it has big gaps during the process, it does have one more appearance towards the end of the process. Considering the definition of the target GRUL group, without any training data for a long period of time, the segregated target group may have some RULs that do not indicate variation at all. As we showed in Section 3.3.2.3, when the target RUL takes the form of random noise, the algorithm can always find variables from the input that fit the noisy target better than the others. In the case of recipe 8, the majority of data were fitted better on the noise target than the rest on the small portion with degradation variation, which is the cause of poor trainability.

3.6.2 Elimination of Variables with Inconsistent Patterns

As mentioned earlier, it is expected that the PLS algorithm would select the variables with consistent patterns from multiple training sources to indicate the degradation, while suppressing those with contradictive variations. We use the function type defined in Fig. 3.2 to produce challenge data to validate this. From Fig. 3.6, we see that both linear and logistic functions have dominant variations on the regression, while “exp1” has a relatively smaller effect. We define two training sets with inconsistent patterns for the linear and logistic function while keeping the rest intact as shown in Fig. 3.22. We can see that in the first training set (Fig. 3.22(a)), the linear function becomes random noise to make it inconsistent with that from the second training set. Similarly, the logistic function in the second training set (Fig. 3.22(b)) is made to be noisy to contrast with its counterpart in first training set. Assuming each training set has 1000 points, 4 different magnitudes of noise and 3 different constant levels are
used to generate a total 96 input variables from the 8 function types.

Fig.3.23 shows the variable blocking contributions using the combined training data set. It can be seen that contributions from both the linear and logistic functions are eliminated from the regression due to their lack of repeatability in the training sets, and the dominant variations are now from the “exp1” function type. Also notice that in terms of the noise magnitude, the variables with higher contamination have smaller contributions. And for the constant level, the difference in constant levels do not affect the regression. These observations are consistent with previous examples.

To summarize, the simulation examples have validated our expectations that, when widely spread, the effect from a particular process can be approximated by modeling partially observed data. The relationship between the training data spread quantified by the process switch index (PSI), and the modeling accuracy defined as trainability, was also evaluated. Finally, it was validated that by using multiple training data sets, possible coincidental patterns can be eliminated from consideration, thus improving the diagnosis capability for root cause identification.
3.7 Summary

In this chapter, we propose a method of equipment fault prediction for multi-process manufacturing systems based on PLS regression. By applying multivariate analysis, the PLS algorithm identifies the best correlation between feature variables that are extracted from input data and a group of predefined RUL curves, which represent various degradation patterns. Then, an optimal RUL target is chosen from the group with the minimum cross validation error. Based on the shape of this RUL, we are able to determine the predictability of the problem, which could be used to define the maintenance strategy for the corresponding failure. Furthermore, by evaluating the variable blocking contributions of the regression model, one can identify variables indicative of the equipment condition and diagnose if the degradation is determined by the physical property that is measured by those sensors. Finally, RUL prediction is provided before replacement is needed.
In addition, we analyzed the scenario where multiple processes are operated on a tool. The challenges and needs for predictability evaluation and RUL estimation are identified. Based on these analyses, an *independence* assumption is made that the spontaneous effects of a process on the equipment are independent from other processes and are solely determined by the process itself. With this assumption, we extend the PLS modeling procedure for multiprocess problems. By dividing the target RUL group into different sections and building separate models for each context, we can approximate the effect that a process has on the tool given that the disruptions of the particular process is small enough. A process switch index is defined to quantify the interruption and its relationship with model accuracy, defined as trainability, is obtained from simulation.
CHAPTER IV

Case Studies: Prognostic Evaluation for Semiconductor Tools

This chapter provides case studies with semiconductor tools from a wafer fab, and evaluates the effectiveness of the proposed solution. Section 4.1 gives three prognostics evaluation examples for a single process with different predictability scenarios, and section 4.2 provides two use cases for multiple process system. Finally, Section 4.3 will conclude the chapter with discussions of the results and possible future work for improvement.

4.1 Single Process Cases

In this section, three case studies are provided to demonstrate the capabilities of the proposed solution for prognostic evaluation of single process operation.

4.1.1 Feature Extraction

At a given semiconductor manufacturing operation, a recipe is typically performed on the wafer in which various physical or chemical procedures are applied in a series of steps. During the process, numerous sensors collect signals off of the tool at some sampling rate. Thus, the raw data from each wafer can be stored in a 2-dimensional
matrix \( \mathbf{R} \in \mathbb{R}^{t \times n} \), with \( n \) being the number of sensors and \( t \) the number of sampled points, which is the product of sampling rate and process time. The feature variables \( f_m \) we extract in this section are from \( s \) summary statistics calculated at \( p \) critical steps for all \( n \) sensors, namely:

\[
\begin{align*}
    f_m &\rightarrow \text{Sensor}^{(i)}, \text{Step}^{(j)}, \text{Statistic}^{(k)} \\
    \forall i \in [1 \rightarrow n], j \in [1 \rightarrow p], k \in [1 \rightarrow s].
\end{align*}
\] (4.1)

Thus there are totally \( m = n \times p \times s \) variables in the vector input of the PLS regression model.

4.1.2 Case 1 - A Predictable Example

In this example, an electrostatic chuck (ESC) failure of a deposition tool is considered. The ESC is an important component in various process chambers. It holds the wafer by electrical forces and ensures that it maintains a static position in the chamber as the process proceeds. Therefore any performance degradation of the ESC would result in uneven reaction across the wafer surface and hence cause undesired electrical properties on the device, which will eventually lead to production loss. In this example, we demonstrate application of the proposed strategy for the RUL estimation of the ESC on an etch tool.

Signals are collected from 48 sensors on the tool. Six critical steps are selected from a particular recipe, from which the mean (\( \mu \)), standard deviation (\( \sigma \)) and processing time (\( t \)) are calculated. Therefore, \( 48 \times 6 \times 3 = 864 \) feature variables are available for each wafer according to Eq. (4.1). Training samples are collected from over 1100 wafers before the failure, and are used to build the PLS model. Fifty target RUL curves are defined for the group, with the coefficient \( a \) evenly distributed along the interval of \([-4, 4]\).
Fig. 4.1 shows the loading scores $\mathbf{T}$ given by PLS modeling without variable selection, for which 3 principal directions are chosen. Two trends are seen in $t_1$ and $t_2$, which are indicative signals for degradation. But $t_3$ fails to show any evidence of the degradation, especially when it gets close to the failure. Instead, it remains at the same level without significant change. In addition, all three scores have a significant number of outliers, which could affect model accuracy.

Fig. 4.2 shows the block contributions from variable blocks that are grouped by sensor (top), step (middle) and statistic (bottom). From among the sensors, it can be seen that while some sensors have heavy weights on the regression, a lot of them only have little contribution. Concerning the steps, it is shown that all steps have comparable contributions to the model. The statistic contributions show that variables from mean dominate the regression. Based on observations from Fig. 4.2, one can check feature variables with large contributions and analyze if they are the cause of the degradation.
As seen above, although the PLS algorithm performs the regression on the principal direction in the latent space, the model can still be contaminated by noise when a substantial amount of feature variables do not have enough variation to indicate degradation. From the score on the third principal direction $t_3$, it is evident that the model starts fitting the noise. Based on these observations, we further adopt a variable selection strategy in which only variables with significant contribution are kept and those with small regression weights are not included in the model.

Fig. 4.3 shows the loading score of the same application with selected feature variables that have high contributions to the regression. Only one principal direction is selected by the model, and the trend of the score is smoother than in Fig. 4.1. In addition, the training error is reduced by 20% after eliminating the low contributors. Fig. 4.4 shows the blocking contributions for the reduced selection of feature variables.

One can see that only mean values of the critical steps are chosen, while processing time and standard deviation in those steps are not indicative for the degradation and
Figure 4.3: PLS Loading Score with Selected High Contribution Variable

Figure 4.4: Variable Blocking Contribution with Selected High Contribution Variable
hence are not considered by the model. Meanwhile variables from many sensors are also eliminated, which implies that the physical properties that are measured by those sensors are irrelevant to the component condition.

It is worth noting that by comparing Fig. 4.2 and 4.4, from a sensor perspective, while most variables with small contribution are eliminated, some variables (i.e. 18, 19) that have relatively large contribution are also removed from the model. Fig. 4.5 shows the mean values of all critical steps from sensor 19, in which the signals are all increasing during the first 600 samples, while keeping almost steady after a drop. The effect of these signals as large contribution variables in the model can be seen in the first loading score $t_1$ in Fig. 4.1, where the score shifts before the sample 600. The patterns shown in these signals are not consistent within the training cycle and are not good indicators for the condition change. The reason for the high contribution of sensor 19 in Fig. 4.2 is that although each individual variable computed from this sensor has relatively low contribution, after combining them the overall

![Figure 4.5: Mean Values from Sensor 19 for Model Building](image-url)
sensor level contribution yields a big number.

On the other hand, with the variable selection strategy, contributions from all the individual variables are below the threshold and hence those variables are not considered by the model. In contrast, Fig. 4.6 shows the mean values of all critical steps from sensor 2, which are identified as high contributors in both Fig. 4.2 and 4.4. The mean signals from steps 3, 6, 7 all have steady decreasing trends, while values from steps 1, 9, and 11 do not correlate to the degradation. This conclusion is also consistent with the step block contribution in Fig. 4.4.

The RUL prediction by the PLS model with selected variables is shown in Fig. 4.7, in which the failing point is marked by the vertical red line. Data before failure are used to build the model, and the predictions are provided for wafers processed after a new component is installed. Note that the predicted RUL has values that are larger than 1, which indicates that the component condition at that point is better than the best condition in the training cycle. This is observed because the training data
does not cover the entire life-cycle of the component. Based on the RUL prediction, engineers would be able to adjust production in the chamber to maximize the tool utilization.

![Normalized Remaining Useful Life Prediction](image)

Figure 4.7: RUL Prediction by Selected Variables

Finally, the model built in Fig. 4.7 is validated with another Electrostatic Chuck (ESC) failure instance from a different chamber on a different tool. Fig. 4.8 shows the estimated RUL. One could see that before the failure, the condition deterioration is represented by the by the decreasing RUL. And the ESC is actually replaced after the 518th wafer being processed, after which the RUL estimation jumps to a higher level to indicate better condition of the new component.

This example indicates that although numerous data can be collected from a semiconductor processing tool, the majority of signals are used for internal process control within the tool. These sensors are typically maintained at a specified level without much fluctuation. Therefore it is important for equipment engineers to quickly discover the alternative signals, assuming there are any, that are useful for tool condition
monitoring. The proposed method has enabled users to find important features for fault prediction from a large amount of variables by statistical analysis, thus avoiding the potentially time-consuming work of evaluating each feature variable manually.

4.1.3 Case 2 - A Detectable Example

In this example, we demonstrate the usage of PLS modeling for a cryopump failure on a PVD tool. After a wafer process completes, the chamber must be cleaned in order to obtain the desired operation condition for the next wafer. A cryopump is an entrapment pump which evacuates exhaust gases in an adsorbed state and creates a vacuum environment for the following process cycle. Due to the pumping mechanism, the effectiveness of cryopump decreases as its sorbent gets saturated. An under-performing cryopump usually causes exhaust remains in the chamber with a low vacuum condition, which could jeopardize the next wafer. Therefore, the performance of cryopumps is critical to the process and every so often the sorbent needs to be re-
generated in order to maintain its high effectiveness. During the pumping process

![Figure 4.9: Optimal RUL Target of Cryo Pump](image)

between wafers, pressure is recorded by the tool. We extract four statistics; namely mean, min, max and standard deviation, from three steps of the recipe. These are the input variables of the PLS model. Training data are collected before a historical regeneration of the pump. The optimal target RUL obtained using training data from over 1000 wafers is shown in Fig. 4.9. The selected curve has a coefficient of $a = -4$, which does not imply any change for the majority of the training cycle, and the decreasing trend at the end indicates further validation is necessary to ensure that a corresponding variation exists in the inputs.

Fig. 4.10 shows the loading score of input data. Only the first principal direction is selected by the model, which indicates for most of the time, the signal is stable with occasional excursions along the process. Furthermore, the dominant variation exists at the end of the training cycle, in which a shift of the signal is observed at the opposite direction about 100 wafers before the regeneration. Although at this
Figure 4.10: Loading Score of a Detectable Failure Mode

point one can suspect that this might be an indicator of the deteriorating pumping performance, further confirmation is needed from the input signal.

Fig. 4.11 shows the contribution from variable blocks. It can be seen that pressures in steps 1 and 3 have relatively large impact on the pump performance. And from the statistic standpoint, the minimum and mean value of the pressure are major indicators. Further checking input variables in Fig. 4.12 one can see that the pressure levels are much higher in steps 1 and 3 when it gets closer to the regeneration. And the excursions we observed in Fig. 4.10 are actually corresponding to those cycles with relatively lower pressure. Since the objective of the cryo pump is to evacuate gas from the chamber to create a vacuum for next process, the lower pressures are actually better a condition.

On the other hand the higher pressure indicates a decrease in the efficiency of the pump. However, we can see here that the pressure shift only happens about 4 lots before the pump needs a regeneration, which is too short to schedule any maintenance
Figure 4.11: Contribution of the Pressure Variables

Figure 4.12: Summary Statistic Variables of the Pressure
in the near future. Thus, the indicator is only sufficient to provide a detection solution rather than prediction. From a maintenance perspective, since the abnormal pattern only provides a short lead time, it is recommended to have replacement parts be stored so that the pump can have a regeneration once the up-shifting minimum pressure level is detected.

### 4.1.4 Case 3 - An Unpredictable Example

For this case, an investigation is conducted into data collected from a CMP tool, during a time period in which a spin motor failure was known to have occurred. Summary statistic features are extracted from 2 statistics (mean and standard deviation) of 2 critical steps for 5 sensor readings, giving a total of $2 \times 2 \times 5 = 20$ variables.

The optimal target RUL obtained using training data from over 600 wafers is similar to the previous example shown in Fig. 4.13. However, the loading score, shown in Fig. 4.14, implies there is no indication of a process degradation signal. Specifically, at the end of the training cycle, the score does not have any trend pattern that correlates to selected RUL curve. Next, a review of the blocking contribution chart (Fig. 4.15) indicates that variables from sensor 5 have the biggest impact on the model. However, the corresponding feature variables for the sensor (Fig. 4.16) are not indicative of the abrupt deterioration expected based on the RUL curve.

This particular example demonstrated the aforementioned scenario where one of the extreme RUL curves was selected, but no significant variation existed in the input data related to tool degradation. Therefore, one can conclude that this is an unpredictable failure mode based on current feature variables. In order to address this issue, a reevaluation of the feature extraction process should be performed in an effort to collect more sensor signals that may be more useful for prediction.
Figure 4.13: Optimal RUL Target of Spin Motor

Figure 4.14: Loading Score of Unpredictable Failure Mode
Figure 4.15: Variable Blocking Contribution

Figure 4.16: Feature Variables from Sensor 5 for Model Building
4.2 Multiple Process Cases

Now we will illustrate the usage and effectiveness of proposed method with semiconductor case studies for RUL estimation with multiprocess operation.

4.2.1 Case 4 - Electrostatic Chuck on Etch Tool

4.2.1.1 Operation Overview

Fig. 4.17 shows two production recipe switch processes in two etch chambers. At the red marker line, an identical ESC failure happens in both chambers. We can see in Chamber 1, recipes 2 and 5 have the most frequent appearances during the process, recipes 8 and 9 have moderate presence, and recipes 1, 4, 6 and 7 only appear a few times. A special note for recipe 3 is that it does not start running in the chamber until it is very close to the component replacement. Although it runs frequently after the start, it lacks appearance in a significant amount of period in the beginning of the process. For Chamber 2, recipes 3, 5 and 9 are frequent runners while recipe 7 has sporadic appearance. Recipes 4, 6 and 8 have very few in production. Note that for recipe 3, training data are available from record 2 at early and middle age of the degradation, which are a good complement for Chamber 1 and will reduce the PSI significantly after combining two records into a training data set.

To illustrate the difference between those recipes, Fig. 4.18 shows an example of step definition as well as one sensor readings for recipes 5, 8 and 9. We can see that although recipes 5 and 9 share some similarity in sensor readings in terms of waveform, each step has its own magnitude and duration. On the other hand, recipe 8 is much different compared with the other two recipes, which indicates the difference in the operation conditions in the chamber. Table 4.1 summarizes the feature variables and PSI for each recipe. For each critical step in the recipe, we calculate 3 statistics as feature variables. With different key steps, the total number of feature variable is
not the same across different recipes. Based on the evaluation of PSI and model trainability in this example, we build model for those recipes whose PSI is no larger than 0.05, and the training data set are combined from the two failure incidents whenever possible.
4.2.1.2 Results

Let’s first evaluate the effectiveness of combining multiple sources of training data on the identification of degradation root causes. Fig. 4.19 shows the contribution charts blocked by sensor names for different training data sets from recipe 5. By using Chamber 1 alone, sensor 18 has the biggest contribution for the degradation. However, this observation is contradictory to what is indicated by using Chamber 2 alone, in which sensor 18 is completely irrelevant to the model while sensor 11 is a dominant factor. Furthermore, if we combine the training records it implies that neither of the two sensors has any influence on the degradation, instead sensor 2 and 29 are the major contributors.

The explanation of the difference can be found in Fig. 4.20, which shows the mean values of step 6 in those sensors from both training records. One can see that while the feature variable from sensor 18 has a decreasing trend in Chamber 1, it largely does not show any similar pattern in Chamber 2. Similarly for sensor 11, an increasing trend is found from Chamber 2, but no significant pattern is seen from the same sensor in Chamber 1. Thus by combining both training records together, neither of these two sensors are identified as critical parameters for the degradation assessment. On the other hand, feature variables from both sensor 2 and 29 have shown consistent monotonic trend in the training data and thus are identified as major contributors.

The reason of this observation is easy to understand. Since we are simultaneously

Table 4.1: Process Summary for High Frequency Recipes in Fig. 4.17

<table>
<thead>
<tr>
<th>Recipe index</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSI-1</td>
<td>0.014</td>
<td>0.025</td>
<td>0.008</td>
<td>NA</td>
<td>0.022</td>
<td>0.024</td>
</tr>
<tr>
<td>PSI-2</td>
<td>NA</td>
<td>0.016</td>
<td>0.003</td>
<td>0.036</td>
<td>NA</td>
<td>0.008</td>
</tr>
<tr>
<td>Sensor #</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
<td>48</td>
</tr>
<tr>
<td>Key Steps #</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Stats #</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Variable #</td>
<td>864</td>
<td>720</td>
<td>864</td>
<td>576</td>
<td>576</td>
<td>720</td>
</tr>
</tbody>
</table>
Figure 4.19: Sensor Blocking Contribution by Different Training Data Set of Recipe 5

Figure 4.20: Mean Values of Recipe 5, Step 6 from Sensors with Contradictory (11 & 18) and Consistent (2 & 29) Blocking Contribution
evaluating variables from 48 sensors on the tool, all these sensors are measuring different physical properties of different components on the tool. As all the components are degrading over the course of usage, there are patterns to indicate various failure modes on different components. By combining multiple sources of training records, one can eliminate the variables with contradictive contribution from the model and reduce the chance of having incorrect cause-and-effect relations for the failure. But it is no guarantee that the identified contributors are indeed the cause of the failure. Therefore, in addition to finding variables that are statistically coincidental to the failure, it is important to also use engineering judgment and troubleshoot the failure to confirm the root cause based on the indication given by the model.

Next, let’s take a look at the modeling results. Fig. 4.21 shows the optimal target RUL curves that are generated by each individual model. One can see that for all recipes, the optimal RUL is identified as linear or close to linear, which indicates that during the whole training cycle, all the models are able to distinguish the status of the ESC and provide RUL estimation. Based on this observation, engineers should evaluate if the lead time provided by the prediction is long enough for maintenance scheduling. If so, then predictive maintenance can be implemented. Otherwise, one should enlarge the window to include more training wafers in order to provide longer RUL estimation.

The observation in the optimal RUL chart can also be confirmed by evaluating the loading scores on the principal components that are selected by the model. Since the scores are the indication of the dominant variations in the data based on which the model is built, they are in general in line with the shape of the optimal RUL. From Fig. 4.22, only one principal direction is selected for all recipes expect 3. The scores all demonstrate a monotonic trend in the training process, which could be considered as the indication of degradation. Noted that for the scores of recipe 3, 5 and 9, since we are training the model with data from two failure incidents, the scores showed two
For recipe 3, we notice that the trends in the scores are not in the same scale, specifically for $t_2$, the increasing trends from two records do not end at the same level; and for $t_1$, the start level of the trend from Chamber 2 is roughly the same as the ending level of Chamber 2. This implies that the absolute values of the feature variable before each failure incidence are not the same, which suggests that the failure conditions are different for the two incidences. On the other hand, for recipe 5 and 9, after we combine their training records, although the start level of each trend are different, which is due to the fact that they have different numbers of wafer before failure, the ending level of the trends are roughly the same for both recipes.

Furthermore, Fig. 4.23 shows the RUL estimation for each recipe based on the training data in two chambers. In general, all recipes have a decreasing estimation before failure in the training phase. Depending on data availability, some validation data from recipe 3, 5, 8 and 9 are also extracted after the ESC is replaced. We can see that the RUL values are all at a higher level after replacement, which indicates a better condition of the component. In particular, for recipe 2, the lowest RUL is only around 0.2, indicating it stopped running well before failure (Fig. 4.17). For
Figure 4.22: Loading Scores on Selected Principal Component of Different Recipes

recipe 3, as indicated by the loading scores (Fig. 4.22(b)) that the feature variables have different scales for two chambers, which also implies that the baseline failing conditions are not the same from the two ESCs. Consequently the variation captured by the model is the compromise between the two failure incidents. This inconsistency
of training data also cause the inconsistent RUL estimation for the wafer after the ESC is replaced. As one can see in Fig. 4.23(a), the prediction of recipe 3 is much higher than others, while the optimal RUL indicates all recipes have similar impact on the tool.

Finally, the models of recipes 3 and 5 are validated by data from other chambers. Fig. 4.24 shows the RUL estimations. For recipe 3 (Fig. 4.24(a), although a decreasing trend is seen for the RUL predictions, the value drops below 0 several lots before the

Figure 4.23: RUL Estimation of Different Recipes
ESC is replaced, which implies the component is changed at a worse condition than that observed from the training data. On the other hand, for recipe 5 (Fig. 4.24(b)), as the failing baseline is more consistent from the training data (Fig. 4.22(c)), the RUL prediction is at 0 before the ESC replacement. In both recipe, we can see the prediction jumps to a higher level right after replacement, which indicates the better condition for the new components.
4.2.2 Case 5 - Polish Pad Change

Polishing is a mechanical planarization process in semiconductor fabrication. It is usually performed after a deposition process in order to prepare the wafer surface to a specified condition for next process. As the polish pad wears out during usage, the polish rate cannot be guaranteed at the required level. This can cause undesired surface roughness on the wafer, which will affect subsequent processes.

4.2.2.1 Polishing Mechanism and Feature Variables

Polishing is one of the few semiconductor processes that operates outside a chamber. Fig. 4.25 is a schematic diagram of a typical polish tool. The mainframe of the tool consists of three platens for different types of polishing. The polish pad is attached to the platen and rotates at a certain speed during the process. The robotic cross-arms above the platens hold wafers and maintain contact with the polish pad. During the process, slurry flow is provided through the flow systems that attached to each platen. In addition, a scanner is also equipped for each platen to provide \textit{in-situ} thickness measurements of the polish layer. The polish process is controlled by condition parameters, such as platen and wafer rotation speeds, slurry flow rate, etc. Finally, the tool also has a mechanism for wafer cleaning and metrology before and after the polishing.

![Figure 4.25: Schematic of a Polisher Configuration](image)
In this example, we use the thickness measurements provided by the scan head to evaluate the condition of the polish pad. During the process, the scan head sweeps back and forth on the wafer surface. There are 150 measuring points on the scanner for the measurement of layer thickness. Fig. 4.26 provides an example of the layer thickness from one measuring point, and the position of the scanner, during a polishing process. We can see that although the thickness is decreasing, it is neither linear nor monotonic. This is because the measurement is not taken for a static point on the wafer surface. As both wafer and scanner move with respect to different reference points, the measurement is taken for different points on the wafer. This explains the fluctuation on the signal.

Figure 4.26: Example of Reflexion Normalized Scan Head Measurement Data

For each measurement point, the feature variables we use in this application are extracted according to the following steps:

1. Remove a few sample points from the beginning and the end of measurement which are not take during the polishing step.
2. Fit a linear function \( kx + b \) using Least Square to the data obtained from step 1. The slope, \( k \), of the function can be considered as an approximation of the polishing rate, and is specified as one feature variable for the wafer.

3. Remove the linear trend from the data in step 2, such that the residuals reflect the fluctuation of the polishing on the surface and can be considered as the non-uniformity indicator.

4. In order to quantify this non-uniformity property, we calculate the standard deviation (\( \sigma \)) of the data obtained from step 3 and define it as another feature variable of this wafer.

Therefore, for each measurement we have defined 2 feature variables \((k, \sigma)\), and with 150 measurement points, we will have a total \( 2 \times 150 = 300 \) feature variables for each wafer.

Fig. 4.27 shows the production recipes over the course of over 3500 wafers on one of the platens before the pad is replaced. The PSIs of these recipes are summarized in Table. 4.2.

Figure 4.27: Production Recipes Before a Pad Change
4.2.2.2 Results

We build separate PLS model for each recipe. Fig. 4.28 gives the optimal RUL curves for each recipe. In contrast to previous example, the optimal RULs spread widely over the spectrum of target group, among which recipes 2, 5 and 8 have demonstrated some predictable pattern. The optimal RULs for the rest of recipes are on the edge of the group. A close look at the loading score charts (Fig. 4.29) will further help us determine if the variations are indeed as they are indicated by the optimal RULs.

![Optimal RUL for Recipes](image)

**Figure 4.28: Optimal Target RUL for the Polish Recipes**

For recipe 2, the loading scores confirm that although noisy, the variation displays a decreasing trend over the long term course. Similarly for recipe 5, scores on the first principal direction show a decreasing change, although at the end of the process the pattern becomes oblique. For the second principal component, an opposite
Figure 4.29: Loading Scores on Selected Principal Components of Different Recipes
observation is obtained, in which a increasing trend showed up while the rest are largely stable. For recipe 8, an obvious shift is seen by the end of the training part. Among others, recipe 1 and 7 have showed a shift at the early stage and remain relatively stable afterwards, which is consistent with the pattern showed by their optimal RULs. For the rest of recipes, the patterns shown in the score charts are generally inconsistent and heavily contaminated.

Therefore, the conclusion one can make at this point is that for recipe 2 there could be a prediction solution if the lead time is enough. For recipe 8, a detection solution is more suitable as the condition is indistinguishable after the shift. One of the common characteristics of the scores is that they all contain a large amount noise, which implies that the input is heavily contaminated. This leads us to question the data quality as well as the suitability of extracted features.

Furthermore, from the blocking contribution standpoint, one needs to validate that the variations from major contributors have a physical explanation for the degradation. In this case, since all sensors are of the same type, physically they are measuring the same properties of the process. And since both the scanner and wafer are not in a static position, the measurements do not have any site-specific information of the wafer. Therefore we can conclude that from measuring point perspective, all the variables should have similar influence on the degradation.

However, based on the blocking contribution charts (Fig. 4.30), the contributions from the measuring points are very different for each recipe. While we do not expect the contribution to be exactly the same for the measuring points, the difference where only variables from a few measuring points are relevant to the pad condition is unexplainable. Moreover, the difference in measurement point contribution of different recipes is also unexplainable. Additionally, from a statistical point of view, although the slope \( k \) and standard deviation \( \sigma \) are mathematically different quantities, their respective influence on the pad condition across recipes should be similar. But from
Fig. 4.30, we cannot see a consistent indication of their contributions.

Figure 4.30: Variable Block Contribution of Different Recipes
Based on these observations, we may conclude that the variations that are identified from the variables are not consistent with pad condition deterioration. More likely they are just some statistical coincidence in the training set. Considering that the thickness properties are important characteristics for control purposes, it is not uncommon that they are maintained at the specified range by the controllers, thus do not pertain any useful variations to indicate pad condition. In this case, further investigation is necessary to extract more useful feature variables. For example, as the pad is wearing out, the platen needs to apply more pressure to the wafer in order to achieve the desired polish rate, which not only increases the risk of breaking a wafer during the process, but also makes it difficult to control the process. Thus, if the force applied on the wafer can be measured, presumably it would be a better indicator for the pad condition.

4.3 Discussion

This chapter gives prognostic evaluation examples for both single and multiple process operation equipment. For the single process operation in case 1, we can see that the method is able to discover indicative feature variables for the failure, and the chosen variables are validated from another failure incident. Case 2 showed a detectable case where the abnormity is only observed prior to the failure, for which a boundary target RUL is identified and the actual variation is confirmed from the principal scores. Case 3 demonstrated a unpredictable failure mode where there is no signal to indicate degradation or provide early alarm. Although a boundary target RUL is chosen for the input, the principal score is not commensurate with the level of variation indicated by the optimal target. Thus it is concluded that no obvious variation exists in the data to indicate tool condition.

For multiple process operation, case 4 showed similar predictability for different recipes in terms of contributor type and variation quantity. These actually reflect
the process similarity. As noticed earlier, although the processes are different, their influences on the tool degradation are expected to be only quantitatively different instead of qualitatively. In other words, if a certain sensor is a good indicator for a specific failure type, it should have dominant effects regardless of which process is running on the tool. The difference is only the relative scale of the sensor reading from different processes. This prompts us to evaluate the similarity between process and utilize this information for RUL approximation when the historical data are not enough for a particular process.

In case 5, the model indicates different variations in the data, which leads to the conclusion of distinctive predictabilities. However, further review of variable contributions indicates the effects of inputs on the degradation are different even though they are the same type of features. From engineering analysis, it is obvious that if a physical property measured by a sensor is consistently indicative of the tool condition, then the same measurements from other similar types of sensors should also be indicative for the condition change. Thus, one can conclude in this case the variations identified by the model from the sensor are mere statistical coincidences. The example has demonstrated that there is no replacement of engineering judgment in any data-driven approach to determine if the discovery revealed by the model is indeed representation of the root-cause of the failure.
CHAPTER V

Conclusions and Future Works

5.1 Contributions and Broader Impacts

In this dissertation, we develop a unified method to systematically address some critical issues in prognostics for multiprocess manufacturing systems, including predictability evaluation, component remaining useful life estimation and degradation root cause identification. The relationship between failure mode predictability, data characteristics, and maintenance strategy is first clarified. In particular, the unique challenges as well as requirements are identified for semiconductor equipment prognostics.

In Chapter III, a novel method is developed for prognostics evaluation. With a group of target Generalized RUL curves defined to reflect a comprehensive library of degradation patterns, the partial least square algorithm is able to evaluate the variation from input variables and identify the best pattern that can be correlated with the target RUL. Based on the shape of the optimal RUL target, the user knows the degradation characteristics of the failure mode and thus is able to determine the predictability by comparing the RUL estimation given by the model with the lead time requirement of the maintenance practice. Several modeling issues such as optimal model parameter selection, predictability evaluation and root-cause diagnosis are properly addressed by various statistical techniques, such that the requirements
for engineering tuning is minimized and thus the usability is improved.

Both a simulation study and semiconductor use cases validate the performance of the proposed method. With multiple statistical variables defined from various recipes steps, the PLS model is able to identify the variations from input that are best for degradation representation. Based on that, an estimation of the component RUL is given by the model. Furthermore, a heuristic variable selection strategy is implemented in the model to further reduce the existence of noisy inputs and improve modeling results.

In addition, a variable blocking strategy is also formulated to help the user troubleshoot the failure. By grouping the variables hierarchically, users are able to drill down through the variables and identify the contributing parameters that demonstrate the variation for degradation. Based on the major contributors, engineering knowledge can be applied to diagnose if the measured properties are indeed the cause of the failure. With this manner, users can rapidly locate the failure cause and confirm the indication given by the model is physically explainable.

Furthermore, the unique challenges of prognostics for multiprocess equipment are identified. Motivated by the gaps that exist in current solutions, we extend the PLS approach to develop a novel approach for prognostics in a multiprocess environment. The discontinuity effect of multiprocess operation on a tool is described by the process switch index, and the feasibility of building individual models from partially observed data is quantified by the model trainability index. Furthermore, the relationship between process switch and model trainability is obtained by simulation, which validates the practicality of the proposed method.

With multiple run-to-failure historical records available for each type of failure mode, it is recommended to merge those records together for modeling such that the gap of each process is reduced. This in turn, minimizes the process switch index and improves the trainability of the model. Additionally, by using multiple sources
of training data, the inconsistent variation patterns can be avoided being selected as critical contributors.

Based on the assumption that the instantaneous impacts of individual processes on the equipment are independent from each other, we obtained the equivalent optimal RUL for different processes. With these, the equivalent RUL for all processes can be estimated when the tool is operating on any process at any stage of the overall degradation. With the process specific prognostics information, the production planning can be optimized such that longer lead time is available for the replacement by scheduling more runs of processes with smaller degradation rates as the failing point approaches.

5.2 Assumptions and Limitations

Although the developed method shows the capabilities of prognostic evaluation, we need to emphasize the assumptions as well as limitations such that it can be properly applied and improvement can be made in the future.

First, the method assumes the availability of long term run-to-failure data. Since the model assumes that the end of the training cycle corresponds to the failing condition, if the training data does not cover the failing period of the component, which can be the result of preventive maintenance practice, the RUL estimation given by the model would be shorter than the actual value and cause unnecessary waste of the replacement. Furthermore, as indicated by the trajectories of the predefined RUL group, the long term degradation would demonstrate a global decreasing trend over time if it is predictable. Thus, if the condition change in the short-term is a dynamic process, the model is unable to determine the predictability.

Secondly, the method assumes the process type is known at various period of the life-cycle. For semiconductor fabrication this will not be an issue as the recipe is defined to distinguish the process. But for other multi-regime engineering systems,
the change of operation condition may not be explicitly known. Thus, one needs to perform partitioning of the operational space in order to obtain the homogeneous condition for each regime. In addition, the transient periods between condition changes would introduce more dynamic features in the process and make the boundaries hard to discern.

Finally, for multiprocess operations, the method assumes that the instantaneous impact of each process on the tool is independent from the others. This assumption allows us to evaluate the equivalent impact of one process by the degradation pattern of other processes. But when the estimated RUL of one process is larger than one, it is infeasible to find the equivalent RUL of another process. In addition, if a section of the optimal target has a constant value of zero or one, it is also infeasible to find the equivalent RUL of other processes when the RUL estimations take the value of those same constants.

5.3 Recommendations of Future Work

While the results of the research have demonstrated the effectiveness of performing multiple prognostics tasks in an unified fashion, there are several aspects of the method that can be further improved in future.

5.3.1 Data Preparation

Although the techniques of data pre-processing are beyond the scope of this dissertation, the quality of input parameters has a vital effect on the performance of the proposed method. Although many sensors measure condition parameters, due to the nature of semiconductor fabrication, process conditions are required to be maintained at a static level during different steps of the recipe. Therefore, the feature variables extracted from production data are usually step-wise summary statistics, such as mean, standard deviation, etc. of particular steps, which are indicative parameters
for process stability. While these features are useful from a process monitoring perspective, their effectiveness for equipment condition monitoring is not optimal. As long as the recipe specification remains unchanged, the process controller would dictate the equipment to deliver the required operation condition whenever possible, thus disguising the tool degradation and reducing the observability of the tool condition.

In spite of the fact that the PLS model is able to disregard irrelevant variables and select the most useful features for RUL modeling, input feature variables need to be more equipment-oriented in order to reflect the tool condition. Currently, the transient periods between steps are usually neglected as they are not critical for process monitoring, but these data may have more information regarding equipment conditions because they represent the ability of a certain component to change the process condition. For example, as the flow rate changes between steps in a recipe, the smoothness and time of the transient period, which can be measured by the RMS of the second order derivative and point count of the transient period, may reflect the pump condition on the tool. However, as mentioned in section 2.1, the low sampling rate of semiconductor tools prevents further opportunities of applying techniques in the frequency domain of the signal for its dynamic properties. Recent advances in signal processing such as compressive sensing [12] may be worthwhile for investigation.

5.3.2 Definition of GRUL

In this work, the target RUL group for the PLS regression is defined with truncated quadratic curves. Whilst these target curves reflect various long term degradation possibilities for different types of components, as the system becomes more complicated they might not be the best match of the actual degradation. There are two possibilities to improve the definition of target GRUL. First, if engineering knowledge is available such that the degradation pattern of a particular component is known, one might define the shape of target RUL based on the expertise in order to have
a better representation of the actual degradation. In addition, the RUL estimation would also be more precise when the optimal RUL has improved fidelity. Another possibility is to apply adoptive training. Instead of using a group of predefined RUL, one can parameterize the target by spline curves [56] with variables controlling their shapes over the training cycle. Ranges of different spline parameters would then be specified to constrain the curves within a realistic region of component degradation. Iteration would start with initial target value and stop when a target is found to yield minimum fitting errors.

Another potential for modeling improvement is the estimation of prediction uncertainty. As bootstrapping is applied in this work, the requirement for computational resources is significantly high. In order to get the RUL prediction interval, it can take up to a minutes of computer run time. Thus, the relatively long data processing time may jeopardize the decision making process. Opportunities for improvement include estimating the confidence interval in the latent space of least square regression or implementing a recent method developed by Liao [53] using Gaussian Mixture Model (GMM) in the original input space.

### 5.3.3 Process Similarity

As observed in Chapter IV, due to the increasing product diversity and more frequent introduction of new technology, process requirements keep changing and new recipes are required to deliver the desired specifications. This progression in semiconductor manufacturing has posed significant challenges from an equipment monitoring standpoint as the impact of new processes on the tool is unknown at the beginning. Furthermore, as many of the components for which we desire to have a predictive maintenance solutions have relatively long life-cycles, the failure incidences for model training are hard to obtain within a short period of time. Therefore, the evaluation of equipment prognostics for a new process requires an even longer period.
In order to accelerate the developing cycle, and make full usage of available knowledge regarding equipment degradation from previous processes, it is necessary to evaluate the similarities between different processes and hence the possibilities of using RUL prediction models of available processes to approximate the impact of new ones. In addition, similar recipes can be grouped into more general categories to reduce the complexities in the system. Statistical classification techniques such as Fisher Discriminant Analysis (FDA) [64], and Discriminant Partial Least Square (DPLS) [5] have been applied for process state comparison and fault diagnosis [21, 33], both of which assume Gaussian-distributed state parameters. However, for degradation process similarity evaluation, challenges in comparing dynamics processes with non-Gaussian distribution need to be properly addressed. Recently Wang [85] developed a technique for RUL estimation based on the similarity of degradation trajectory. RULs of historical failing records are combined according to the degree of similarity between current process and historical process.
BIBLIOGRAPHY


[23] Deprost, D., and H. Ansquer (2005), Customized Data Elaboration (Pre-Treatment) for Advanced FDC, *AEC/APC XVII Symposium*.


[34] Heavlin, W., and J. Koslov (2007), Reliability Analysis of Rare Events, *ISMI Symposium*.


125


APPENDICES
APPENDIX A

The Directions of Maximal Covariance Optimization

Proposition A.1. The directions that solve the maximal covariance optimization (3.4) are the first singular vectors \( w_x = u_1 \) and \( w_y = v_1 \) of the singular value decomposition of \( C_{xy} = U\Sigma V' \); the value of the covariance is given by the corresponding singular value \( d_1 \).

Proof: Suppose the singular value decomposition of \( C_{xy} \) is \( U\Sigma V' \), thus the matrices \( U \) and \( V \) are orthonormal, and any \( w_x \) can be express as \( Uu_x \) for some \( u_x \). Then the solution to Eq. (3.4) becomes

\[
\max_{w_x, w_y: \|w_x\|_2 = \|w_y\|_2 = 1} C(w_x, w_y) = \max_{u_x, v_y: \|u_x\|_2 = \|v_y\|_2 = 1} (Uu_x)'C_{xy}Vv_y = \max_{u_x, v_y: \|u_x\|_2 = \|v_y\|_2 = 1} u_x'U\Sigma V'Vv_y = \max_{u_x, v_y: \|u_x\|_2 = \|v_y\|_2 = 1} u_x'\Sigma v_y. \tag{A.3}
\]

Clearly, when \( u_x = e_1 \) and \( v_y = e_1 \), the last equal sign (=) has a maximum of the largest singular value \( d_1 \). Hence Eq. (3.4) is solved by taking \( w_x = u_1 = Ue_1 \) and \( w_y = v_1 = Ve_1 \), the first columns of \( U \) and \( V \) respectively.