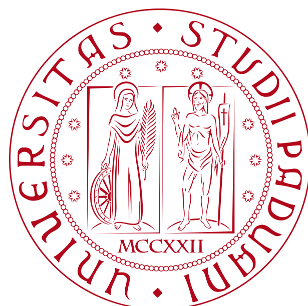


Università
degli Studi
di Padova

Statistical Methods for Semiconductor Manufacturing

Ph.D. candidate: Gian Antonio Susto

Director: prof. Matteo Bertocco
Advisor: prof. Alessandro Beghi

Ph.D. School in Information Engineering
Department of Information Engineering
January 2013

Essentially, all models are wrong,
but some are useful.

GEORGE E.P. BOX

Summary

In this thesis techniques for non-parametric modeling, machine learning, filtering and prediction and run-to-run control for semiconductor manufacturing are described.

In particular, algorithms have been developed for two major applications area:

- *Virtual Metrology (VM)* systems;
- *Predictive Maintenance (PdM)* systems.

Both technologies have proliferated in the past recent years in the semiconductor industries, called *fabs*, in order to increment productivity and decrease costs.

VM systems aim of predicting quantities on the *wafer*, the main and basic product of the semiconductor industry, that may be physically measurable or not. These quantities are usually 'costly' to be measured in economic or temporal terms: the prediction is based on process variables and/or logistic information on the production that, instead, are always available and that can be used for modeling without further costs.

PdM systems, on the other hand, aim at predicting when a maintenance action has to be performed. This approach to maintenance management, based like VM on statistical methods and on the availability of process/logistic data, is in contrast with other classical approaches:

- *Run-to-Failure (R2F)*, where there are no interventions performed on the machine/process until a new breaking or specification violation happens in the production;
- *Preventive Maintenance (PvM)*, where the maintenances are scheduled in advance based on temporal intervals or on production iterations.

Both aforementioned approaches are not optimal, because they do not assure that breakings and wasting of wafers will not happen and, in the case of PvM, they may lead to unnecessary maintenances without completely exploiting the lifetime of the machine or of the process.

The main goal of this thesis is to prove through several applications and feasibility studies that the use of statistical modeling algorithms and control systems can improve the efficiency, yield and profits of a manufacturing environment like the semiconductor one, where lots of data are recorded and can be employed to build mathematical models.

We present several original contributions, both in the form of applications and methods.

The introduction of this thesis will be an overview on the semiconductor fabrication process: the most common practices on *Advanced Process Control (APC)* systems and the major issues for engineers and statisticians working in this area will be presented. Furthermore we will illustrate the methods and mathematical models used in the applications.

We will then discuss in details the following applications:

- A VM system for the estimation of the thickness deposited on the wafer by the *Chemical Vapor Deposition (CVD)* process, that exploits *Fault Detection and Classification (FDC)* data is presented. In this tool a new clustering algorithm based on *Information Theory (IT)* elements have been proposed. In addition, the *Least Angle Regression (LARS)* algorithm has been applied for the first time to VM problems.

- A new VM module for multi-step (CVD, Etching and Litography) line is proposed, where *Multi-Task Learning* techniques have been employed.

- A new Machine Learning algorithm based on *Kernel Methods* for the estimation of scalar outputs from time series inputs is illustrated.

- Run-to-Run control algorithms that employ both the presence of physical measures and statistical ones (coming from a VM system) is shown; this tool is based on IT elements.

- A PdM module based on filtering and prediction techniques (*Kalman Filter*, *Monte Carlo* methods) is developed for the prediction of maintenance interventions in the Epitaxy process.

- A PdM system based on *Elastic Nets* for the maintenance predictions in Ion Implantation tool is described.

Several of the aforementioned works have been developed in collaborations with major European semiconductor companies in the framework of the European project UE FP7 *IMPROVE (Implementing Manufacturing science solutions to increase equiPment*

pROductiVity and fab pErformance); such collaborations will be specified during the thesis, underlying the practical aspects of the implementation of the proposed technologies in a real industrial environment.

Sommario

Nella tesi vengono descritte tecniche di identificazione non-parametrica di modelli, apprendimento automatico, filtraggio e predizione e controllo run-to-run con applicazione all'industria manifatturiera di semiconduttori.

In particolare sono stati sviluppati algoritmi per due applicazioni principali:

- sistemi di *Virtual Metrology (VM)*, Metrologia Virtuale;
- sistemi di *Predictive Maintenance (PdM)*, Manutenzione Predittiva.

Entrambe le tecnologie si stanno diffondendo nelle fabbriche di semiconduttori, dette *fab*, grazie al crescente bisogno di incrementare la produttivitá e diminuire i costi.

I sistemi di VM hanno lo scopo di predire quantitá, fisicamente misurabili o non, sul *wafer*, il principale prodotto dell'industria di semiconduttori. Le quantitá predette sono solitamente 'costose' da misurare, in termini economici o temporali: la predizione viene fatta a partire dalle variabili di processo e/o da informazioni logistiche sulla produzione che, contrariamente, sono sempre disponibili senza costi aggiuntivi per il loro utilizzo.

I sistemi di PdM hanno invece lo scopo di predire quando un intervento manutentivo sará necessario. Quest'approccio alla gestione delle manutenzioni, basato come la VM su metodi statistici e sulla disponibilitá di dati di processo/logistici, si contrappone alle classiche filosofie:

- *Run-to-Failure (R2F)*, dove non si agisce sulla macchina/processo fintantoché non si verifica una rottura o una violazione delle specifiche di produzione;
- *Preventive Maintenance (PvM)*, Manutenzione Preventiva, dove le mantenzioni vengono pianificate in anticipo in base ad intervalli temporali o a cicli produttivi.

Entrambi gli approcci sovraccitati non sono ottimali, in quanto non scongiurano rotture e sprechi di wafer e, nel caso della PvM, portano ad effettuare diverse manutenzioni non richieste o ad incrementare il numero di interventi non sfruttando a pieno il potenziale della macchina in esame o del processo.

L'obbiettivo principale di questa tesi é quello di dimostrare, attraverso una serie di applicazioni e studi di fattibilitá, come l'utilizzo di algoritmi di modellizzazione statistica e di controllo possano migliorare efficienza, produttivitá e guadagni di un ambiente manifatturiero, come quello dei semiconduttori, in cui si dispone di un ricco insieme di informazioni su processi/macchine che possono essere utilizzate per costruire modelli matematici.

Nella tesi vengono presentati diversi contributi originali, sia in termini di applicazione che metodologici.

Nella prima parte della tesi viene proposta una panoramica sull'industria di semiconduttori: saranno illustrate le pratiche piú diffuse per quanto concerne i sistemi di *Advanced Process Control (APC)* e le sfide maggiori e piú importanti per gli ingegneri e statistici che lavorano in questo settore. Successivamente verrá fornita una carrellata sui metodi e modelli matematici utilizzate nelle applicazioni.

Piú in dettaglio vengono discussi i seguenti argomenti:

- Un sistema di VM per la stima dello spessore depositato dal processo di *Chemical Vapor Deposition (CVD)* sul wafer, a partire da dati di *Fault Detection and Classification (FDC)*, dove é stato proposto un nuovo algoritmo di clustering basato su elementi di *Information Theory (IT)*. Inoltre, l'algoritmo *Least Angle Regression (LARS)* é stato per la prima volta applicato in tale applicazione.

- Un modulo di VM per una configuzione di multi-processo CVD, Etching e Litografia, dove sono state utilizzate tecniche di *Multi-Task Learning*.

- Un nuovo algoritmo di Machine Learning basato su *Kernel Methods* per la stima di uscite scalari a partire da ingressi di tipo serie temporale.

- Algoritmi di controllo Run-to-Run che sfruttano la presenza di misure statistiche provenienti da sistemi di VM basato su elementi di IT.

- Applicazione di tecniche di predizione e filtraggio (*filtro di Kalman*, metodi *Monte Carlo*) per la predizione di interventi correttivi per il processo di Epitassia in un modulo PdM.

- Sistema PdM basato su *Elastic Net* per la predizione di rotture in macchine di Ion Implanting.

La ricerca che ha portato ai risultati sopra descritti é stata svolta per la maggior parte in collaborazione con importanti aziende di semiconduttori europee, nell'ambito del progetto UE FP7 *IMPROVE (Implementing Manufacturing science solutions to*

increase equiPment pROductiVity and fab pErformance); tali collaborazioni saranno specificate nel corso di questa tesi, cercando di mettere in risalto anche gli aspetti pratici dell'implementazione in una realtá industriale delle tecnologie descritte.

Contents

I Semiconductor Manufacturing: Introduction and Challenges in Advanced Process Control **1**

1 The Fab World and Advanced Process Control **3**

1.1 Motivations and Thesis Overview 3

1.2 Fabrication of Semiconductor Devices 6

1.3 Advanced Process Control (APC) Systems: Virtual Metrology 9

1.4 APC Systems: Predictive Maintenance 12

1.5 APC Systems: Fault Detection and Classification 15

1.6 APC Systems: Run-to-Run Control 16

II Methods and mathematical tools **19**

2 Machine Learning for Regression **21**

2.1 Elements of Machine Learning and Regularization 22

2.2 Kernel Methods 27

2.3 Neural Networks 29

3 Variable Selection Techniques **33**

3.1 Forward Stepwise Regression 34

3.2 Stagewise Regression 34

3.3 The Least Angle Regression Algorithm 35

4 Filtering and Prediction **41**

4.1 Recursive Bayesian Estimation 42

4.2 Kalman Predictor 43

4.3 Particle Filter 44

4.4 Kernel Density Estimation 46

III Virtual Metrology 51

5 A VM Case Study for Chemical Vapor Deposition (CVD) Modeling 53

5.1 Introduction to Virtual Metrology and Applications 53

5.2 Introduction to VM for CVD 57

5.3 Problem Formalization and Data Preprocessing 59

5.4 Experimental Results 63

5.5 Clustering 71

6 Multi-Step Virtual Metrology 75

6.1 Introduction 76

6.2 Multistep Virtual Metrology 78

6.3 Results 81

6.4 Conclusions 86

7 Virtual Metrology with Time Series Data 87

7.1 Introduction and Problem Statement 88

7.2 Feature extraction 90

7.3 SAFE: Supervised Aggregative Feature Extraction 91

7.4 SAFE: Time Series Approximation 93

7.5 SAFE: Shape Function Parametrization 94

7.6 SAFE: Derivatives Basis Expansion 97

7.7 Experimental results 99

8 Run-to-Run Control and VM 105

8.1 Introduction 106

8.2 Review on R2R and VM 107

8.3 The proposed approach 112

8.4 Experimental Results 116

8.5 Conclusions 120

IV Predictive Maintenance 121

9 Predictive Maintenance for Epitaxy 123

9.1 Introduction 124

9.2 Equipment and Data Description 127
9.3 Problem Formalization and Proposed Module 130
9.4 Experimental Results 135
9.5 Optimal Tuning of Epitaxy Pyrometers 142
9.6 Fab Implementation and Conclusions 149

10 Predictive Maintenance for Ion-Implantation 153

10.1 Introduction 154
10.2 Tool Description and Problem Formalization 156
10.3 Experimental Results 158
10.4 Conclusions 162

V Conclusions 163

11 Conclusions 165

References 169

Part I

Semiconductor Manufacturing: Introduction and Challenges in Advanced Process Control

1

The Fab World and Advanced Process Control

1.1 Motivations and Thesis Overview

Semiconductor manufacturing is one of the most technologically advanced industrial sectors. This field, while being among the most technology-oriented and cost-intensive industrial sectors, has a massive impact on everyday's life. As a matter of fact, semiconductor-based devices are pervasive and ubiquitous: personal computers, mobile phones and cars are only the most straightforward examples.

For more than 50 years, the capabilities (processing speed, memory capacity, sensors, etc.) of semiconductor-based digital devices has improved exponentially following Moore's law (see Fig. 1.1 and Schaller, 1997). This improvement rate has dramatically enhanced the impact of digital electronics in nearly every segment of the world economy.

Given such premises, it is not surprising that semiconductor manufacturing companies are spending effort and resources to improve quality and open the way to smaller, faster,

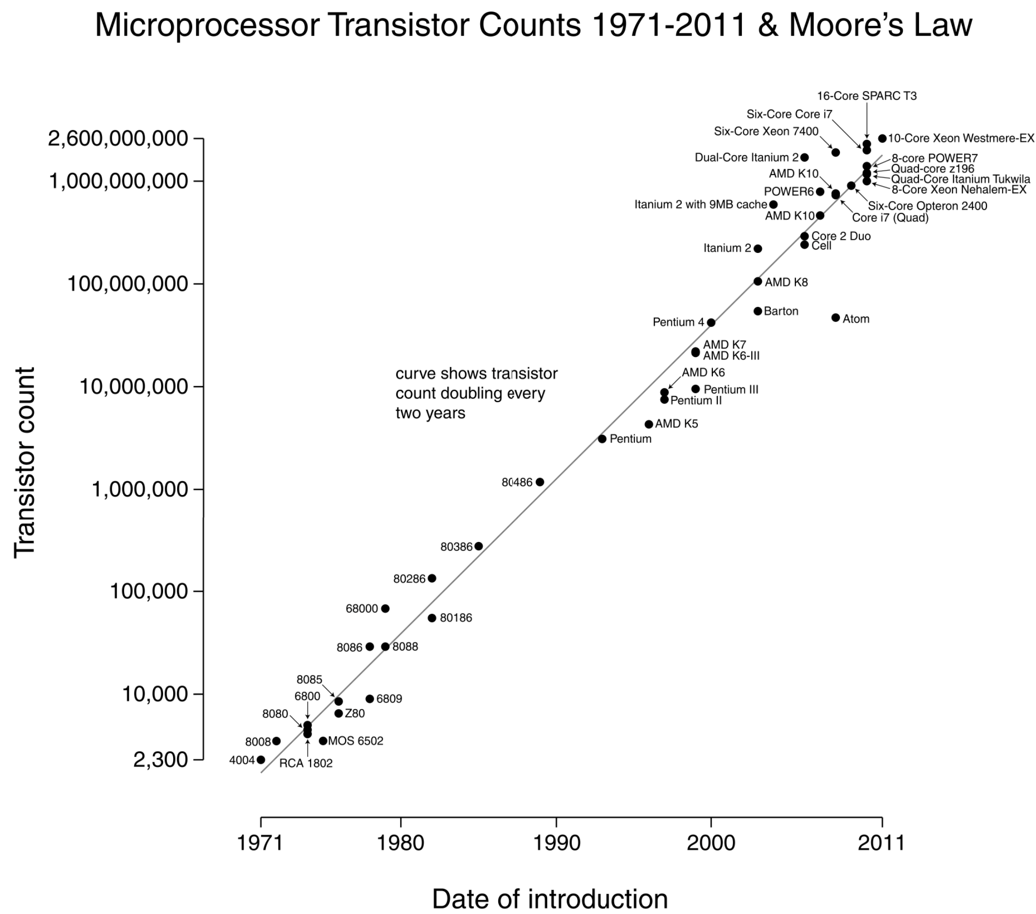


Figure 1.1: Plot of CPU transistor counts against dates of introduction. The line corresponds to exponential growth with transistor count doubling every two years in the logarithmic vertical scale. Photo courtesy of `http://commons.wikimedia.org/wiki/User:Wgsimon`

higher quality devices. Process quality and control are critical for decreasing costs and increasing yield. The contribution of automatic controls and statistical modeling in this area can drastically impact production performance.

In the milestone paper Edgar, Butler, Campbell, Pfeiffer, Bode, Hwang, Balakrishnan, and Hahn (2000), the (at that time) future challenges for modeling and control in microelectronics manufacturing were presented. In the past 12 years intense research activity has been going on in this area, largely enabled by the advances in machine learning and computation capability. As described in Edgar et al. (2000), the variations in process and tool properties due to long-term production runs, the limited understanding on such complicated processes and the lack of automated operational practices (especially from the maintenance point of view), suggest that there is a huge margin for improvements in this area.

In this thesis the contributions of non-parametric modeling, machine learning, statistical methods and, partially, automatic controls to semiconductor manufacturing are reviewed and some original works have been produced. The final goal of this thesis is to prove, through several examples and applications, that the use of statistical modeling algorithms and control systems can improve efficiency, yield and profits of a manufacturing environment such as the semiconductor one, where lots of data are recorded and can be employed in mathematical models. Semiconductor companies are investing more and more resources in these topics to improve their manufacturing capabilities. Recently, for example, the major European Nanoelectronics Industries have focused their efforts on developing statistical metrology/predictive systems to decrease the number of defective products, increase process stability and even decrease the number of physical measures performed, see the websites of ENIAC (2012) and of IMPROVE (2012).

In this introductive part of the thesis some examples of the following applications areas are shown:

- *Virtual Metrology (VM)* systems;
- *Predictive Maintenance (PdM)* systems;
- *Fault Detection (FDC)* systems;
- *Run-to-Run (R2R)* control.

All of these technologies have proliferated in the past few years in semiconductor manufacturing facilities, called *fabs*, in order to improve the productivity and decrease costs.

The rest of this introductive Part is organized as follows: in Section 1.2 an overview of semiconductor fabrication is provided, while the most common practices in *Advanced Process Control (APC)* systems and the major issues for engineers and statisticians working in this area are then presented in Sections 1.3 (VM), 1.4 (PdM), 1.5 (FDC) and 1.6 (R2R control with VM measures in the control loop).[1]

The core of this thesis will be focused on Virtual Metrology (VM) and Predictive Maintenance (PdM) topics, to whom Part III and Part IV are respectively dedicated. Those two APC dedicated Parts will be preceded by a methodological one, Part II, that contains most of the mathematical tools employed in the thesis.

1.2 Fabrication of Semiconductor Devices

This section describes, with no aim of completeness, the fabrication process of a semiconductor device. For a more detailed description the interested reader is referred to Quirk and Serda (2001).

The entire semiconductor manufacturing process, from the first stage up to final product shipping, takes usually six to eight weeks and it is performed in highly specialized fabrication plants. The process is composed of four main steps (Chang, 1997):

i) Wafer formation: a wafer (see Fig. 1.2) is a thin (125 - 300mm diameter and 525 - 775 μm) slice of semiconductor material - usually silicon crystal - that serves as the substrate for microelectronic devices. Wafers are formed from extremely pure (99.9999% purity) crystalline material; the process to create such crystalline wafers, is the Czochralski process, depicted in Figure 1.3.
The wafer, being the main product of the semiconductor fabrication will be usually considered in the following of this thesis as defining the discrete step iteration of any process, i.e. one process iteration will corresponds to one wafer being processed on that tool (if not differently stated). Production is usually organized in group of 25 or 50 wafers, called *lots*.

ii) Front end processing: this step relates to the formation of transistor chips on the silicon wafer and is performed in controlled environments known as *clean rooms*; in such rooms the level of *pollutants* (dust, vapors, particles) is artificially kept at a fixed level by means of air filtering and restricted access policies. The front end process encompasses the following sub-steps:

1. *Wafer-cleaning*: since Ultra-Large Scale Integration (ULSI) technology is char-

[1] Part of the contents of this chapter have been presented in Susto, Pampuri, Schirru, DeNicolao, McLoone, and Beghi (2012c).

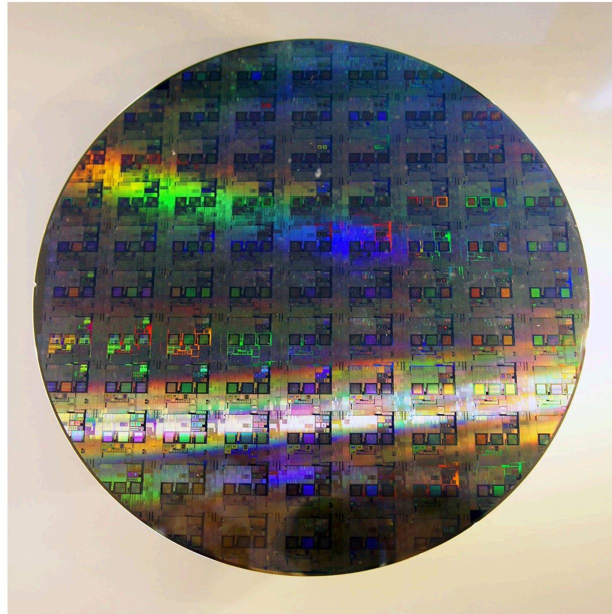


Figure 1.2: A 30 cm silicon wafer. Photo courtesy of `http://commons.wikimedia.org/wiki/File:12-inch_silicon_wafer.jpg`

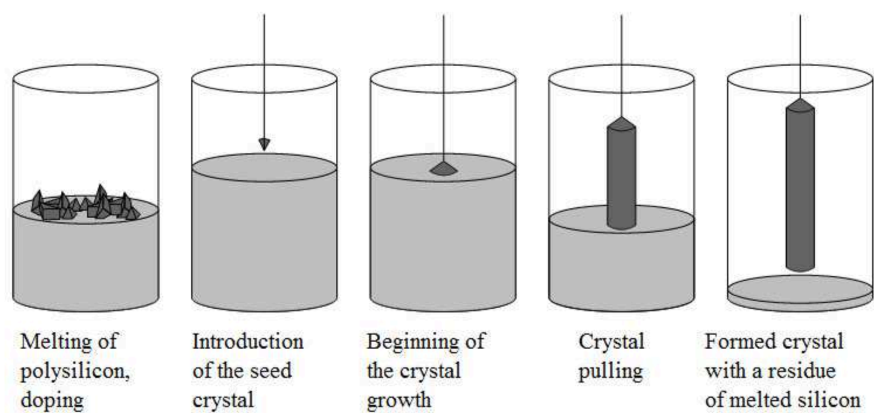


Figure 1.3: The main stages of the Czochralski process.

acterized by strict requirements concerning surface smoothness and particle contamination, the wafers need to be prepared for further processing by means of cleaning procedures. Table 1.1 summarizes the sources and effects of the various contaminations (Chang and Chao, 1996).

Table 1.1: Sources and effects of the various contaminations

Contamination	Possible source	Effects
Particles	Equipment, ambient, gas, chemical	Low oxide breakdown
Metal	Equipment, chemical, reactive ion etching	Low breakdown field, reduced minority lifetime
Organic	Vapor in room, residue of photoresist	Change in oxidation rate
Microroughness	Initial wafer material, chemical	Low oxide breakdown
Native oxide	Ambient moisture	Degraded gate oxide, high contact resistance

2. *Deposition*: dieletric and polysilicon film deposition is widely used in Integrated Circuits (IC) fabrication. Dieletric films, including silicon dioxide and silicon nitride, serve as isolation, mask, and passivation layers; polysilicon film can be used as a conducting layer, semiconductor, or resistor by proper doping with different impurities. The main deposition techniques are *CVD* (Chemical Vapor Deposition) and *PVD* (Physical Vapor Deposition); other processes include plasma-assisted deposition, photo CVD, laser CVD, Rapid-Thermal Processing CVD (RTPCVD), and Electron-Cyclotron Resonance (ECR) CVD, see Cheng (1996).

3. *(Photo)Lithography*: several techniques may be used to create ULSI circuit patterns on wafers; the most common process relies on *photomask* exposition. An ultraviolet radiation is transmitted through the clear part of the mask, while the opaque part blocks the rest of the radiation. The resist film, being sensitive to the radiation, is then coated on the wafer surface. The mask is aligned within the required tolerance on the wafer; then radiation is applied through the mask and the resist image is developed, see Nakamura (1996).

4. *Etching*: devices are built from a number of different layer materials sequentially deposed. Lithography techniques are used to replicate circuit and device features, and the desired patterns are transferred by means of etching. In ULSI technology, the etching process is very sensitive because of strict dimensional requirements (fraction of a micrometer). The etching process can be *dry* or *wet*, see Lii (1996).

It should be noted that the above mentioned process steps are repeated several times during front-end processing to produce multiple interconnected layers on the wafer

surface.

iii) Testing: before a wafer is sent to chip preparation, every single IC on the wafer is tested for functional defects (test end-of-line) (DiPalma, 2005). The tests can be *parametric* or *electrical*.

- Parametric tests are performed on *ad-hoc structures* prepared on the device to monitor the efficiency of process steps and the goodness of the design. Such structures are called *TAG*, and lie in the *scribe lines*. Usually there are less than 10 TAGs per wafer. Parametric tests consist of electric measurements of physical quantities (impedance, capacitance, resistance, etc.).

- Electrical tests verify that the behavior of each device is consistent and within specifications; this capability is assessed by means of electrical testing with sequential measurements; if some value is out of specification range, the circuit is flagged as faulty. The non-passing die is marked with a small dot of ink, and the passing\non-passing information stored in a *wafermap*.

iv) Packaging (or Back end): the purposes of packaging are to provide electrical connection, protect the chip from mechanical and environmental stress and provide a proper thermal path for the heat that the chip generates. Packaging plays a crucial role with respect to performance and reliability of the chip and the system in which the package is applied (Tachikawa, 1996).

The Front end processing step is the one where machine learning and automatic control techniques can have the most impact on production quality. In the next sections we will gave an overview of the main technologies that have been developed in the last decade in this area.

1.3 Advanced Process Control (APC) Systems: Virtual Metrology

This section is a short introduction to the topic of Virtual Metrology (VM), that will be intensively developed in Part III of this thesis.

A VM system consists of a mathematical model of the system under consideration (Ringwood, Lynn, Bacelli, Ma, Ragnoli, and McLoone, 2010) for estimating a 'costly to measure' physical variable where tool variables are used as inputs. These quantities are usually 'costly' to measure in economic or temporal terms and just few wafers in a lot (Section 1.2) are measured: the prediction is based on process variables and/or logistic

information on the production that, instead, are always available and that can be used for modeling without further costs.

The benefits of the introduction of a VM system are several; with few measurements on a lot, equipment-performance drifts between lots are difficult to be promptly detected (Ringwood et al., 2010; Hung, Lin, Cheng, and Lin, 2007). A VM system allows to predict values of the relevant variables (in the situation at hand, CVD thickness), without increasing the number of physical measurements by exploiting statistical analysis on tool data and available measurements. Moreover, several semiconductor manufacturing processes benefit of the presence of a Lot-to-Lot (L2L) controller (Sachs, Hu, and Ingolfsson, 1995; Khan, Moyne, and Tilbury, 2007). Based on the physical measurements performed on one or two wafers in a lot, the process parameters acting on the following lot are updated. The introduction of a VM system may lead to a more accurate, Wafer-to-Wafer (W2W), control policy that allows to detect and reduce the number of faulty wafers.

VM systems have been proposed in the literature for CVD (Hung et al., 2007; Cheng, Huang, and Kao, 2007; Huang, Huang, Cheng, Liao, and Chang, 2008; Ferreira, Roussy, and Conde, 2009; Wu, Lin, Wong, Jang, and Tseng, 2008), Etching (Kang, Lee, Cho, Kim, Park, Park, and Doh (2009); Lynn, Ringwood, Ragnoli, McLoone, and MacGearailt (2009); Cheng, Chen, Su, and Zeng (2008); Lin, Cheng, Wu, Kao, Ye, and Chang (2009); Ragnoli, McLoone, Lynn, Ringwood, and MacGearailt (2009); Monahan (2005); Lin, Cheng, Ye, and Hung (2008), and Lithography (Huang, Cheng, and Hung, 2009) processes. Also, fab-wide VM structures have been proposed by Khan et al. (2007); Huang, Su, Cheng, and Jian (2007) and Su, Yu, and Ogunnaike (2008).

Besides high prediction accuracy, desirable properties of an efficient VM system are:

- *reasonably low computational times*, since new products are added monthly to fab production and the behavior of tools change over their maintenance cycles, therefore models need to be constantly updated and computed;
- *interpretability*, so that it is possible to identify which variables in the model are the most meaningful, a very appealing property for FDC purposes.

As partly described in Susto and Beghi (2012c), modeling of semiconductor manufacturing processes is a challenging task mostly due to four main factors:

1. *high dimensionality* - hundreds of input variables are available making the regression problem computationally expensive and difficult to solve;
2. *data fragmentation* - hundreds/thousands of products are run on the same machine, with different tool settings (called *recipes*); in the case of some tools, the dataset

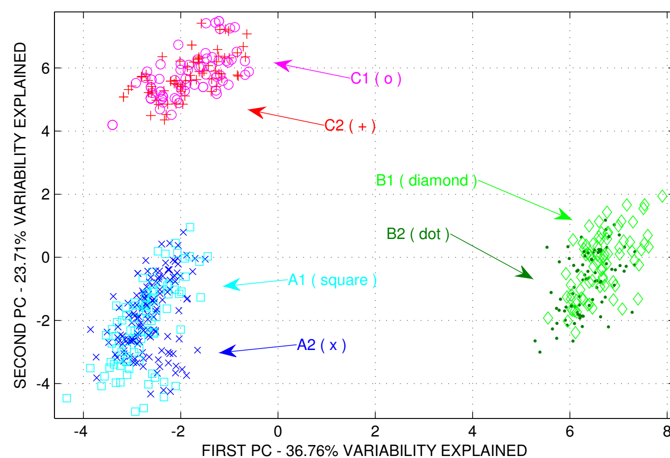


Figure 1.4: The first two Principal Components (PCs) of the physical variables of a CVD tool with three chambers (A, B and C), each one with 2 sub-chambers (1 and 2). Picture adapted from Susto and Beghi (2012c).

is even further complicated by the fact that each product has a different target, and the equipment may be composed of 2 or 3 separated chambers that exhibit different behaviors. As shown in the example reported in Fig. 1.4, chambers of the same tool can usually be considered as completely different machines.

3. *multi processes modeling* - semiconductor production processes involve a high number of sequential operations and the quality features of a certain wafer depend on the whole processing and not only on the last step before measurement; unfortunately VM modules proposed so far only take into account one physical process.

4. *time series input data* - many semiconductor modeling problems require the estimation of a scalar output from one or more time series. Such VM problems are usually tackled by extracting a fixed number of features from the time series (like their statistical moments), with a consequent loss in information that leads to suboptimal predictive models. Moreover, feature extraction techniques usually make assumptions that are not met by real world settings (e.g. uniformly sampled time series of constant length), and fail to deliver a thorough methodology to deal with noisy data.

A substantial part of modern VM literature is focused on how to tackle the aforementioned issues. The previous issues will also be addressed in the VM dedicated Part of this thesis (Part III): high-dimensionality and data fragmentation will be discussed in Chapters 5 and 6; Chapter 6 will be dedicated to a multi processes modeling example, while Chapter 7 will be focused on a new approach for statistical learning and modeling

with time series data.

1.4 APC Systems: Predictive Maintenance

Efficient management of maintenance and control actions on a process is essential to decrease the costs associated with defective wafers and equipment inactivity. Maintenance policies can be divided into four categories, with different levels of complexity and efficiency, (Susto, Beghi, and DeLuca, 2012a; Mobley, 2002):

- *Run-to-Failure (R2F) Maintenance*: when repairs or restoration actions are performed after the occurrence of a failure. This is the simplest approach to maintenance management and usually the most costly one due to the large number of defective products obtained as a consequence of the failure.

- *Preventive Maintenance (PVM)* (or Scheduled Maintenance): when the maintenance is carried out periodically on a planned schedule with the aim of anticipating the process failures. In this approach, failures are usually avoided, on the other hand, unnecessary maintenances are sometimes performed.

- *Condition-Based Maintenance (CBM)*: when the actions on the process are taken after the verification of one or more conditions indicating a degradation in the process or equipment. This approach is based on continuous monitoring of the machine/process health and enables maintenance to be performed only when it is actually needed. The drawback of CBM management is that maintenance cannot be planned in advance.

- *Predictive Maintenance (PdM)* (or Statistical Based Maintenance): similarly to CBM, maintenance actions are taken only when necessary. However, prediction tools are used to assess when such actions are likely to be required, facilitating planning and scheduling schemes. PdM systems can employ ad-hoc defined health factors or, in general, statistical inference methods.

Several authors (Mobley, 2002) indicate with the names CBM and PdM the same class of maintenance policy while others consider the two categories separated. Sophisticated maintenance tools, such as those belonging to the CBM and PdM classes, are clearly associated with initial, installation, and development costs, that are however paid off by the increase in system uptime and percentage of non defective products and decrease in the number of test wafers employed. Besides the above mentioned advantages, it has also been shown (Hyde, Doxsey, and Card, 2004) that the introduction of a PdM system in

the production line can increase the Process Capability Index C_{pk} (Montgomery, 2007), that is a widely adopted statistical measure of the ability of a process to produce output within specification limits.

The PdM techniques usually define and exploit a *Health Factor (HF)* (Chen and Blue, 2009), that is a quantitative index of the status of the equipment. It is a function of observable facilities parameters (historical time series, characteristic behavior of the equipment, sensor data, and so on) and can be employed to:

- assess future status of the equipment or one of its components;
- take strategic decisions about maintenance scheduling;
- provide information for dynamic sampling plans, (Pasadyn and Toprac, 2002).

The concept of HF is usually widely adopted also in Fault Detection and Classification (FDC) systems and this leads to some overlap between the two categories (see Section 1.5).

While all VM problems can be tackled with regression approaches, for PdM, depending on the problem, several techniques may be suitable for modeling and predicting faults and scheduling maintenance interventions, making this area more complex and challenging than VM. As a result the PdM area is much less developed than VM, albeit significant progress has been made in the last decade. For example:

- in Rying (2001) a *wavelet*-based approach has been used to identify important features for detection of process faults;
- in Pampuri, Schirru, DeLuca, and DeNicolao (2011a) *survival models theory* is employed for the same goal;
- *regression methods* have been employed also in this area; linear approaches, such as Ridge Regression and Elastic Nets, have been used in Susto, Schirru, Pampuri, and Beghi (2012d), while NNs have been adopted for modeling in Wu, Gebraeel, Lawley, and Yih (2007);
- *Filtering and Prediction* techniques, like Kalman Predictor and Particle Filters, have also been recently employed in PdM for semiconductor manufacturing processes in Butler and Ringwood (2010); Schirru, Pampuri, and DeNicolao (2010b) and Susto, Beghi, and DeLuca (2011a);
- *Classification Methods*, more specifically, Support Vector Machines have been considered in Baly and Hajj (2012).

Given the variety of available methodologies and goals, PdM problems are typically addressed in a customized approach.

The PdM problems usually suffer, even more than the VM ones, from the *lack of a sufficient amount of observations* to prepare a reliable statistical model: this is due to the fact that the maintenance interventions are of course in far fewer than the number of measured wafers (observations for VM problems). For this reason, it is of paramount importance in the modeling to exploit the information coming from similar processes/equipments. This concept has been adopted in Susto, Pampuri, Schirru, and Beghi (2012b) with the employment of Multi-Task techniques.

Another major issue is represented by the *non-trivial evaluation of the impact of a PdM* in an industrial environment and the comparison of the performance of a PdM system versus a R2F/PvM approaches. In Susto et al. (2012d,a) the performances of the proposed PdM systems are evaluated in terms of two indicators:

1. *type I error* - number of not prevented maintenances N_{UB};
2. *type II error* - number of process iterations that may have been performed if the maintenance interventions suggested by the PdM systems would not have been performed N_{BL}.

Based on the costs associated with the two errors, the maintenance system can be tuned to be more or less reactive: in the example reported in Fig. 1.5 the tuning is done through the choice of scalar parameter k_T, see Susto et al. (2012d) for details. Clearly this performance evaluation can only be done on R2F dataset and this is a huge limitation. Not only that, but before adopting a PdM approach instead of a PvM, the costs associated with the lack of planned scheduling should be taken into account, see Susto et al. (2012d).

All the aforementioned difficulties explained why semiconductor industries are still tackling several maintenance problems with R2F and PvM approaches (Kalir, 2012).

In Part IV of this thesis the topic of PdM will be further investigated with some examples; we will also justify the adoption of PdM tools in a industrial environment by showing the performances of the lasts w.r.t. classical maintenance management approaches.

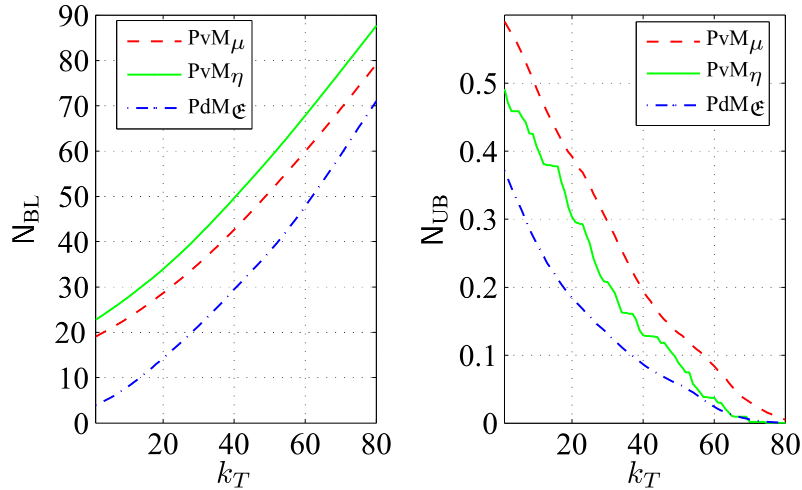


Figure 1.5: The performances of two PvM systems (PvM$_\mu$ and PvM$_\eta$) versus the ones of the PdM system PdM$_\mathcal{E}$ as a function of the threshold k_T. Figure adapted from Susto et al. (2012d).

1.5 APC Systems: Fault Detection and Classification

Fault Detection and Classification (FDC) methods have been widely applied in the past years (Adamson, Moore, Passow, J.Wong, and Xu, 2006; Moore, Harner, Kestner, Baab, and Stanchfield, 2006; Schirru, Pampuri, and DeNicolao, 2010a). In contrast with PdM techniques, an FDC system does not predict the future behavior of the tool/process, but, in the case of a fault, aims at identifying the root cause of the abnormal behavior. This is of particular interest in the everyday work of a semiconductor plant: the root causes of faults in a complex process may be dozens, sometimes hundreds, and even expert process engineers have difficulty understanding their pathology and, therefore, how to properly cope with the faulty process/tool.

Sometimes PdM modules/approaches are based on FDC systems and this is the reason why FDC is sometimes a misused word for PdM; in Goodlin, Boning, Sawin, and Wise (2003) for example the PdM module constantly monitor the FDC results as a sort of HF. In that work the FDC system simultaneously detects and classifies different faults from different *control charts*. Another work where control charts are used for defining a FDC system is Schirru et al. (2010a), where chamber matching is obtained with multi-level linear models (see Fig. 1.6).

The FDC modules usually employ *classification* techniques (Hastie, Tibshirani, and Friedman, 2009); for example *K-Nearest-Neighbour (kNN)* in He and Wang (2007), *Principal Component-based kNN* in He and Wang (2010) and *Support Vector Machines*

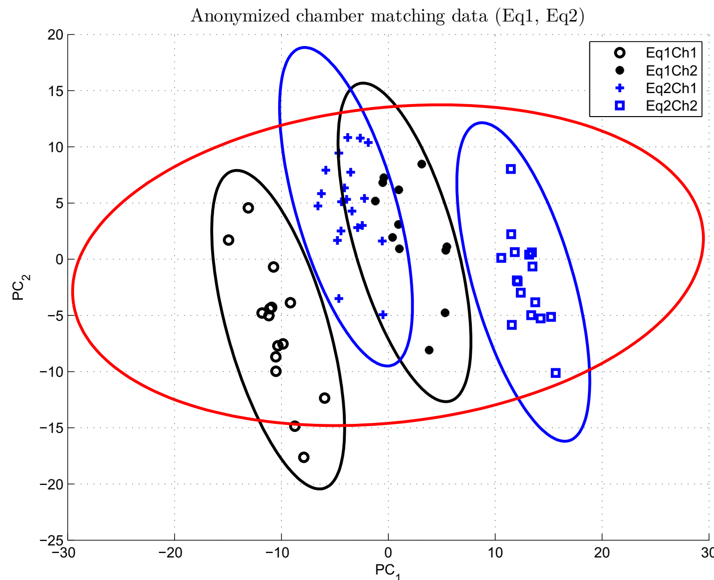


Figure 1.6: Figure adapted from Schirru et al. (2010a). The ellipses represent the confidence intervals for single chambers and intra-chambers: this enables a double level of monitoring of the process under examination.

(SVMs) in Sarmiento, Hong, and May (2005).

FDC systems are affected by some of the same data challenges described for VM and PdM: lack of observations, huge data fragmentation, high-dimensionality, multi process causes. A common problem for FDC and PdM is usually the lack of structured data for maintenances; usually faults and corrective actions are recorded manually by process/maintenances engineers and the resulting lists are incomplete or the same maintenance or fault cause may be indicated with different names. This, and several other problems not cited in this Section, underline how, to be successful in the work of applying machine learning and control methods to semiconductor manufacturing, it is of paramount importance to closely collaborate with industrial partners to understand the problem and the complexity of the datasets.

1.6 APC Systems: Run-to-Run Control

Run-to-Run (R2R) has become the standard approach for process control in Semiconductor plants (Boning, Moyne, Smith, Moyne, Telfeyan, Hurwitz, Shellman, and Taylor, 1996; Toprac, Downey, and Gupta, 1999) in the last decades. Despite its simplicity, R2R control presents several advantages such as improved process and device performance, decreased tool downtime, improved process throughput, reduction of defective wafers and

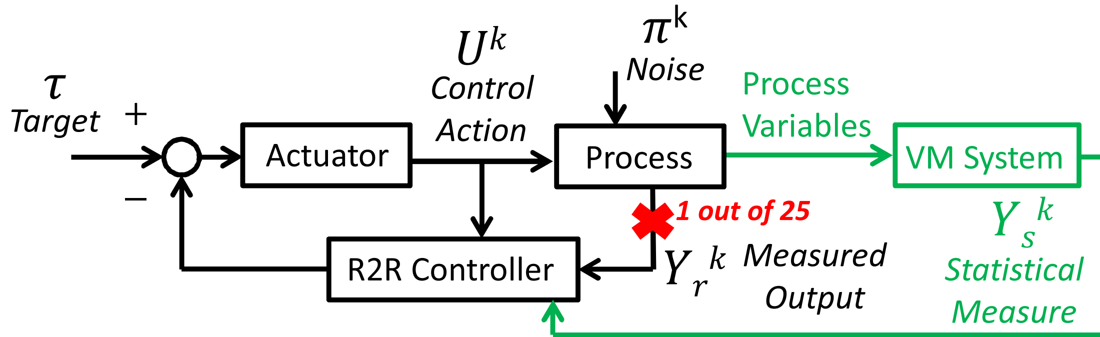


Figure 1.7: R2R control scheme with Physical and VM (statistical) Measures.

early detection of process drifts (Anderson and Hanish, 2007).

R2R techniques are based on physical measurements of quality parameters (such as layer thickness or Critical Dimensions). Considering the common sampling practices of measuring a small number of wafers for each lot (usually 1 out of 25 wafers), it is apparent why R2R controllers operate on a Lot-to-Lot (L2L) control policy that allows for corrective actions to be taken at lot level (Toprac et al., 1999). R2R controllers are generally implemented through EWMA-based algorithms, see Chen and Guo (2001) and Zhang, Deng, and Baras (2003).

With the development and adoption of VM systems in recent years, this scenario has changed as control systems have the possibility of incorporating this new information source in their calculations. The presence of statistical measurements for each wafer and the reduction of physical measure should be taken into account when implementing a control strategy. In Fig. 1.7 a qualitative block scheme for R2R controllers with VM module in the loop is depicted.

In Cheng et al. (2008) and Susto, Schirru, Pampuri, DeNicolao, and Beghi (2012e) VM and physical measurements are treated differently depending on their probabilistic distributions, with different approaches. This research topic is however in its infancy, largely because VM has only become a well established technology in the last few years.

The approach firstly presented in Susto et al. (2012e) will be resumed and expanded in Chapter 8.

Part II

Methods and mathematical tools

2

Machine Learning for Regression

Machine learning methodologies are nowadays applied in many industrial and scientific environments including technology-intensive manufacturing (Monostori, 2003; Facco, Doplicher, Bezzo, and Barolo, 2009), biomedical sciences (Pillonetto, Dinuzzo, and DeNicolao, 2010), and in general every data-intensive field that might benefit from reliable predictive capabilities, like the semiconductor industry.

Machine learning techniques exploit organized data to create mathematical representations (*models*) of an observable phenomenon. It is then possible to rely on such a model to provide predictions for unobserved data.

Depending on the type of output that we are trying to estimate, we have two classes of statistical learning problems

- if the output is quantitative, we are dealing with a *regression* problem;
- if the output is qualitative, the problem is a *classification* one.

While both types of problems have been faced in the past years in the modeling of semiconductor manufacturing, regression topics have dominated the scene, given the fact that all Virtual Metrology and also some Predictive Maintenance problems are regression ones.

In this thesis several user cases exploit regression techniques (the entire VM dedicated Part of the thesis, Part III, and the PdM Section 10). In this Section we will provide an introduction to regression.

In mathematical terms, let

$$\mathcal{S} = \left\{ x_i \in \mathbb{R}^{1 \times p},\ y_i \in \mathbb{R} \right\}_{i=1}^{n} \tag{2.1}$$

be a *training dataset*. In this formalism, n observations of a certain phenomenon are available. The i-th observation (or *example*) is characterized by p input features, constituting the vector x_i, and a scalar target value y_i.

In practical terms, the input space usually relates to easily obtained data, while the target value is either not always available or results from a costly procedure; in a typical industrial application, x_i would collect sensor readings during a process operation, while y_i would be a quantitative indicator of product quality. The goal is then to exploit the information provided by $\mathcal{S}$ to create a predictive model f

$$f: \quad \mathbb{R}^p \to \mathbb{R} \tag{2.2}$$

$$x \mapsto f(x) \tag{2.3}$$

such that, given a new observation $\tilde{x} \notin \mathcal{S}$, $f(\tilde{x})$ will provide an accurate prediction of the unobserved $\tilde{y}$: in the case of the above mentioned industrial example, the model f would be able to estimate the final product quality relying only on sensor readings collected during process operation.

2.1 Elements of Machine Learning and Regularization

From $\mathcal{S}$ we can construct a regressor matrix X, where $X[i,j] = x_i^j$ is the value of the i-th observation of the j-th regressor. Once the regressor matrix X is obtained, it is possible to employ a machine learning technique to find a predictor model f.

As a first assumption, let the structure of the model be specified by a vector of parameters θ. Consider the fitness function

$$\mathcal{L}(\theta) = \mathcal{F}(\theta) + \lambda \mathcal{R}(\theta) \tag{2.4}$$

and the solution of the optimization problem

$$\theta^* = \arg\min_{\theta} \mathcal{L}(\theta).$$

The *error term* $\mathcal{F}$ measures the approximation power of f (with respect to $\mathcal{S}$), while the *regularization term* $\mathcal{R}$ governs the trade-off between prediction accuracy (driven by the generic loss function $\mathcal{L}$) and model complexity. Furthermore, $\lambda \geq 0$ is a *hyperparameter* that acts as a tuning knob for the trade-off between approximation and variability: too small a value results in an overfitted model (specifically tuned on the training set, with low predictive power), while too large a value results in an underfitted model (which would not incorporate the necessary information for making good predictions). The insight is that the correct value of λ would result in only the relevant information being incorporated into the model, yielding the highest predictive power.

While a wide variety of choices are possible for both $\mathcal{F}$ and $\mathcal{R}$, for tractability most learning techniques require the minimization problem to be convex with respect to θ; in order to exploit this desirable feature, it is sufficient for $\mathcal{F}$ and $\mathcal{R}$ to be convex.

Let first consider $\mathcal{R} = 0$. Notably, if f is defined as a linear function of θ such as

$$f(X;\theta) := X\theta$$

and $\mathcal{F}$ is the sum of squared estimation residuals

$$\mathcal{F} := \mathcal{Q}(\theta) \tag{2.5}$$

$$:= ||Y - f(X;\theta)||^2 = ||Y - X\theta||^2 \tag{2.6}$$

$\mathcal{Q}(\theta)$ is a quadratic Loss Function and therefore its minimum always exists (even if it is not necessarily unique). Minimizing equation (2.6) is equivalent to minimizing the Residual Sum of Squares (Khan, Moyne, and Tilbury, 2008; Hastie et al., 2009):

$$\text{RSS}(\theta) = \sum_{i=1}^{n}(y_i - x_i^T\theta)^2 = (Y - X\theta)^T(Y - X\theta) \tag{2.7}$$

where θ is a row vector of p real coefficients, Y is the column vector of the output observations and $X \in \mathbb{R}^{n\times p}$ is the matrix of the inputs.

By differentiating (2.7) w.r.t. θ we get

$$X^T(Y - X\theta) = 0, \tag{2.8}$$

and, if (X^TX) is nonsingular, then the solution of the *Ordinary Least Square (OLS)*

problem (2.10) is given by the well-known solution

$$\hat{\theta} = (X^T X)^{-1} X^T Y \, . \tag{2.9}$$

The prediction

$$\hat{y}_i = x_i^T \hat{\theta} = x_i^T (X^T X)^{-1} X^T Y \, . \tag{2.10}$$

gives the best fitting estimate, in terms of minimization of prediction residuals.

Equation (2.10) is prone to numerical issues and instability, since there is no guarantee that $X'X$ will be full rank or well conditioned. Also OLS (2.10) solution usually does not guarantee good prediction on a validation dataset: this phenomenon is related to overfitting of training observations. Moreover, when dealing with high-dimensional regression problem (typical of semiconductor manufacturing datasets) where the number of regressors p can be larger then the number of observations n, the OLS estimations (2.10) can be prone to overfitting and high variance problems (Hastie et al., 2009).

In order to prevent such issues, we consider an objective function where we also penalize complicated models with $\mathcal{R} > 0$. With different choices of $\mathcal{L}$ and $\mathcal{R}$ we obtain different well known regularization problem

- $\mathcal{L} = \mathcal{Q}$, $\mathcal{R} = \mathcal{R}_{\mathfrak{R}} = \sum_{j=1}^{p} \theta_j^2$: *Ridge Regression ($\mathfrak{R}$)* (Hoerl and Kennard, 1970);
- $\mathcal{L} = \mathcal{Q}$, $\mathcal{R} = \mathcal{R}_{\mathfrak{L}} = \sum_{j=1}^{p} |\theta_j|$: *LASSO ($\mathfrak{L}$)* (Tibshirani, 1996).

In general, such methodologies make additional assumptions on the properties of $f(x)$, in order to improve new observations prediction accuracy (Hastie et al., 2009): from a Bayesian point of view, this is equivalent to imposing a prior on the model structure.

Ridge Regression is a well-known shrinkage method that, thanks to the L_2 penalization term, shrinks the coefficients θ; by doing so the variance of the prediction is decreased (by paying with more bias) with, generally, an improvement of predictions accuracy. Ridge Regression allows a closed form solution

$$\hat{\theta}_{\mathfrak{R}} = \left(X^T X + \lambda I \right)^{-1} X^T Y, \tag{2.11}$$

where the presence of term λI also prevents bad conditioning in computation of the inverse. This method is equivalent to maximizing the conditional probability $p(Y|X)$ when assuming $Y|X \sim \mathcal{N}(X\theta, \sigma^2 I)$ or $Y = X\theta + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ (that is i.i.d. Gaussian noise). The Bayesian interpretation of RR sees the $\lambda\theta'\theta$ term derive from the prior distribution of θ, $p(\theta)$ (assuming $\theta \sim \mathcal{N}(0, \lambda^{-1} I)$), while $\mathcal{F}_{\mathfrak{R}} = \mathcal{L} + \mathcal{R}_{\mathfrak{R}}$ is

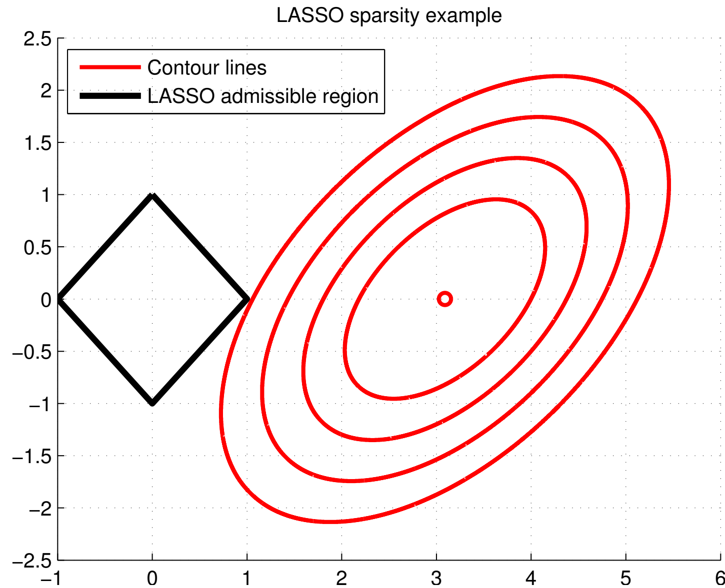


Figure 2.1: 2-D graphical example of the sparsity of the LASSO: the contour lines of a quadratic score function (red), whose optimal unconstrained solution is (3.1, 0), meet the LASSO constraint (black) in (1, 0): a sparse model results.

the log posterior distribution (Hoerl and Kennard, 1970). The larger λ, the smaller the "complexity" of the selected model (the variance of the estimator), at the cost of worsening the performances on the training set $\{X, Y\}$ and introducing some bias; in practical applications, λ is often used as a "tuning knob" controlling the bias/variance trade-off. The best value of λ is usually obtained via cross-validation or other statistical criteria;

The L_1 penalization term, instead, leads LASSO coefficients to be sparse (see Fig. 2.1); this makes the LASSO a really appealing technique nowadays, given the fact that variable selection has become one of the key problems in statistics. Sparse solutions are also preferred given the high level of interpretability. Under a Bayesian framework, LASSO optimal coefficients can be interpreted as maximum a posteriori estimates when the coefficients θ_j have independent and identical Laplace priors distribution (Tibshirani, 1996). This formulation allows to obtain a sparse solution for θ (that is, some entries of the selected θ are 0) if λ is low enough. This extremely convenient property of the LASSO allows for the creation of low-order models even when the input space has high dimension: intuitively, such a model is able to improve the stability of the prediction without sacrificing its precision (Ramirez, Lecumberry, and Sapiro, 2010).

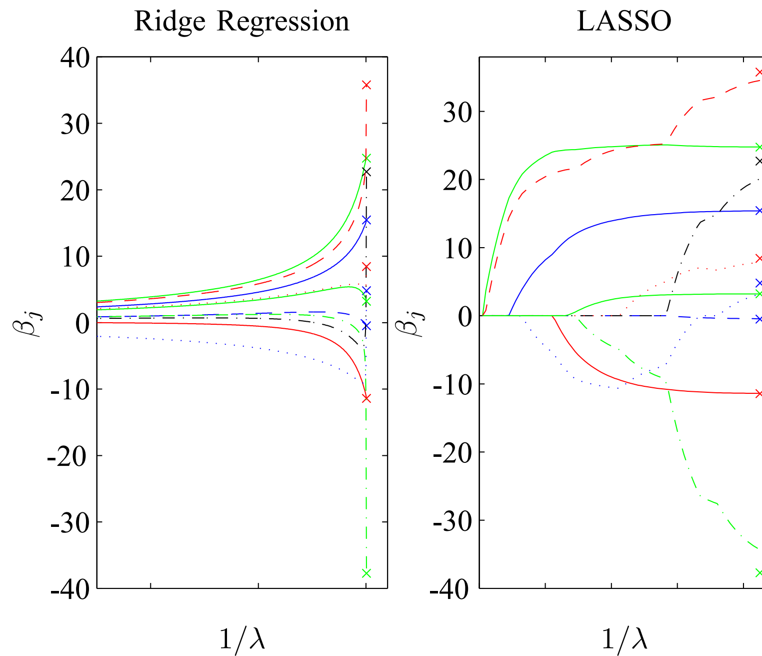


Figure 2.2: Ridge Regression and LASSO coefficients path for the diabetes data. With the crosses are indicated the coefficients of the OLS.

LASSO can be implemented through the Least Angle Regression (LARS)[1] (Efron, Hastie, Johnstone, and Tibshirani, 2004) or with Sequential Minimal Optimization (Platt, 1999). LASSO just recently has gained attention in the statistical metrology and modeling of semiconductor manufacturing process, for VM (Pampuri, Schirru, Fazio, and DeNicolao, 2011b; Schirru, Pampuri, DeLuca, and DeNicolao, 2011) and Predictive Maintenance (Susto et al., 2012b,d) purposes.

The previous characteristics are illustrated in a classical toy example based on diabetes data (Efron et al., 2004) where $n = 442$ and $p = 10$. In Figure (2.2) are reported the evolution of the $\mathfrak{R}$ and $\mathfrak{L}$ the coefficients at the change of the regularization parameter λ. $\mathfrak{R}$ coefficients tends to stay closer one to another the more λ is great. $\mathfrak{L}$ coefficients tends to enter the model (to be different from zero) one at the time; we can choose how many coefficients are different from zero by modifying the value of λ. From Fig. 2.2 it can also be appreciated that, with λ close to zero, the coefficients obviously tends to converge to the OLS solution (as complexity of the model is poorly penalized).

Ridge Regression and LASSO usually outperform OLS solution and none of the two methods always guarantees better prediction accuracy than the other. However, LASSO

[1] The LARS is a Variable Selection algorithm based on geometric considerations; the algorithm and its strong relation will the LASSO will be discussed in details in Chapter 3.

is generally more appealing and widely adopted in modern data analysis due to the importance of having sparse results. However, LASSO also presents some drawbacks (Li and Lin, 2010):

(i) if $p > n$, the LASSO selects at most n variables (Zou and Hastie, 2005);

(ii) if there is a group of variables with very high correlation, then LASSO selects only one variable from the group and does not care which one is selected;

(iii) if $p < n$, with highly correlated variables, it has been shown in Tibshirani (1996) that the prediction performance of the LASSO is dominated by Ridge Regression.

High correlation of variables and the possibility of having $p > n$, are often encountered in semiconductor manufacturing dataset and therefore LASSO seems not to be perfectly suited to deal with the problem at hand.

To overcome the aforementioned issues, a method called Elastic Net ($\mathfrak{E}$) (Zou and Hastie, 2005) has been developed. The Elastic Net is a sort of combination of $\mathfrak{R}$ and $\mathfrak{L}$ and minimize the following objective function

$$\mathcal{F}_{\mathfrak{E}} = \mathcal{Q} + \lambda_1 \sum_{j=1}^{p} \theta_j^2 + \lambda_2 \sum_{j=1}^{p} |\theta_j| \,, \tag{2.12}$$

in which the regularization term is a combination of L_1, like in Ridge Regression, and L_2 penalization term, like in the LASSO.

Usually the Elastic Net outperforms the LASSO in terms of prediction accuracy while still encouraging sparse representation (Zou and Hastie, 2005). In the following work the algorithm *glmnet* presented in Friedman, Hastie, and Tibshirani (2010) will been used for the implementation of the Elastic Net, which is based on cyclical coordinate descent methods.

2.2 Kernel Methods

Kernel Methods allow the use of linear techniques for solving nonlinear learning problems (Scholkopf and Smola, 2001). In fact, it is typical of semiconductor manufacturing modeling problems that several processes exhibit non-linear relationships between variables, it is therefore convenient to include non-linear relationships into the models.

We introduce the concept of Kernel Methods with a 2-Dimensional Classification problem example. Two classes of labeled data are presented in this problem, and the goal of the classifier is to assign new non-labeled data to one of the two classes. A simple

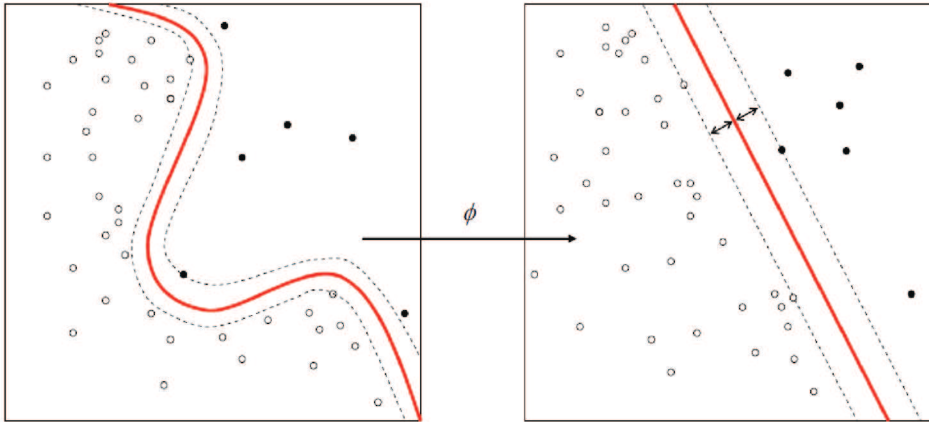


Figure 2.3: Kernel Methods applied to a 2-Dimensional Classification problem. The kernel map allow to split class 1 (○) and class 2 (●) with a line.

linear approach to do so is to divide the two dimensional space with a line that separates the two labeled data.

As depicted in the left panel of Fig. 2.3 this is not feasible: the data are not separable with a line. However, if we apply a transformation ϕ of the data, as shown in the right panel of Fig. 2.3, the data transformed in the new dimensions are separable with a linear approach. This kernel mapping allows to resolve a non-linear problem with linear techniques. This is true also the case of regression problem. In the following we formalize such type of approaches.

We introduce a map

$$\phi : \quad \mathbb{R}^p \to \mathbb{R}^q \tag{2.13}$$

$$x \mapsto \phi(x) \tag{2.14}$$

where the dimension q of the feature space can be much more greater than p.

In order to obtain a nonlinear model f without giving up the desirable convexity features of the optimization problem, it is possible to exploit the so-called *kernel trick* (Aizerman, Braverman, and Rozoner, 1964) to embed a nonlinear projection of X on a Reproducing Kernel Hilbert Space (RKHS) (Aronszajn, 1950) in a quadratic optimization problem. In the case of Ridge Regression for example, the regression on the feature space is

$$f_{\mathfrak{R}}(x) = x \left[\phi(X)'\phi(X) + \lambda I\right]^{-1} X'Y; \tag{2.15}$$

Noting that

$$\left[\phi(X)'\phi(X) + \lambda I\right]^{-1} X' = X' \left[\phi(X)\phi(X)' + \lambda I\right]^{-1},$$

then equation (2.15) may be rewritten as

$$f_{\mathfrak{R}}(x) = xX' \left[\phi(X)\phi(X)' + \lambda I\right]^{-1} Y \quad (2.16)$$

$$= \langle x, X\rangle \left[\langle \phi(X), \phi(X)\rangle + \lambda I\right]^{-1} Y, \quad (2.17)$$

where $\langle \cdot, \cdot \rangle$ defines the inner product.

It can be seen from (2.17) how the new features $\phi(X)$ enter the model via the inner products only. It is therefore not necessary to compute the mapping $\phi(X)$ explicitly, but instead we just need to compute the *Kernel Function* (Scholkopf and Smola, 2001)

$$K(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle \quad (2.18)$$

for each couple $\{x_i, x_j\}_{i,j=1,\dots,n}$ and for each new observation $\{x, x_j\}_{j=1,\dots,n}$.

Common choices of Kernel Functions $K(x_i, x_j)$ are (Hastie et al., 2009):

$$d\text{th Degree Polynomial:} \quad (1 + \langle x_i, x_j\rangle)^d, \quad (2.19)$$

$$\text{Radial Basis:} \quad \exp\left(-\|x_i - x_j\|^2 / c\right), \quad (2.20)$$

$$\text{Neural Network:} \quad \tanh\left(a\langle x_i, x_j\rangle + b\right). \quad (2.21)$$

Ridge Regression has been used here for introducing kernel methods, but even other regression algorithms can be 'kernelized' if the feature space enters the algorithm only as inner product.

A thorough review of machine learning techniques and Kernel-based techniques is beyond the scope of this thesis. The interested reader is referred to Hastie et al. (2009); Muller, Mika, Ratsch, Tsuda, and Scholkopf (2001) and Scholkopf and Smola (2001).

2.3 Neural Networks

A pure non-linear approach is represented by Neural Networks.

A Neural Network (NN) is a network of interconnected artificial neurons (ANs) where the outputs are weighted, possibly nonlinear transformations of the inputs. NN-based models exhibit excellent flexibility and computational properties. A NN is composed by 3 kinds of layers:

- an *input layer (L_{in})*, where the corresponding parameters are associated with input

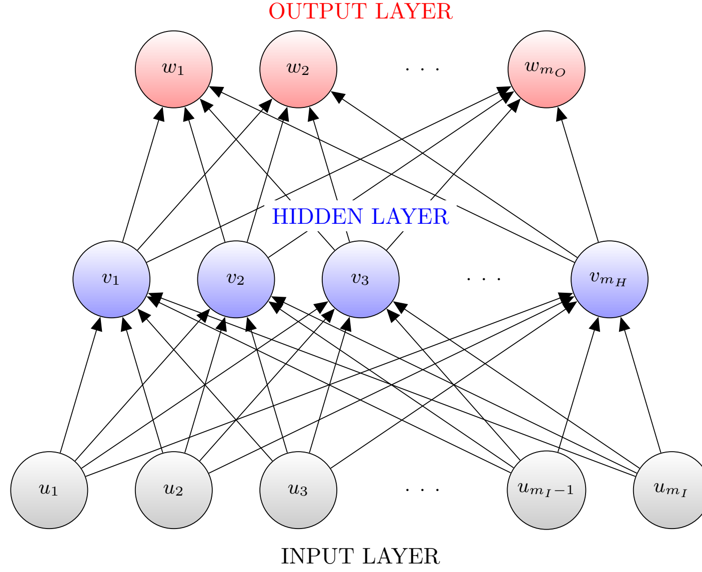


Figure 2.4: Schematic of a single hidden layer, feed-forward neural network.

variables (in the problem considered here, the FDC parameters);

- *hidden layers* (L_{hidden}) (one or more);
- an *output layer* (L_{out}), where the nodes correspond to the parameters that have to be predicted (thickness, as far as the CVD process is concerned).

In this thesis we consider Feed-Forward Multilayer Perceptron (MLP) NNs where no loops are present between the layers. It has been shown that a Feed-Forward NN with one hidden layer can approximate any function, and this is the most used scheme amongst NN in black-box identification (this is also referred as *vanilla NN*, see Hastie et al. 2009).

In Fig. 2.4 a general scheme for a Feed-Forward MLP with one hidden layer is shown. Nodes represent variables while arches are associated to functions that describe interconnections between variables. In the scheme there are $\{u_i\}_{i=1}^{m_I}$ inputs and $\{w_i\}_{i=1}^{m_O}$ outputs to be modeled. Features $\{v_i\}_{i=1}^{m_H}$ are created from linear combinations of the inputs

$$v_i = h_a(\alpha_{0i} + \alpha_i^T U), \quad i = 1, \dots, m_H, \tag{2.22}$$

where U is the matrix of the inputs, while outputs in turn are created from linear combinations of the created features

$$w_i = h_b(\beta_{0i} + \beta_i^T V), \quad i = 1, \dots, m_O, \tag{2.23}$$

where V is the matrix of the hidden features. The *activation function* $h_a(\cdot)$ is usually chosen to be non-linear (sigmoid, arctan, radial-basis function, as shown in Lu, Sundararajan, and Saratchandran (1998)), while the *output function* $h_b(\cdot)$ is typically chosen linear for regression problems.

Let us suppose that there are $\{u_i\}_{i=1}^{m_I}$ inputs and $\{w_i\}_{i=1}^{m_O}$ outputs to be modeled. Features $\{v_i\}_{i=1}^{m_H}$ are created from linear combinations of the inputs

$$v_i = h_a(\alpha_{0i} + \alpha_i^T U), \quad i = 1, \ldots, m_H, \tag{2.24}$$

where U is the matrix of the inputs, while outputs in turn are created from linear combinations of the created features

$$w_i = h_b(\beta_{0i} + \beta_i^T V), \quad i = 1, \ldots, m_O, \tag{2.25}$$

where V is the matrix of the hidden features. The *activation function* $h_a(\cdot)$ is usually chosen to be non-linear (sigmoid, arctan, radial-basis function Lu et al. (1998)), while the *output function* $h_b(\cdot)$ is typically chosen linear for regression problems. Coefficients $\alpha.$ and $\beta.$ are called *weights*. For more details we refer the reader to Hastie et al. (2009) or Hecht-Nielsen (1989).

Coefficients $\alpha.$ and $\beta.$ are called *weights* and are chosen in such a way to minimize the Mean Squared Error (MSE)

$$\text{MSE} = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}, \tag{2.26}$$

where, as before, n is the number of observations, y the real output and $\hat{y}$ the predicted output. The algorithm that is commonly employed for the training of the NN is the *back-propagation*, where weights are computed in a two-phase procedure where, after an initial guess, the prediction error are computed and the propagated backwards in the NN structure to correct the weights; then with the new weights the new prediction errors are computed; this procedure is iterated several times in order to reach small values of (2.26). For more details we referred the reader to Hastie et al. (2009); Hecht-Nielsen (1989).

NNs usually grant good performance in terms of data fitting, however, they are specified in terms of a number of parameters that are critical to be set. We will discuss in detail these issues in Chapter 5.

Another important class of NN is represented by *Radial Basis Neural Networks* that exploit Radial Basis Functions (Buhmann, 2003); this kind of NN played a major role in NN modeling in the past years and have been also used in the modeling of semiconductor

manufacturing problems (Hung et al., 2007). We refer the interested reader to Karayiannis and Mi (1997).

3

Variable Selection Techniques

Beside belonging to the class of Machine Learning techniques, *Variable Selection* techniques have a important role in the modeling of semiconductor manufacturing problems, for this reason this Chapter is entirely dedicated to this class of algorithms. In Variable selection techniques the reduction of the model complexity is done within the modeling algorithm and not *a-priori*, like with correlation analysis (Susto and Beghi, 2012c).

The LASSO has already been introduced in previous section, we will illustrate Forward Stepwise and Stagewise Regression in Section 3.1 and 3.2 respectively, while the Least Angle Regression (LARS), a technique that is closely related to LASSO, is presented in detail in Section 3.3.

3.1 Forward Stepwise Regression

Stepwise Selection (SS) is the most widely adopted approach in this class of techniques. SS is an iterative method where at each iteration the regressor that is more correlated with the current output residual is included in the model. In the field of modeling for semiconductor manufacturing, SS has been widely adopted in VM problems: see Ferreira et al. (2009); Kang et al. (2009); Lynn et al. (2009) and Ragnoli et al. (2009).

Algorithm 1: Forward Stepwise Linear Regression

Data: Training data: X, y.
Result: Linear Model Coefficients: θ.

1. Start with $r = y$, $\theta = 0$.
2. Find the predictor x^j most correlated with r.
3. Update $\theta_j \leftarrow \theta_j + r^T x^j$.
4. Set $r \leftarrow r - (r^T x^j) \cdot x^j$, where $\cdot$ indicates the scalar multiplication.
5. *Stopping Rule* - Repeat from 2) until the model is not 'improving enough' to justify the inclusion of another term.

Stepwise Forward Selection (FS) is illustrated in Algorithm 1. The *stopping rule* described at point 5) of the algorithm can be implemented in several ways; a common strategy is to base this decision on the F statistics (Hastie et al., 2009). A different approach is called *Backward SS*, where the algorithm starts with the full model and sequentially deletes regressors.

SS is a simple way to eliminate regressors that do not have much influence on the output, however, this approach is considered in the statistical community as an aggressive fitting procedure that can eliminate predictors that are statistically significant.

3.2 Stagewise Regression

A more effective approach than SS, less greedy, but computationally more expensive, is the Stagewise Selection (SgS). The Forward Stagewise Linear Regression, or simply Stagewise, procedure is described in Algorithm 2. The choice of ϵ is crucial for the performances of the Stagewise; if $\epsilon = |r|$ the Stagewise clearly became the classical Stepwise Selection, while with small values of ϵ we can avoid the greed of the FS at the cost of more iterations of the algorithm. As stated in Section 1.3, computational cost

Algorithm 2: Forward Stagewise Linear Regression

Data: Training data: X, y.

Result: Linear Model Coefficients: θ.

1. Start with $r = y$, $\theta = 0$.
2. Find the predictor x^j most correlated with r.
3. Update $\theta_j \leftarrow \theta_j + \epsilon \cdot \text{sign}(r^T x^j)$, where ϵ is a small positive coefficient.
4. Set $r \leftarrow r - \epsilon \cdot \text{sign}(r^T x^j) \cdot x^j$.
5. *Stopping Rule* - Repeat from 2) until the model is not improving enough to justify the inclusion of another term.

is a serious issue in VM problems, and this is probably the reason why SgS has never been considered for modeling in VM problems and, more generally, for Semiconductor manufacturing modeling.

In next Section we describe in detail Least Angle Regression (LARS) (Efron et al., 2004), a model selection algorithm that yields solutions similar to those of SgS, but with a smaller computational cost (like SS). LARS is closely related to another very popular variable selection technique, the LASSO, described in the previous Chapter.

3.3 The Least Angle Regression Algorithm

The LARS procedure is described in Algorithm 3.

Point (3) of the algorithm is the fundamental difference between LARS and Stepwise Selection. Instead of continuing along x^j, the LARS proceeds in a direction equiangular between x^j and x^k until a third variable has the same correlation with the current residual as the equiangular versor of x^j and x^k.

The procedure is depicted in Fig. 3.1, in the case of $p = 2$. $\bar{y}_2$ is the projection of the output in $\text{span}(x_1, x_2)$. The initial estimate is $\hat{\mu}_0 = 0$. The covariate with most correlation with the residual vector $(\bar{y}_2 - \hat{\mu}_0)$ is x_1: after next iteration of the LARS we have the estimation $\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 x_1$, where $\hat{\gamma}_1 x_1$ is chosen such that $(\bar{y}_2 - \hat{\mu}_0)$ bisects the angle between x_1 and x_2. Analogously, the next estimation will be $\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 u_2$ where u_2 is the versor that bisects x_1 and x_2. This geometrical interpretation of the LARS should clarify where the name Least Angle Regression came from. In next subsections, 3.3 and 3.3, we will provide all the practical details of the LARS algorithm, namely, how to compute the equiangular versor and when to stop when we move along it.

Algorithm 3: Least Angle Regression

Data: Training data: X, y.

Result: Linear Model Coefficients: θ.

1. Start with $r = y$, $\theta = 0$.
2. Find the predictor x^j most correlated with r.
3. Increase θ_j in the direction of sign$(r^T x^j)$ until some other competitor x^k has as much correlation with current residual as does x^j.
4. Move (θ_j, θ_k) in the joint least squares direction for (x^j, x^k) until some other competitor θ_l has as much correlation with the current residual.
5. Continue until the desired number of predictors has entered in the model (at each iteration of the process one variable enters in the model). At p-th iteration we obtain the Ordinary Least Square solution.

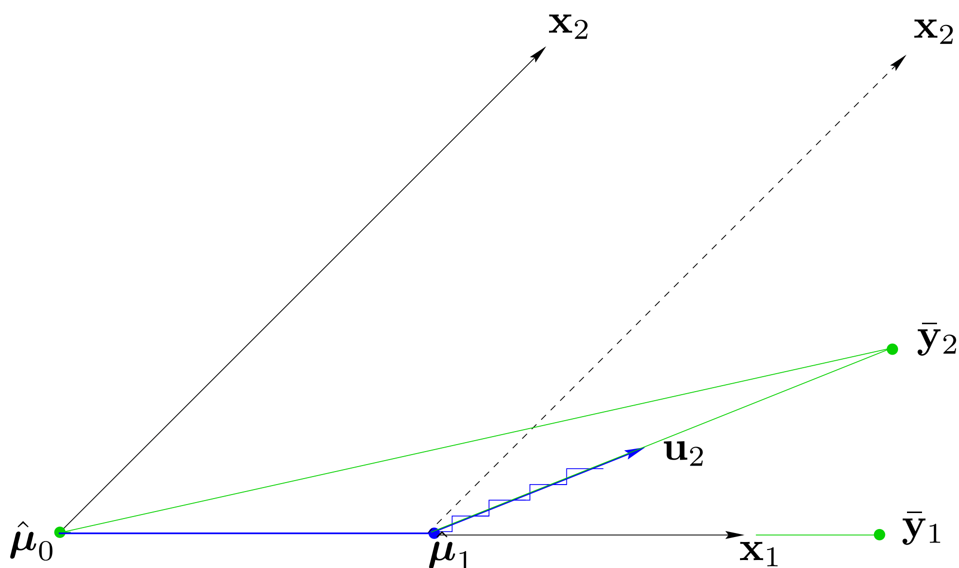


Figure 3.1: The LARS algorithm in the case of $p = 2$ covariates; picture adapted from Efron et al. 2004.

SgS and LASSO seem to be very different methods from LARS, however, it can be shown (Efron et al., 2004) that by slightly modifying the LARS algorithm, the so called *Stagewise* and *LASSO Modification*, the SgS and the LASSO can be implemented exactly as a LARS procedure.

In the next subsection we will show how to compute the equiangular versor. Details on how to stop the procedure when moving on the equiangular versor can be found in Efron et al. (2004).

Actually, in Fig. 3.1, the staircase indicates the path for the Stagewise Linear Regression. An ideal SgS procedure with $\epsilon \to 0$ will collapse on the LARS solution.

The Equiangular Versor

In point (3) of Algorithm 3 it is required to determine the equiangular direction of the vector belonging to the active set of regressors already entered in the model. To this aim, let $\mathcal{A}$ be the set of subscripts of the variables already selected by the LARS procedure. The following computation is taken from Khan et al. (2007). Let $X_{\mathcal{A}} = \left(\ldots s_l x^l \ldots\right)$, $l \in \mathcal{A}$, where s_l is the sign of x^l as it enters in the model and $B_{\mathcal{A}}$ the equiangular versor. $B_{\mathcal{A}}$ has to satisfy three conditions.

1. $B_{\mathcal{A}}$ has to be a linear combination of the vectors in the active set.
$$B_{\mathcal{A}} = X_{\mathcal{A}} w_{\mathcal{A}}, \tag{3.1}$$
where $w_{\mathcal{A}}$ is a vector of weights to be determined;
2. $B_{\mathcal{A}}$ has unit variance:
$$\frac{1}{n} B_{\mathcal{A}}^T B_{\mathcal{A}} = 1. \tag{3.2}$$
3. $B_{\mathcal{A}}$ has equal correlation with each of the active predictors. Let a be the value of correlation of $B_{\mathcal{A}}$ with each one of the variable in the active set, we have
$$\frac{1}{n} X_{\mathcal{A}}^T B_{\mathcal{A}} = a\mathbf{1}_{\mathcal{A}}, \tag{3.3}$$
where $\mathbf{1}_{\mathcal{A}}$ is a vector of dimension $|\mathcal{A}|$ of ones.

Using (3.1) in (3.2) we have
$$\frac{1}{n} w'_{\mathcal{A}} X_{\mathcal{A}}^T X_{\mathcal{A}} w_{\mathcal{A}} = 1,$$
that can be expressed as
$$w_{\mathcal{A}}^T R_{\mathcal{A}} w_{\mathcal{A}} = 1, \tag{3.4}$$

where $R_{\mathcal{A}} = \frac{X_{\mathcal{A}}^T X_{\mathcal{A}}}{n}$ is the correlation matrix of the active variables. Using (3.1) in (3.3) we have

$$R_{\mathcal{A}} w_{\mathcal{A}} = a\mathbf{1}_{\mathcal{A}},$$

so the weight vector can be expressed as

$$w_{\mathcal{A}} = a\left(R_{\mathcal{A}}\right)^{-1}\mathbf{1}_{\mathcal{A}}. \tag{3.5}$$

The matrix $R_{\mathcal{A}}$ can be expressed as

$$R_{\mathcal{A}} = D_{\mathcal{A}} R_{\mathcal{A}}^{-} D_{\mathcal{A}}$$

where $D_{\mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$ is the diagonal matrix

$$D_{\mathcal{A}} = \begin{bmatrix} s_1 & & & \\ & s_2 & & \\ & & \ddots & \\ & & & s_{|\mathcal{A}|} \end{bmatrix}$$

and $R_{\mathcal{A}}^{-}$ is the correlation matrix of the unsigned active predictors. From (3.5) we have

$$w_{\mathcal{A}} = a\left(R_{\mathcal{A}}\right)^{-1}\mathbf{1}_{\mathcal{A}}. \tag{3.6}$$

Using (3.5) into (3.4) we have that

$$a = \left[\mathbf{1}_{\mathcal{A}}^T\left(R_{\mathcal{A}}\right)^{-1}\mathbf{1}_{\mathcal{A}}\right]^{-0.5}. \tag{3.7}$$

Now the correlation a_j of an inactive covariate x_j with $B_{\mathcal{A}}$ can be expressed as

$$a_j = \frac{1}{n} \tag{3.8}$$

The equiangular vector can be expressed as

$$B_{\mathcal{A}} = X_{\mathcal{A}}\left[\mathbf{1}_{\mathcal{A}}^T\left(R_{\mathcal{A}}\right)^{-1}\mathbf{1}_{\mathcal{A}}\right]^{-0.5}\left(R_{\mathcal{A}}\right)^{-1}\mathbf{1}_{\mathcal{A}}. \tag{3.9}$$

Updating the Model

Once the equiangular versor $B_{\mathcal{A}}$ is determined we have the direction upon which we have to improve our estimation but we don't know how much to move along that direction.

Suppose $\hat{\mu}_{\mathcal{A}}$ is the current estimate of the LARS algorithm: the vector of the actual

correlation of predictors with the actual residual is

$$\hat{c} = X^T r = X^T (y - \hat{\mu}_{\mathcal{A}}) = \begin{bmatrix} \hat{c}_1 \\ \hat{c}_2 \\ \vdots \\ \hat{c}_p \end{bmatrix}.$$

We can now characterize the active set as

$$\mathcal{A} = \left\{ i : |\hat{c}_i| = \max_{j=1,\ldots,p} |\hat{c}_j| = \hat{C} \right\},$$

because the actual set of predictors are the ones with equal and maximum correlation with the actual residual. The next step of the LARS algorithm updates the estimation $\hat{\mu}_{\mathcal{A}}$ in the direction of the equiangular vector:

$$\hat{\mu}^+_{\mathcal{A}} = \hat{\mu}_{\mathcal{A}} + \lambda B_{\mathcal{A}}. \tag{3.10}$$

How to determine λ? For every $j \in \mathcal{A}^c$ we compute

$$\lambda^+ = \frac{\hat{C} - \hat{c}_j}{a - a_j} \quad \text{and} \quad \lambda^- = \frac{\hat{C} + \hat{c}_j}{a + a_j}, \tag{3.11}$$

where $a_j = x'_j B_{\mathcal{A}}$ is the correlation of x_j and $B_{\mathcal{A}}$; we then choose in (3.10)

$$\lambda = \min^+_{j \in \mathcal{A}^c} \left\{ \lambda^-, \lambda^+ \right\}, \tag{3.12}$$

where $\min^+$ indicates that the minimum is taken over only positive components within each choice of j.

To prove that equations (3.11)-(3.12) are correct we define

$$\hat{\beta}(\lambda) = \hat{\mu}_{\mathcal{A}} + \lambda B_{\mathcal{A}}; \tag{3.13}$$

the current correlation depending on λ is

$$c_j(\lambda) = x^{jT} \left(y - \hat{\beta}(\lambda) \right) = \hat{c}_j - \lambda a_j; \tag{3.14}$$

we have in fact that, for $j \in \mathcal{A}$,

$$|c_j(\lambda)| = \hat{C} - \lambda a, \tag{3.15}$$

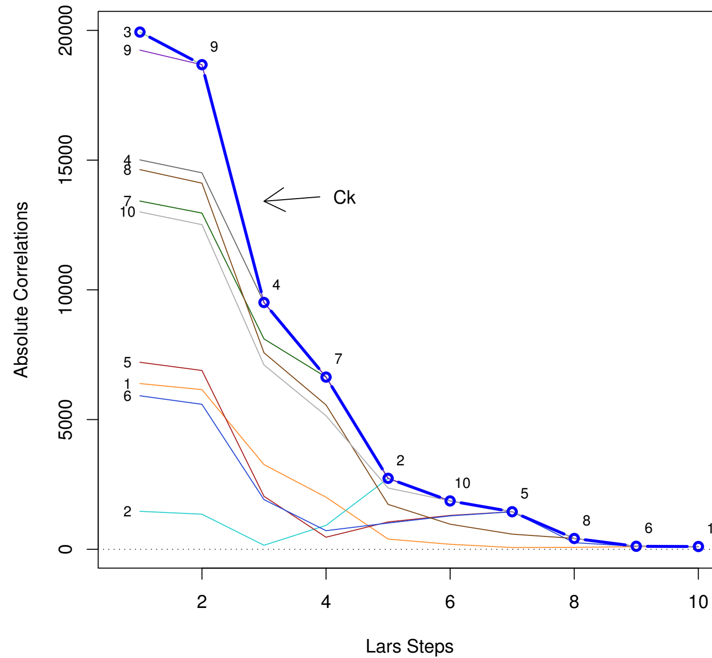


Figure 3.2: Absolute current correlation as function of the LARS step; it can be seen how maximum current correlation decreases with k and that once in the active set all the variable correlation with the residual decrease in the same way (as expected) (Picture adapted from Efron et al. 2004).

showing that all absolute current correlations decay in the same way.

In Fig. 3.2 it is shown the evolution of single input correlation for the Diabetes data problem used in Section 2.1: it can be seen how correlation of input in the active dataset decreases equally and how the maximum correlation is always decreasing.

LARS and LASSO are strongly connected techniques: LASSO solutions can be obtained with a modification of the LARS algorithm. For more details on the LASSO-LARS relations refer to Efron et al. 2004.

4 Filtering and Prediction

Filtering algorithms use series of measurements observed over time corrupted by noise and produce estimates of the real state of the variables in exam that tend to be more precise than those that would be based on the single measurements alone. More formally, filtering techniques operate recursively on streams of noisy input data to produce a statistically optimal estimate of the underlying system state. The filtering step goes with a prediction of next state or states of the variable in exam.

Filtering and prediction techniques will be exploited in a PdM user case presented in Chapter 9.

The Chapter is organized as follow: Section 4.1 provides a Bayesian formalization of the filtering and prediction problem. The most famous approach to Filtering and Prediction, the Kalman Filter, is presented in Section 4.2, while Monte Carlo Sequential methods are briefly introduced in 4.3. Finally, in Section 4.4, Kernel Density Estimators are presented.

4.1 Recursive Bayesian Estimation

Consider the following dynamical system in state space form

$$\begin{cases} z_{k+1} &= f(z_k) + v_k \\ y_k &= h(z_k) + w_k \end{cases} \tag{4.1}$$

where z_k is the state variable, y_k represents the noisy measurement of z_k, v_k and w_k are respectively the model and output noise.

From a Bayesian perspective the problem we are facing is to estimate the conditional probability density distribution

$$p(z_{k+1}|y_{k:1}) \tag{4.2}$$

where $z(\cdot)$ is the hidden state of the system defined in 4.1, $y(\cdot)$ is the noisy measure of the output $h(z_k)$ and the notation $y_{k:1}$ indicates $y_{k:1} = \{y_1, y_2, \ldots, y_k\}$: we are looking for a recursive probabilistic description with some degree of belief on the hidden state $z(\cdot)$ at next time $k+1$ based on the knowledge of its past noisy measures. If the initial distribution $p(z_0|y_0)$ is known than theoretically is possible to compute (4.2) in two different stages: *a priori prediction* and *a posteriori update*.

Assume that the process given in (4.1) is Markovian of order one, so that

$$p(z_{k+1}|z_k, y_{1:k}) = p(z_{k+1}|z_k).$$

Then, the a priori prediction can be computed as

$$p(z_{k+1}|y_{1:k}) = \int p(z_{k+1}|z_{1:k})p(z_k|y_{1:k})dz_k \tag{4.3}$$

If the conditional probability $p(z_{k+1}|z_{1:k})$ is available, then the a posteriori update can be computed by exploiting the Bayes' law as

$$p(z_k|y_{1:k}) = \frac{p(y_k|z_k)p(z_k|y_{k-1})}{p(y_k|y_{1:k-1})}, \tag{4.4}$$

where

$$p(y_k|y_{1:k-1}) = \int p(y_k|z_k)p(z_k|y_{1:k-1})dz_k. \tag{4.5}$$

Equation (4.5) is based on a probabilistic description of the measurements $p(y_k|z_k)$. Equations (4.3)-(4.4) represent the optimal Bayesian solution to the prediction problem. In general such solution cannot be computed analytically and a Monte Carlo approach has to be used (Section 4.3), however, in the case of Gaussian assumptions on noise

distributions, a closed form solution can be computed via the Kalman Predictor (Section 4.2).

In the following we are going to consider a linear, state-space represented version of system 4.1:

$$\begin{cases} z_{k+1} &= Az_k + Bu_k + v_k \\ y_k &= Cz_k + w_k \end{cases} \tag{4.6}$$

where we also allow the presence of an external input u in the state equation, v_k is the model noise, $v_k \sim g(x)$, $g(x)$ is continuous probability density function (pdf) on $\mathbb{R}$. Based on the distribution class $g(x)$ belongs to, different approaches for filtering and prediction can be adopted as it will be shown in Sections 4.2 and 4.3.

4.2 Kalman Predictor

If $g(x)$ is Gaussian, we can derive a closed-form explicit prediction of $x_{k+1|k}$ by using a Kalman Predictor (Kalman, 1960), that is the best linear predictor in terms of minimum error variance.

Assuming $v_k \sim \mathcal{N}(0, Q)$, then the optimal linear, minimum variance predictor $\hat{z}_{k+1|k}$ of the state z_{k+1} of (9.4)-(9.5) is the Kalman Predictor described by the following set of equations:

$$\hat{z}_{k+1|k} = A\hat{z}_k + K_k \left[Y_k - C\hat{z}_{k|k-1} \right] + Bu_k \tag{4.7}$$

$$K_k = AP_{k|k-1}C'\Lambda_k^{-1} \tag{4.8}$$

$$\Lambda_k = CP_{k|k-1}C' + R, \tag{4.9}$$

where $P_{k+1|k}$ is the variance matrix of the prediction error $\tilde{z}_{k+1|k} = \hat{z}_{k+1|k} - z_{k+1}$:

$$P_{k+1|k} = \mathbb{E}\left[\tilde{z}_{k+1|k}\tilde{z}'_{k+1|k} \right].$$

The matrix $P_{k+1|k}$ is updated through the Discrete Riccati Equation

$$P_{k+1|k} = \tilde{Q} + A\left[P_{k|k-1} - P_{k|k-1}C'\Lambda_k^{-1}CP_{k|k-1} \right] A',$$

where

$$\tilde{Q} = \begin{bmatrix} Q & 0 \\ 0 & 0 \end{bmatrix}.$$

The tuning of $\tilde{Q}$ and R can be done by computing a test on the residuals correlation

(Ljung, 1999)

$$\mathcal{R}_{\tilde{Q},R}(\sigma) = \mathbb{E}\left[e_{\tilde{Q},R}(k)e_{\tilde{Q},R}(k+\sigma)\right], \tag{4.10}$$

with $e_{\tilde{Q},R}(k) = Y(k) - C\widehat{z}_{k|k}$ where the estimation $\widehat{z}_{k|k}$ depends on the choice of $\tilde{Q}$ and R. A grid search on different set of values of $\tilde{Q}$ and R can be performed to minimize

$$\max_{\sigma>0}\left|\mathcal{R}_{\tilde{Q},R}(\sigma)\right|. \tag{4.11}$$

The Kalman Predictor provides both a prediction of the next value of the state z_k and a distribution of the estimation error.

4.3 Particle Filter

If $g(x)$ is not Gaussian, the Kalman Predictor is no more optimal, and other techniques are to be preferred. Given the linear model (9.4)-(9.5) and the estimation of g provided by the Gaussian KDE, $\hat{g}(x,\hat{\gamma^*})$, one can compute the best prediction of z_{k+1} at discrete time k as

$$\hat{z}_{k+1|k} = A\hat{z}_{k|k} + G\mathbb{E}[\widetilde{v}], \tag{4.12}$$

where $\widetilde{v} \sim \hat{g}(x,\hat{\gamma^*})$.

As far as the filtering step is concerned, the a posteriori estimation of $\hat{z}_{k|k}$ given the measurements $Y_{1:k}$ can be obtained by using Sequential Monte Carlo Methods (SMCM), or *Particle Filters*, (Liu and Chen, 1998; Arulampalam, Maskell, Gordon, and Clapp, 2002). Such methods provide suboptimal filtering algorithms that can be exploited when the noise distributions are not Gaussian.

In the SMCM approach, the a posteriori density function is represented by using a set of N_P random samples (particles) with associated weights. The estimates are then computed based on such particles and weights by averaging. As the number N_P increases, the estimates converge to the real state. The main drawback of such approach is its high computational complexity.

Let $\{x_i^j\}_{j=1}^{N_P}$ for $i = 0,\ldots,k$ be the set of particles and $\{w_i^j\}_{j=1}^{N_P}$ the associated weights such that $\sum_{j=1}^{N_P} w_i^j = 1$ for every $i = 0,\ldots,k$. The a posteriori distribution

$$p(x_k|Y_{1:k}) \approx \sum_{j=1}^{N_P} w_k^j \delta(x_k - x_k^j),$$

where δ is the Dirac function, and the update equation for the weights is given by

$$w_k^j = w_{k-1}^j \frac{p(Y_k|x_k^j)p(x_k^j|x_{k-1}^j)}{q(x_k|x_{0:k-1}, Y_{1:k})}, \tag{4.13}$$

where $q(\cdot)$ is a proposal distribution called *importance density* (Arulampalam et al., 2002). A simple choice for $q(x_k|x_{0:k-1}, Y_{1:k})$ is the a priori distribution of the state $p(x_k|x_{k-1})$. According to such choice, (4.13) becomes

$$w_k^j = w_{k-1}^j p(Y_k|x_k^j)$$

The particle filtering algorithm is sketched in Algorithm 4 (Douchet, deFreitas, and Gordon, 2001). The algorithm mainly consists in a recursive propagation of the particles and the associated weights.

Algorithm 4: Sequential Importance Sampling

Data: $\{x_{k-1}^j, w_{k-1}^j\}_{j=1}^{N_P}$, the new measure Y_k
Result: $\{x_k^j, w_k^j\}_{j=1}^{N_P}$
for $j = 1, \ldots, N_P$ **do**

- Draw x_k^j according to the distribution $q(x_k|x_{k-1}^j, Y_k)$;
- Update the weights according to (4.13);

- Compute the total weight $w_{TOT} = \sum_{j=1}^{N_P} w_k^j$;

for $j = 1, \ldots, N_P$ **do**

- Normalize the weights $w_k^j = w_{TOT}^{-1} w_k^j$.

A critical issue of Algorithm 4 is that of degeneracy. After some iterations, one of the weight usually becomes equal to one while the others go to zero, so that only one particle provides support for the estimation. A measure of degeneracy is given by the effective sample size

$$N_{\text{eff}}(k) = \frac{N_P}{1 + \text{Var}(w_k^{*j})}, \tag{4.14}$$

where w_k^{*j} are referred as the true weights (Arulampalam et al., 2002)

$$w_k^{*j} = \frac{p(x_k^j|Y_{1:k})}{q(x_k^j|x_{1:k-1}^j, Y_k)}.$$

An estimate of (4.14) is given by

$$\widehat{N_{\text{eff}}}(k) = \frac{1}{\sum_{j=1}^{N_P} (w_k^j)^2}. \tag{4.15}$$

N_{eff} can be seen as a measure of statistical significance of the particles. If $N_{\text{eff}}(k) < N_{\text{thr}}$, a given threshold value, then the statistical significance of the set is increased by performing the so-called resampling. There are several ways to perform resampling. In Algorithm 5, a systematic approach is described where particles associated with 'small' weights are eliminated from the set while more samples are taken from those support points associated with 'large' weights.

Algorithm 5: Systematic Resampling

Data: $\{x_k^j, w_k^j\}_{j=1}^{N_P}$
Result: New support points $\{\bar{x}_k^j, \bar{w}_k^j\}_{j=1}^{N_P}$
- Let $c_1 = 0$;
for $j = 2 : N_P$ **do**
  $c_j = c_{j-1} + w_k^j$;
- Set $j = 1$;
- Draw u_1 from the uniform distribution $\mathbb{U}[0, N_P^{-1}]$;
for $m = 1 : N_P$ **do**
  - $u_m = u_1 + N_P^{-1}(m-1)$;
  while $u_m > c_j$ **do**
    $i = i + 1$;
  - $\bar{x}_k^m = x_k^j$;
  - $\bar{w}_k^m = N_P^{-1}$.

4.4 Kernel Density Estimation

The distribution of $g(x)$ is generally unknown and must be estimated from the data in order to understand which approaches can be allowed for filtering. Gaussian Kernel Density Estimators (KDEs) may be used for to fulfill this task.

In the Gaussian KDE approach, the estimate $\hat{g}(x,\gamma)$ of $g(x)$ has the form

$$\hat{g}(x,\gamma) = \frac{1}{N}\sum_{i=1}^{N} K_\gamma(x, X_i), \qquad x \in \mathbb{R}, \tag{4.16}$$

where $\{X_i\}_{i=1,\cdots,N}$ are N independent realizations of $g(x)$, and

$$K_\gamma(x, X_i) = \frac{1}{\sqrt{2\pi\gamma}} e^{-\frac{(x-X_i)^2}{2\gamma}} \tag{4.17}$$

is the Gaussian Kernel function with *location* X_i and *bandwidth* $\sqrt{\gamma}$. The performance of the Gaussian KDE strongly depends on the choice of the bandwidth. The value of γ is usually chosen to minimizes the Mean Integrated Squared Error (MISE) Jones, Marron, and Sheater (1996)

$$\begin{aligned} \text{MISE}\,[\hat{g}]\,(\gamma) &= \mathbb{E} \int [\hat{g}(x,\gamma) - g(x)]^2\, dx \\ &= \int \underbrace{(\mathbb{E}\,[\hat{g}(x,\gamma)] - g(x))^2}_{\text{bias of } \hat{g}} + \underbrace{\text{Var}\,[\hat{g}(x,\gamma)]}_{\text{variance of } \hat{g}}\, dx. \end{aligned} \tag{4.18}$$

It is shown in Sheater and Jones (1991) that the value of γ that minimizes the MISE is the same that minimize its first-order asymptotic approximation, called AMISE,

$$\text{AMISE}\,[\hat{g}]\,(\gamma) = \frac{1}{4}\gamma^2 \left\|g''\right\|^2 + \frac{1}{2N\sqrt{\pi\gamma}}, \quad N \to \infty \tag{4.19}$$

where $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}$. It can be shown that the optimal choice of γ is given by

$$\gamma^* = \left(\frac{1}{2N\sqrt{\pi}\left\|g''\right\|^2}\right)^{2/5}. \tag{4.20}$$

Critical for the computation of γ^* is the fact that equation (4.20) depends on the functional $\|g''\|^2$. The classical approach to estimate (4.20) is the so-called l-stage direct plug-in bandwidth selector Sheater and Jones (1991); Wand and Jones (1995). In this approach $\left\|g^{(l+2)}\right\|$ is computed for some $l > 0$ assuming that the true g is Gaussian. If g is far from being Gaussian, this approach yields bad estimated of γ^*. A more effective approach to compute (4.20) has been proposed in Botev, Grotowski, and Kroese (2010) and is briefly described in the following.

Equation (4.20) requires the computation of the functional $||g''||^2$. For any positive j the following relationship holds

$$||g^{(j)}||^2 = (-1)^j \mathbb{E}_g[g^{(2j)}(X)]. \tag{4.21}$$

Equation (4.21) can be used to compute an estimation of $||g^{(j)}||^2$. Given the Gaussian

KDE in (4.16)-(4.17), the estimator of $(-1)^j \mathbb{E}_g[g^{(2j)}(X)]$ has the following form

$$
\begin{aligned}
(-1)^j \widehat{\mathbb{E}_g[g^{(2j)}]} &:= \frac{1}{N} \sum_{m=0}^{N} \hat{g}^{(2j)}\left(X_m, \gamma_j\right) \\
&= \frac{1}{N^2} \sum_{m=0}^{N} \sum_{i=0}^{N} K_{\gamma_j}^{(2j)}(X_m, X_i), \qquad (4.22)
\end{aligned}
$$

whereas $||g^{(j)}||^2$ can be estimated as

$$
\begin{aligned}
||\widehat{g^{(j)}}||^2 &:= ||\hat{g}^{(j)}(\cdot, \gamma)||^2 \\
&= \frac{1}{N^2} \sum_{m=0}^{N} \sum_{i=0}^{N} \int_{\mathbb{R}} K_{\gamma_j}^{(j)}(x, X_m) K_{\gamma_j}^{(j)}(x, X_i) dx \\
&= \frac{1}{N^2} \sum_{m=0}^{N} \sum_{i=0}^{N} K_{2\gamma_j}^{(2j)}(X_m, X_i). \qquad (4.23)
\end{aligned}
$$

To derive (4.23), we exploited the fact that the kernel function K satisfies the Chapman-Kolmogorov equation

$$
\int_{\mathbb{R}} K_{\gamma_1}(x_1, x_0) K_{\gamma_2}(x_2, x_1) dx = K_{\gamma_1+\gamma_2}(x_2, x_0).
$$

Estimators (4.22) and (4.23) are equal when (4.22) is evaluated at $2\gamma_j$. Both (4.22) and (4.23) estimate the same quantity ($||g^{(j)}||^2$). Then, we can choose γ_j^* such that (4.22) and (4.23) have the same asymptotic mean square error (for $N \to \infty$). It can be proved that this is achieved when

$$
\gamma_j^* = \left(\frac{1 + 1/2^{j+\frac{1}{2}}}{3} \frac{1 \times 3 \times \ldots \times (2j-1)}{N\sqrt{\pi/2}||g^{(j+1)}||^2} \right)^{\frac{2}{3+2j}}. \qquad (4.24)
$$

For further details see Botev et al. (2010).

Using (4.24), γ_j^* can be estimated by computing via (4.23) an estimate of $||g^{(j+1)}||^2$. Clearly, to estimate $||g^{(j+1)}||^2$ an estimate of γ_{j+1}^* is required. As a consequence, the computation of the (infinite) sequence $\gamma_{j+1}^*, \gamma_{j+2}^*, \gamma_{j+3}^*, \ldots$ is also required. On the other hand, once an estimate of γ_{l+1}^* for some $l > 0$ is available, one can recursively compute estimates of $\gamma_l^*, \gamma_{l-1}^*, \ldots, \gamma_1^*$. This fact can be described by stating that

$$
\gamma_j^* = \lambda_j(\gamma_{j+1}^*),
$$

for some function λ_j, and, more generally, by using (4.20) and (4.24), that

$$\gamma^* = \mu\gamma_1^* = \ldots = \mu\lambda^{[l]}(\gamma_{1+l}^*) \tag{4.25}$$

with

$$\mu = \left(\frac{6\sqrt{2}-3}{7}\right)^{2/5} \approx 0.9.$$

As stated before, the typical approach to compute (4.25) is the l-stage direct plug-in bandwidth selector, which can provide bad estimates when g is far from being Gaussian. To avoid this drawback a different approach has been proposed in Botev et al. (2010). Instead of recursively solving (4.25) we find a solution of the non-linear equation

$$\gamma = \mu\lambda^{[l]}(\gamma) \tag{4.26}$$

where l is an integer, using fixed point iteration or Newton's method. The procedure is described in Algorithm 6.

Algorithm 6: Optimal Bandwidth Selection

Data: Nrealizations of X, $l > 2$, $\lambda(\cdot)$.
Result: Gaussian KDE with optimal choice of γ in terms on AMISE
1) Initialize $\gamma_0 = \epsilon$, where ϵ is the machine precision,
and $z = 0$ (algorithm iteration).
2) $\gamma_{z+1} = \mu\lambda^{[l]}(\gamma_z)$.
if $|\gamma_{z+1} - \gamma_z| < \epsilon$ **then**
  $\hat{\gamma}^* = \gamma_{z+1}$, STOP
else
  $z = z + 1$, REPEAT from 2).
3) Gaussian KDE according to (4.16), (4.17) evaluated
at $\hat{\gamma}^*$.

From the point of view of the implementation, as suggested in Botev et al. (2010), there are no meaningful gains in setting l above 5. A free MATLAB implementation of Algorithm 6 is available at Botev (2012).

Part III

Virtual Metrology

5

A VM Case Study for Chemical Vapor Deposition (CVD) Modeling

The results presented in this Chapter were obtained in collaboration with Infineon Technologies Austria, AG and are adapted from Susto and Beghi (2012b,a) and Susto and Beghi (2012c).

5.1 Introduction to Virtual Metrology and Applications

In semiconductor manufacturing, state of the art for wafer quality control relies on product monitoring and feedback control loops; the involved metrology operations are particularly cost-intensive and time-consuming. For this reason, it is a common practice to measure a small subset of a productive lot and devote it to represent the whole lot. Virtual Metrology (VM) methodologies are able to obtain reliable predictions of metrology results at process time; this goal is usually achieved by means of statistical

models, linking process data and context information to target measurements.

The research on VM technologies, as partly introduced in Section 1.3, has been intensively developed in the past recent years, given the dramatic search of semiconductor manufacturers for increased process capabilities and reduced costs described in Chapter 1 and linked to the promises of measurement cost reduction and improvements in production quality (by means of controllers able to handle VM information). From this point of view, VM tools are seen as information providers, able to yield probabilistic information about wafer quality at process time: intelligent tools such as controllers (Chen, Wu, Lin, Ko, Lo, Wang, Yu, and Liang, 2005), dispatching systems and sampling tools (Kurz, Kaspar, and Pilz, 2011) can take advantage of such information to improve the overall process quality. It is apparent that, in order to safely use a Virtual Metrology tool on line, its prediction accuracy must be as high as possible. This goal is usually achieved by means of statistical modelling and machine learning techniques able to find and exploit links between cost-free data (e.g. sensors data, logistic and recipe information) and target measurements (Pampuri et al., 2011b). The results of these algorithms is a model that defines the relationships between process data (input) and metrology data (output), and it is usable for prediction of incoming wafers measurements (Lynn et al., 2009; Susto, Beghi, and DeLuca, 2011b).

On the mathematical point of view, the VM problems are regression ones (Section 2.1). The problem of modeling semiconductor processes has been approached by using different techniques, such as Linear, like Ordinary Least Square (OLS) and Partial Least Squares (PLS), and Non-Linear, such as Artificial Neural Networks (NNs). Also Information Theoretic based approaches have been proposed (Schirru, Pampuri, DeLuca, and DeNicolao, 2012a). It has been shown (Hung et al., 2007; Kang et al., 2009; Lynn et al., 2009; Himmel, Kim, and May, 1992; Himmel and May, 1993) that NNs guarantee better performance in modeling semiconductor manufacturing processes than other linear approaches. NNs are flexible computing frameworks and universal approximators that can be applied to a wide range of learning problems with a high degree of accuracy (Khashei and Bijari, 2010). A common and widely adopted type of NN is the *Multilayer Perceptron (MLP)*; the central idea of MLPs is to extract linear combination of the inputs (in the problem considered here, the tool and logistic data) and then model the target (the critical dimension to be estimated) as a nonlinear function of such features (Besnard and Toprac, 2006). However, NNs can be really hard to train in learning problems with high dimensionality, as is the case in semiconductor manufacturing modeling. Moreover, given the use of non-linear features of the inputs during the algorithm training, the results often lack interpretability.

As stated in Section 1.3, besides high prediction accuracy, desirable properties of an efficient VM system are:

- *reasonably low computational times*;
- *interpretability.*

The VM modeling must take into consideration the two previous requirements, as it will be discussed in Chapter 5.

In Section 1.3 also several of the current challenges for VM systems have been listed:

1. high dimensionality;
2. data fragmentation;
3. multi processes modeling;
4. time series input data.

All of the previous issues will be dealt with in the next chapters of this Part of the thesis by providing modeling examples and innovative solutions.

This part of the thesis will be structured as follows:

- in this Chapter a VM system for predicting CVD thickness is proposed. Here we deal with issue 1. (high-dimensionality). Two basic approaches to deal with that have been proposed in the VM literature:
  - the use of *dimensionality reduction* techniques, like correlation analysis (Susto et al., 2011b; Cheng et al., 2008) and Principal Components Analysis (PCA) (Zeng and Spanos (2009)), that, when applied before the actual modeling part in a two-step approach, reduce the size of the dataset;
  - the use of *variable selection* techniques (Chapter 5), like Stepwise Selection (SS), where parsimonious models are created during the modeling phase.

  The results of variable selection techniques are usually easy to interpret, given the fact that only variables that matter 'enter' the model. However, as explained in Chapter 5, SS Regression, that has been widely adopted (Ferreira et al., 2009; Kang et al., 2009; Lynn et al., 2009; Ragnoli et al., 2009), is considered in the Statistical Learning community as a really 'greedy' approach where important variables may not enter the model due to the algorithm procedure. Usually *Stagewise Selection (SgS)* (Hastie et al., 2009) is preferred for prediction accuracy, but it is much more onerous from the computational point of view. We employ here another approach,

first presented for VM modeling in Susto and Beghi (2012b) and Susto and Beghi (2012c) based on Least Angle Regression (LARS) (Section 3.3) to provide equivalent solutions to SgS, but at the cost of SS.

In Chapter 5 also the issue 2. (data fragmentation) is tackled. As said, huge data fragmentation cannot be dealt with by considering separately every specific case, since there is insufficient data to identify and validate a reliable mathematical model for each product. It is therefore necessary to group together data collected under different equipment operating conditions. A smart data clustering (Susto and Beghi, 2012a) can enhance prediction accuracy and it is necessary to be able to model all the fab production. In Chapter 5 we employ an Information-Theory based approach to data clustering.

- In Chapter 6 a multi-step modeling (issue 3.) is presented. The results of this Chapter have been first presented in Pampuri, Schirru, Susto, DeNicolao, Beghi, and DeLuca (2012). Unfortunately, the modeling of multiple steps makes the dimensionality of the regression problem even bigger and for this reason research has not proceeded far in this direction. However, this is the next step in the VM research, given the fact that the variability of a process cannot be fully captured without looking at the wafer state that is related to the previous processing steps performed.

  In this Chapter also high-dimensionality is dealt with LASSO, while the problem of huge data fragmentation is dealt with *Multi-Task* to model the different logistic paths that a wafer may take.

- Chapter 7 is dedicated to deal with issue 4., modeling with time series input data.

  The methodology described in Chapter 7 and firstly presented in Schirru, Susto, Pampuri, and McLoone (2012b) is based on functional learning; the proposed Supervised Aggregative Feature Extraction (SAFE) approach allows continuous, smooth estimates of time series data to be derived (yielding aggregate local information), while simultaneously estimating a continuous shape function providing optimal predictions. To our knowledge, this in the first approach presented in the literature to deal with this VM issue.

Besides the listed issue for VM modeling, we also deal in Chapter 8 with the integration of a VM tool in a real industrial environment and its relation with R2R control.

Figure 5.1: A Plasma for Chemical-Vapor Deposition. Photo courtesy of Argonne National Laboratory `http://www.anl.gov/`

5.2 Introduction to VM for CVD

As already introduced, the main motivation for the research in VM is that monitoring physical properties of all wafers is crucial to maintain good yield and high quality standards, but is too costly; VM systems allow to partly overcome the lack of physical metrology when this is not performed for saving time and money. This is a common theme for many processes of the line and Chemical Vapor Deposition (CVD) (Fig. 5.1) is one of these: in this Chapter, a VM scheme that use tool data to predict, for every wafer, metrology measurements for CVD is presented.

As described in Section 1.2, CVD is the process of chemically growing a thin layer of of silicon over the wafer; this may be done through the use of plasmas (Fig. 5.2)[1]. The CVD process is the first in line in the semiconductor manufacturing sequence Khan et al. (2007); the processing of wafers defected in the first stages of the sequence, but not detected as such, clearly results in a waste of resources. It is therefore extremely important to monitor wafer quality in such early production stages.

The quality of a performed CVD process can be assessed by measuring the thickness pf the deposited layer and compare it with the desired target thickness. However, performing this kind of thickness measurements from every single wafer is hardly feasible due to the

[1]The interested reader is referred for more details to Konuma (1992) and Pierson (1992)

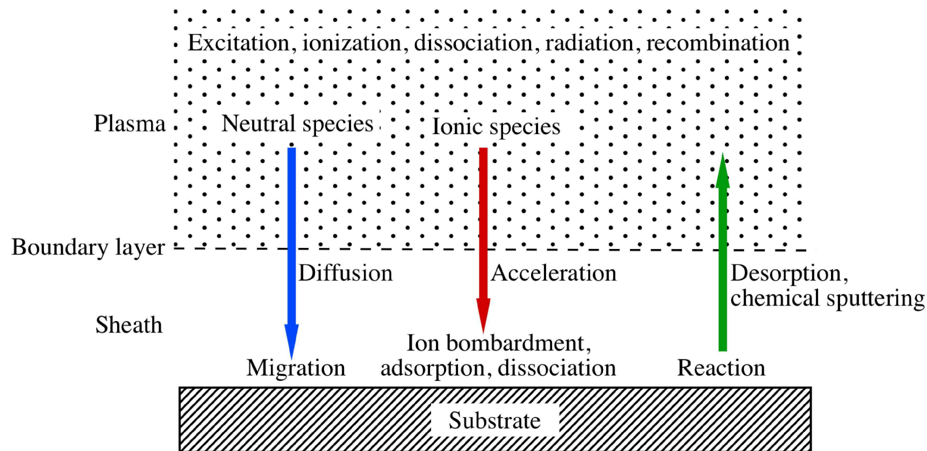


Figure 5.2: Reaction sequence in Plasma-enhanced CVD. Adapted from Barron (2009).

corresponding increase of costs and production time. For CVD processes, the current practice is to only monitor the thickness of few wafers in a lot; in the production setup considered in this Chapter, for instance, only 1 or 2 wafers out of 25 are actually measured. The VM module presented in this Chapter will address the problem of estimating wafer deposition thickness after the CVD has been performed; on the basis of the available metrology results and of the knowledge, for every wafer, of equipment variables, we will estimate the desired CVD thickness.

The prediction of the output of the CVD process (and for any other semiconductor process) is a challenging problem due to several factors, such as:

- hundreds of FDC variables are available, thus making a phase of variable selection crucial;
- for many variables, only statistics, instead of raw data, are available;
- data sets are often not complete, with thousands of missing values;
- high-mix production sets have to be considered, where several recipes are run on the equipment.

Furthermore, in the situation at hand, the equipment is composed of 3 different chambers (A, B and C) that exhibit different behaviors, and each chamber is divided into two sub-chambers (1 and 2) (the structure of the CVD equipment is shown in Fig. 5.3). Summarizing: the issues of high-dimensionality and data fragmentation introduced in Section 1.3 and 5.1 will be addresses.

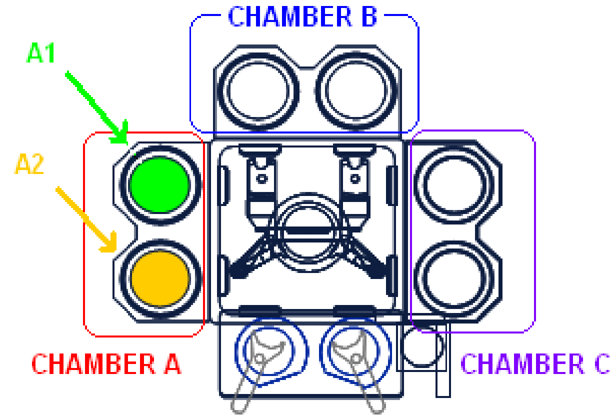


Figure 5.3: Scheme of the CVD equipment in exam: each machine has 3 separated chambers (A, B and C), each one of them divided into two sub-chambers (1 and 2).

The VM module proposed will be based on Least Angle Regression (LARS) to overcome the problem of high dimensionality and model interpretability; the LARS will be compared with other classical modeling approaches for VM.

To deal with the huge data fragmentation we cannot consider separately every specific case, since there are not data enough to identify and validate a confident mathematical model for each product. It is therefore necessary to group together data collected under different equipment operating conditions. A qualitative clustering approach is firstly proposed, in particular, a comparison between a VM system running on groups of data with the same targets and one obtained by considering the three chambers of the CVD equipment as separated machines is discussed. Then, a statical distance-based clustering approach is used for the modeling of the whole tool production and to deal with data fragmentation.

The proposed VM models have been tested on industrial production data sets.

The rest of the Chapter is organized as follows. In Section 5.3 a formalization of the problem and a description of the pre-processing issues are provided. Given the fact that modeling techniques employed have already been presented in Chapter 2, the experimental results are then provided in Section 5.4. Finally in Section 5.5 the proposed statistical clustering approach is presented.

5.3 Problem Formalization and Data Preprocessing

The problem in exam will be formalized as a typical regression one.

Assume that n observations (x_i, y_i) are available, where $x_i \in \mathbb{R}^p$ are the tool variables

value for process iteration i and $y_i \geq 0$ is the value of CVD thickness physically measured for i-th wafer. In the perspective of a typical regression problem, we suppose that there exists a relationship

$$y = f(x), \tag{5.1}$$

and we want to estimate $f(\cdot)$ from the set of observations $\{(x_i, y_i)\}_{i=1}^n$, so that CVD thickness can be estimated for those wafers for which y is not measured.

As stated, we will compare the performance of the LARS (Section 3.3) with other Variable Selection and the most popular approaches to regression (Chapters 3 and 2).

A semiconductor manufacturing process such as Chemical Vapor Deposition (CVD) can be described in terms of a large number, sometimes hundreds, of physical variables (pressures, flows, temperature, etc.). The reduction of the number of variables to take into consideration for process modeling is therefore a critical issue. The CVD tool considered in this work is equipped with a considerable number of sensors and more than one hundred statistical variables (means, variances, maximum and minimum values, etc.) have been collected from the machine. The selection of the variables that are most relevant for VM modeling purpose has been performed according to the following rules Hung et al. (2007):

- only one sensor is selected among those measuring the same physical property;
- sensors providing measures that are linear combination of those obtained by already selected sensors are discarded;
- variables taken during the so-called cleaning step of the process are omitted. In fact, for the cleaning parameters, only global information on the chamber is available, whereas, for modeling purpose, it is always required to be able to associate the variable values to a particular wafer;
- parameters that are collected for every (or almost every) wafer are selected, to reduce the number of missing values and guarantee data completeness.

After the selection of a sufficient amount of parameters according to the above mentioned rules, data are normalized to bring all the parameters to the same baseline.

Also, the effect of maintenances and cleanings on the tool should be taken into account. After a maintenance/big clean (called wet clean), patterns and uncommon behaviors can be noticed on the tool data. Usually, 1 lot is necessary before the tool is brought back to its normal statistical behavior. To improve the performance of the regression model, post-maintenance data should therefore not be included in the dataset. The best way to detect post-maintenance drifts is to have access to the maintenance database.

However, this is not always feasible and a systematic approach to remove such data from the set should be employed. Hotelling T^2 statistics Zeng and Spanos (2009) can be used to detect 'statistically abnormal' observations, however, *ad hoc* strategies can also be employed to detect changes in the dataset due to maintenances, such as monitoring the values of two key variables (related to the flows).

The most common approaches to dimensionality reduction are *Correlation Analysis* and *Principal Component Analysis (PCA)*; the two approaches are detailed in the following.

Correlation Analysis

Correlation analysis is performed to omit parameters bringing little information to the data set. For every couple of FDC parameters $\{(x^i, x^j)\}_{i,j=1,\dots,p}$, where x^i is a vector of dimension n with all the observations of the i-th regressor, the correlation is computed Lynn et al. (2009)

$$\phi_{x^i,x^j} = \frac{\text{cov}(x^i, x^j)}{\sigma_{x^i}\sigma_{x^j}} = \frac{E[(x^i - \mu_{x^i})(x^j - \mu_{x^j})]}{\sigma_{x^i}\sigma_{x^j}}, \tag{5.2}$$

where μ_{x^i}, μ_{x^j} and $\sigma_{x^i}, \sigma_{x^j}$ are respectively the means and the standard deviations of x^i, x^j. The closer the correlation is to 1 in absolute value, the more two parameters are correlated. Variables that are strongly correlated to others in the data set are then omitted.

Principal Component Analysis

PCA is a linear projection-based method that transform a set of uncorrelated variables into a new set of uncorrelated variables, named Principal Component (PS) Lynn et al. (2009); Rao (1964). PCA is run for a dataset defined by an $n \times p$ matrix X where the columns are l variables and the n columns are observations. X is written in terms of the $n \times l$ scores matrix T and the $p \times l$ loadings matrix P, plus a residual matrix E, as follows

$$X = TP^T + E \tag{5.3}$$

$$= \sum_{i=1}^{l} t_i p_i^T + E \tag{5.4}$$

where $t_i = Xp_i$. The vectors p_i are named Principal Components (PCs) Zeng and Spanos (2009); Hastie et al. (2009). PCs are arranged in such a way that the first

principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), therefore the first PC can be geometrically interpreted as the direction where most of the variability of X is explained and other PCs define directions where less and less variability is explained; furthermore each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding component.

By analyzing the magnitudes of the PCs, it is possible to employ only $l < p$ parameters to construct the model. PCA can be therefore considered a Dimensionality Reduction technique and prediction solutions are simpler and faster to be found, however working with PCs in the new feature space leads to losing the interpretability in the modeling. Given its dimensionality reduction property, PCA is widely adopted in VM systems, especially as a preliminary step before NN modeling Hung et al. (2007); Huang et al. (2008); Lynn et al. (2009); Khan et al. (2008); Kang et al. (2009); Zeng and Spanos (2009); Chou, Wu, and Chen (2010).

PCA is also employed for data distribution analysis: by visualizing the first 2/3 PCs, PCA is a useful tool to develop insights on distributions of high dimensional datasets. We will employ PCA for these reasons in Section 5.4.

Neural Networks Training

As stated in Section 2.3, NNs usually grant good prediction accuracy, however, several parameters need to be tuned. In particular, and with reference to the application at hand:

- *initial conditions* - a critical issue is the initialization of the weights. Typical choice for the starting values of the weights are random values near zero (see Hastie et al. (2009)). However, for some choice of the initial conditions, the training process leads the network to local optimum points. To overcome this issue, in the problem at hand every NN structure has been tested over a large number of different random initial conditions and evaluated in terms of MSE (2.26).

- *number of neurons* - while for the input and output layer the choice regarding the amount of neurons is strictly related to the number of inputs and outputs of the model, there is not a systematic way for deciding the number m_H of hidden nodes. As a guideline, it is known that a small value of m_H typically reflects into low prediction quality, whereas a large value of m_H leads to overfitting. To set a proper value of m_H, several configurations of NN have been tested. A widely used strategy is to choose $m_H \leq m_I$ (size of input layer L_{in}) and $m_H \geq m_O$ (size of output layer

L_{out}). In the particular situation at hand, we have that

- $m_I \le p$, number of inputs (that may be smaller than the number of regressors p if we apply some dimensionality reduction techniques before the modeling);
- $m_O = 1$, number of outputs (the CVD thickness);
- $1 \le m_H \le m_I$.

- *weight functions* - the typical functions $h_a(\cdot)$ and $h_b(\cdot)$ employed in NN are tan-sigmoids, arctan, radial-basis and linear functions. Depending on the choice of the type of function, the NN achieves different performances. A non-linear function between L_{in} and L_{hidden} has been adopted to try to describe the process nonlinearities, whereas a linear one between L_{hidden} and L_{out} has been chosen.

5.4 Experimental Results

Dimensionality Reduction

In this Section we apply some of the techniques previously described on a real fab dataset. All the data processing described in the present Section are made on a subset of the whole production dataset consisting of 10 different products for a total of $n = 6703$ observations. All data processing and algorithms implementation have been done in MATLAB. We remind that the output variable y is the deposition thickness of the CVD process. We suggested also the use of R as freeware alternative, for such environment several free implementation of statistical algorithms (i.e. Ridge Regression, LARS) are available on the web.

We first proceed by applying data reduction techniques. Application of the rules described in Section 5.3 results in the selection of $p_0 = 50$ variables to be used for model derivation. The list of the selected parameters is given in Table 5.1. In the list, heterogeneous physical parameters are included, such as flows, powers, temperatures, times, positions.

To further reduce the dataset size, correlation analysis is applied to the variables of Table 5.1. For every possible couple of variables $\{x^i, x^j\}$, $\forall i, j = 1 : p_0$, with $i \neq j$, we compute the correlation coefficients

$$\phi_{x^i,x^j} = \frac{\text{cov}(x^i, x^j)}{\sigma_{x^i}\sigma_{x^j}} = \frac{E[(x^i - \mu_{x^i})(x^j - \mu_{x^j})]}{\sigma_{x^i}\sigma_{x^j}}, \tag{5.5}$$

where μ_{x^i}, μ_{x^j} and $\sigma_{x^i}, \sigma_{x^j}$ are respectively the means and the standard deviations of x^i, x^j. In Fig. 5.4 the correlation matrix $|\Phi|$ is graphically represented, whose elements

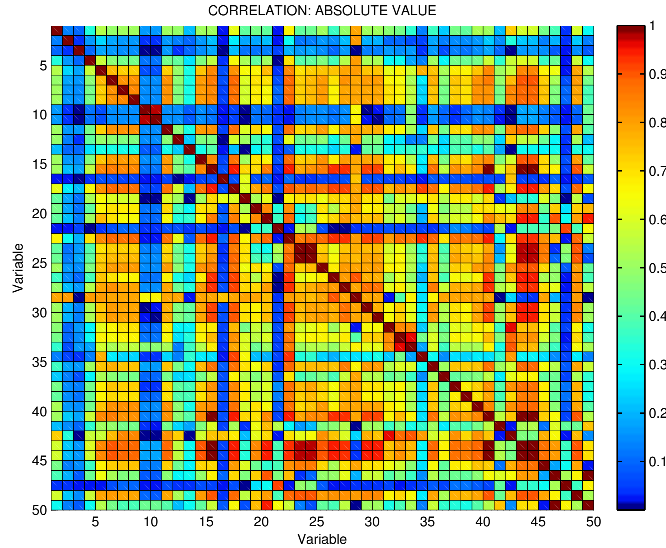


Figure 5.4: Visual representation of the correlation matrix $|\Phi|$

are the absolute values of the correlation coefficients:

$$|\Phi|(i,j) = |\phi_{x^i,x^j}|, \quad \forall i,j = 1,\cdots,n_0 \tag{5.6}$$

A threshold value of 0.99 has been used to consider two parameter as correlated enough so that only one of them is selected. By applying such criterion it is found that 9 couples of variables have a correlation (in absolute value) larger than 0.99; the reduction step leads to a set of $p_1 = 46$ variables, where those parameters that present less missing values are kept in the data set.

Finally, we also apply PCA on the group of p_1 variables, as described in Section 5.3. In Fig. 5.5 the cumulative variability explained by the first l PCs is shown. It can be seen that, with the new set of parameters, $l = 24$ regressors (the first l PCs) are enough to explain more than 99% of the process variability, a confidence limit sufficiently large to model the CVD process with l inputs only.

Qualitative Data Clustering

As introduced in Chapter 1, in current fab plants, high-mix semiconductor manufacturing processes are run. During several months of operation of a single tool, hundreds of different products (with different tool settings) are run. Moreover, the available dataset contains data regarding each of the 6 different chambers (A, B and C) and sub-chambers (1 and 2) of the CVD equipment. Also, in the situation at hand, several recipes are run

No.	Sensor	No.	Sensor	No.	Sensor
1	Divert Valve 1	18	Pressure 5	35	Throttle Valve 5
2	Divert Valve 2	19	Gas A Flow 1	36	Time 1
3	Divert Valve 3	20	Gas A Flow 2	37	Time 2
4	Divert Valve 4	21	Gas A Flow 3	38	Time 3
5	Divert Valve 5	22	Gas A Flow 4	39	Gas B Flow 1
6	He-H Flow	23	Gas A Flow 5	40	Gas B Flow 2
7	Heater Power 1	24	Gas A Flow 6	41	Gas B Flow 3
8	Heater Power 2	25	Temperature 1	42	Gas B Flow 4
9	Heater Power 3	26	Temperature 2	43	Gas B Flow 5
10	Heater Power 4	27	Temperature 3	44	Gas B Flow 6
11	Ozone Rate	28	Temperature 4	45	Gas C Flow 1
12	Ozone Flow 1	29	Temperature 5	46	Gas C Flow 2
13	Ozone Flow 2	30	Temperature 6	47	Gas C Flow 3
14	Pressure 1	31	Throttle Valve 1	48	Gas C Flow 4
15	Pressure 2	32	Throttle Valve 2	49	Gas C Flow 5
16	Pressure 3	33	Throttle Valve 3	50	Gas C Flow 6
17	Pressure 4	34	Throttle Valve 4		

Table 5.1: List of production variables selected.

on the same equipment. Recipes can be grouped by their thickness target value, namely, target 1, target 2, target 3 and target 4 in the considered dataset, where

$$\text{target } 1 \leq \text{target } 2 \leq \text{target } 3 \leq \text{target } 4. \tag{5.7}$$

As mentioned in Section 5.2, given the huge data fragmentation, it is not possible to separately consider each specific case, since there are not data enough to identify a confident mathematical model. A possible approach to deal with this issue is considering two different *qualitative clustering* of data, under similar, but not equal, processing and equipment operating conditions:

1. 3 groups of data sharing the same chamber;
2. 4 groups of data sharing the same thickness target.

Such choices, that are reasonably acceptable by a qualitative point of view, can also be supported by examining the first PCs of the FDC data.

In Fig. 5.6 the first two PCs of the input dataset are represented, where the six sub-chamber of the CVD machine are highlighted. The presence of three 'clouds' of data can be observed (similar to Gaussian bivariate distributions) corresponding to the three

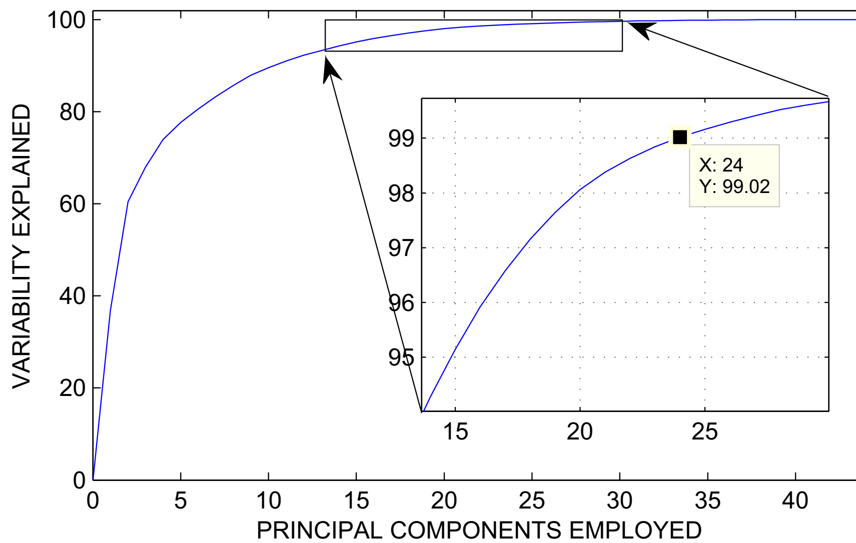


Figure 5.5: PCA: variability explained by the first l PCs.

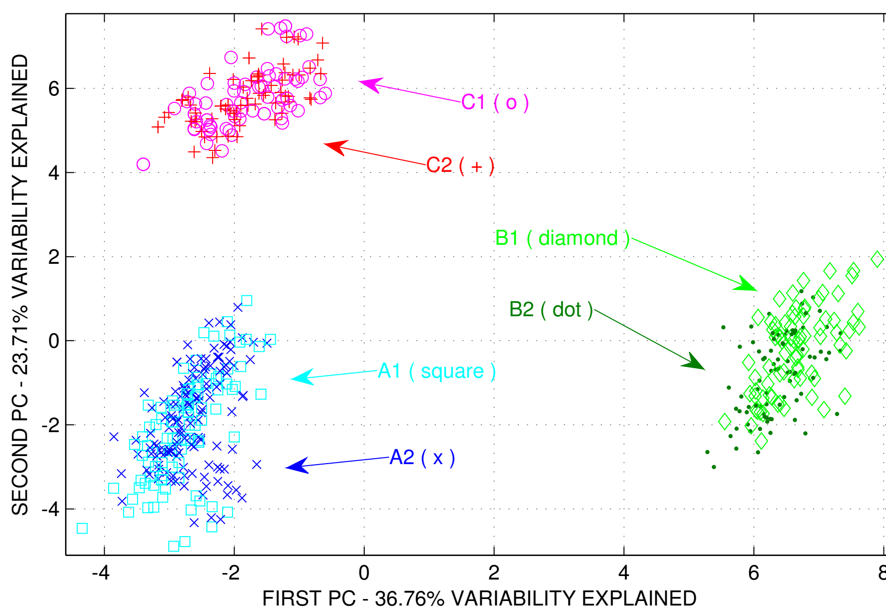


Figure 5.6: Qualitative Clustering: First two PCs, chambers highlighted.

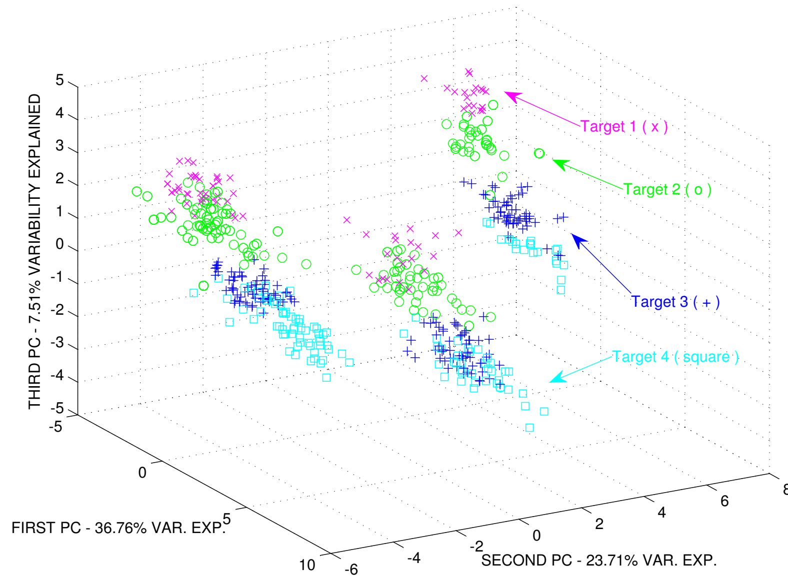


Figure 5.7: Qualitative Clustering: First three PCs, targets highlighted.

chambers A, B and C. It can also be seen that there is no clear separation inside a chamber between data regarding different sub-chambers ($A1$ and $A2$, $B1$ and $B2$, $C1$ and $C2$).

To appreciate a significant separation between data with different targets, the third PC has to be considered; the first two PCs describe more then 60% of the original dataset variability, while the first three sum up to almost the 70%. In Fig. 5.7 the original FDC data are projected onto the first three PCs. In this case, data with the same targets are not clearly grouped together as in the previous case, a separation of the data related to the chambers can still be observed, while separation related to the different targets can be appreciated in the third PC only.

Virtual Metrology Models

We first build up experiments to asses which amongst the modeling techniques previously presented performs better in terms of prediction accuracy, by computing the prediction error $e = y - \hat{y}$. The VM models has been tested on both types of clustering described in the previous Section. We indicate with CLA, CLB and CLC the clusters regarding chambers A, B and C and with CL$T1$, CL$T2$, CL$T3$ and CL$T4$ the ones regarding the targets. As said, we have considered the 10 products for which the largest amount of data are available, for a total of $n = 6703$ data, with $n_{\text{CL}A} = 2410$, $n_{\text{CL}B} = 2205$ and $n_{\text{CL}C} = 2088$ and $n_{\text{CL}T1} = 1341$, $n_{\text{CL}T2} = 1809$, $n_{\text{CL}T3} = 1686$ and $n_{\text{CL}T4} = 1867$.

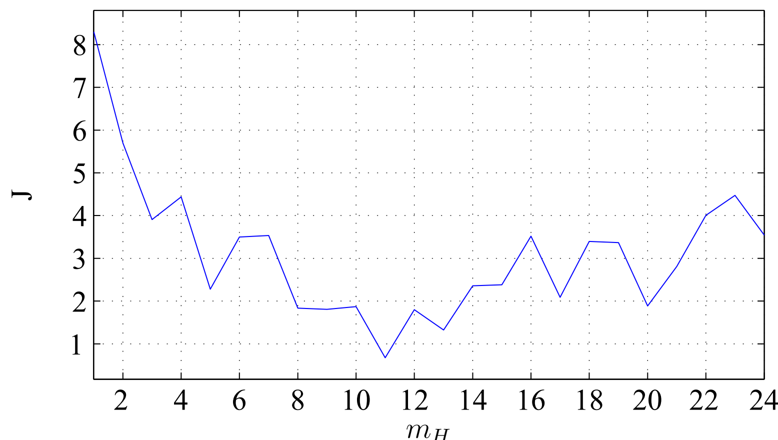


Figure 5.8: Average MSE on the validation dataset for NN with m_H hidden neurons.

The evaluation of methods' performances is done via Repeated Random Sub-Sampling Validation Picard and Cook (1984), also known as *Monte Carlo crossvalidation (MCCV)*, where M simulations are done by randomly splitting the n_{CL} observations into a training dataset of $\lfloor n_{CL}q \rfloor$ maintenance cycles and a validation dataset of $\lceil n_{CL}(1-q) \rceil$ maintenance cycle, with $0 < q < 1$. It has been shown Shao (1993) that MCCV is asymptotically consistent resulting in more pessimistic predictions of the test data compared with full crossvalidation.

To define the structure of the NN, a value for the size m_H of the hidden layer L_{hidden} has to be set. In order to choose m_H we have computed the MSE for 'vanilla' NNs with different values of m_H. $M = 1000$ simulations with 100 different initial conditions on each data cluster have been made and the average MSE is reported in Figure 5.8. It can be seen that the minimum is reached at $m_H = 11$, value that has therefore been chosen as hidden layer size. The same size m_H has been used for all the clusters in order to compare the same NN structure performances for the different groups of data.

The prediction error $e = y - \hat{y}$ distribution are represented in boxplots in Figure 5.9 for some of the different modeling techniques[2] described in Section 2 and 3. We have chosen two different kind of input sets, the $p_0 = 50$ initial regressors and the $l = 24$ principal components after correlation analysis and PCA. For the sake of conciseness, only the results for Cluster CLA are reported, however similar considerations apply for the other clusters. In our experiments we have chosen $q = 0.7$ and we have collected data for $M = 10000$ simulations. The stopping rule for SS has been based on F statistics, while, for comparison, we have chosen to stop the LARS algorithm when it reaches the same

[2]Due to the very large amount of data, outliers have not been depicted in Fig. 5.9 in order to improve readability.

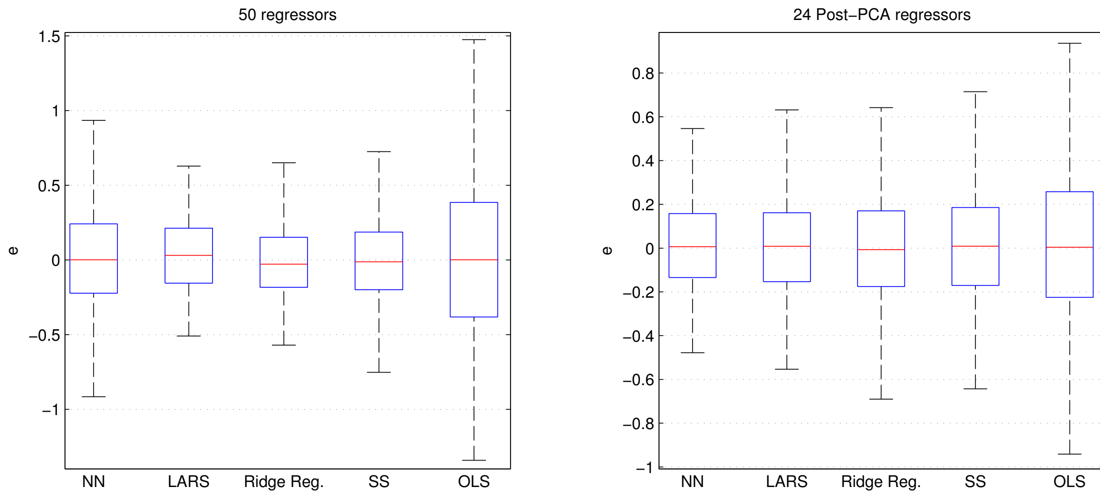


Figure 5.9: Boxplots of the prediction error $e = y - \hat{y}$ obtained with NN, LARS, Ridge Regression, Stepwise Selection and OLS for Cluster CLA with $p_0 = 50$ and with $l = 24$ regressors after PCA and correlation analysis.

number of predictors as SS. RR's λ (eq. 2.11) has been chosen as the one minimizing the MSE on the validation set over 100 simulations.

It can be noticed that:

- the performances of NNs and OLS can be greatly improved by the use of pre-processing techniques. No significant difference on the other hand comes from the use of pre-processing in LARS, RR and SS.
- OLS is outperformed by all the other linear methods;
- LARS provides better prediction accuracy than SS and it is the best methods amongst the ones that provides interpretable results;
- NNs, has already shown in several works of VM, is the method that guarantees the best prediction accuracy when the PCA reduced dataset is employed. However, with the initial p_0 regressors, the performance of NN are less good than LARS, RR and SS; this last result underlies that great attention must be paid to the use of NNs for modeling of high-dimensional dataset.

In Fig. 5.10 the evolution of the coefficients θ_j is reported, with RR at different values of the regularization parameter and SS and LARS at different steps of the algorithm procedure. We have employed a smaller input dataset with just ten variables from the original input dataset; the regression has been made on this new dataset for visualization

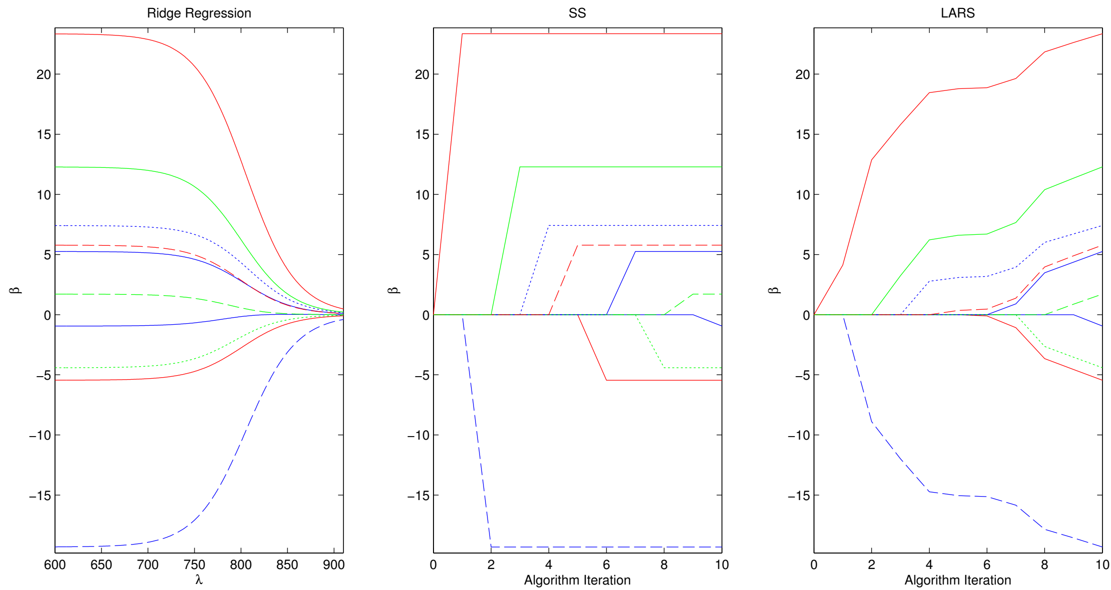


Figure 5.10: On *CLA* data, coefficient paths for RR, SS and LARS for a limited input dataset with 10 regressors.

purposes. It can be seen how all coefficients converge (at the end of SS and LARS procedure and for $\lambda \longrightarrow 0$) to the same solution (OLS).

In Table 5.2 the performances of the proposed models in terms of Mean Squared Error (MSE)

$$\mathrm{MSE} = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}, \tag{5.8}$$

are summarized, whereas in Table 5.3 the performances are given in terms of the *Mean*

Data cluster	NN	LARS	Ridge Reg.	SS	OLS
CLA	**1.8962**	2.4094	2.4419	2.6098	3.7594
CLB	**2.3064**	2.3562	2.9725	3.1436	4.5242
CLC	2.5896	**2.5736**	3.2520	3.4780	5.0867
CL$T1$	**5.2855**	6.8839	7.1175	8.1129	12.1144
CL$T2$	**3.9793**	4.8985	7.2760	5.4976	7.6425
CL$T3$	4.6062	**4.3751**	5.8884	6.2096	8.8097
CL$T4$	**3.8794**	4.6122	4.9829	5.2946	7.3647

Table 5.2: VM model performances: average MSE for each model techniques on $M = 10000$ simulations. In bold are reported for each cluster the algorithm that has minimum MSE.

Data cluster	NN	LARS	Ridge Reg.	SS	OLS
CLA	**1.0979**	1.3971	1.4106	1.5007	2.1823
CLB	**1.3227**	1.3663	1.7282	1.8073	2.6129
CLC	1.5024	**1.5011**	1.8767	2.0261	2.9446
CLT1	**3.0398**	3.9491	4.0934	4.7043	6.9392
CLT2	**2.2752**	2.8248	4.1965	3.1794	4.4379
CLT3	2.6558	**2.5328**	3.3433	3.58	5.0892
CLT4	**2.2705**	2.6638	2.8876	3.0936	4.2417

Table 5.3: VM model performances: average MAPE for each model techniques on $M = 10000$ simulations. In bold are reported for each cluster the algorithm that has minimum MAPE.

Absolute Percentage Error (MAPE) defined as

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|. \tag{5.9}$$

NNs generally grant lower values of MSE and MAPE than linear methods. It is important to underline that prediction accuracy seems to be affected more by the clustering than the modeling algorithm. Clustering is then discussed in further detail in the next Section.

5.5 Clustering

As pointed out in the previous Sections, effective data clustering is an important element to obtain accurate VM systems. As stated before, on a CVD machine, hundreds of different products are run, each one of them with its own tool settings. If we wanted to model each one of these products singularly, we would have several of them associated to

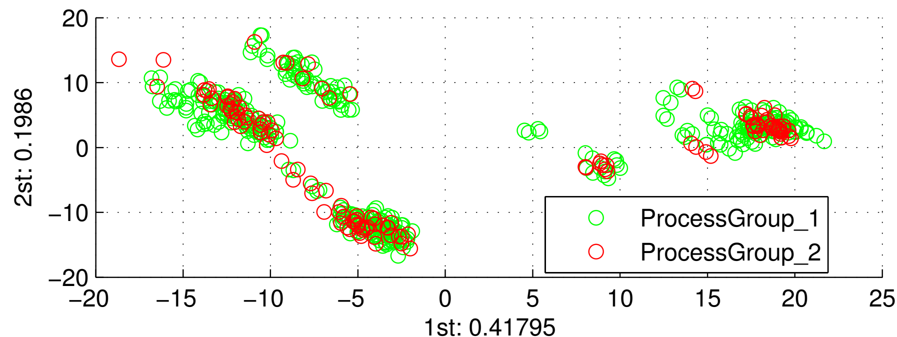


Figure 5.11: First two PCs for two different products (ProcessGroup_1 and ProcessGroup_2). Variability explained by the corresponding PC is shown on the two axes.

really few data, not enough to build a reliable statistical model. However a great amount of products present similar FDC data, as can be appreciated by exploiting again PCA and visualizing the first PCs (an example is reported in Fig. 5.11). For those products whose FDC data distributions are 'similar', it is reasonable to model them together in order to increase the available amount of data available and consequently the confidence on the statistical model.

On the other hand, it would be impossible to examine through PCA all pairs of products, and, besides, visualizing the first 2/3 PCs could not be enough to discriminate if two products are statistically 'close' or not. We propose here a *quantitative* approach to clustering based on the statistical distance of products' data distributions.

Let P and Q be the probability distributions for two different products. We define with $D_f(P\|Q)$ a f-divergence function Ali and Silvey (1966) that measures the difference between P and Q

$$D_f(P\|Q) = \int_{\mathbb{R}^p} f\left(\frac{dP}{dQ}\right) dQ. \tag{5.10}$$

f-divergence are non-negative, monoton and convex functions. The most famous f-divergences are the Kullback-Leibler divergence Kullback and Leibler (1951) and the Hellinger distance Pollard (2002), that enjoys the property of being simmetric for P and Q ($D_f(P\|Q) = D_f(Q\|P)$). The data clustering based on the statistical distance defined by (5.10) is illustrated in Algorithm 7.

Algorithm 7: f-divergence-based data clustering for chamber A.

Data: FDC tool data.
Result: Data Clustering.

1. For each couple of products $\{i, j\}$, with probability distributions P_i and P_j, in the production dataset we compute the $D_f(P_i\|P_j)$.
2. *First Clustering* - If $D_f(P_i\|P_j) < \tau$, where $\tau > 0$, is a 'small' threshold, products i and j are grouped together.
3. *Second Clustering* - If a group of products, or a single product, G has, after step 2), a total amount of observations N_G that is smaller than a threshold T_n, we add to this group the data of the product i_1 outside this group with the smallest $D_f(P_{i_1}\|P_G)$. This operation is iterated adding other products i_2, i_3, ... until $N_G \geq T_n$.

The clustering approach described in Alg. 7 is done for each chamber separately; as we have seen in Section 5.4, each chamber can be in fact considered as a different tool. To estimate distributions from data, it is possible to use a Kernel Density Estimator

Clustering	None	Chamber	Target	f-divergence
MSE	13.8902	9.4217	10.9382	**6.1954**
MAPE	8.0153	5.3997	6.3140	**3.6183**

Table 5.4: Clustering performances: average MSE and MAPE for each cluster on $M = 1000$ simulations.

Scott (1992).

We test the clustering approach based on f-divergence on the entire production dataset available that consists of N_P products. For each product $i = 1, 2, \dots, N_P$ we have n_i observations; we split them through MCCV into a training dataset of $\lfloor 0.7n_i \rfloor$ maintenance cycles and a validation dataset of $\lceil 0.3n_i \rceil$.

We compare several kind of clustering:

- *Chambers* - each product modeled by its own, with different models for different chambers;
- *Target* - products with the same target CVD thickness are modeled together;
- *f-divergence* - clustering as described in Alg. 7, with different models for different chambers;
- *None* - a model for each product.

The performances in terms of MSE (2.26) and MAPE (5.9) of the proposed clustering approaches are summarized in Table 5.4. The modeling has been done through NNs in each clustering approach. It can be appreciated that f-divergence clustering outperforms all the other approaches.

6

Multi-Step Virtual Metrology

Typically a VM module takes into consideration a single process. However, since production processes involve a high number of sequential operations, it is reasonable to assume that the quality features of a certain wafer (e.g. layer thickness, electrical test results) depend on the whole processing and not only on the last step before measurement.

In this Chapter, we investigate the possibilities to improve the VM quality relying on knowledge collected from previous process steps. We will present two different schemes of multistep VM, along with dataset preparation indications; special consideration will be reserved to regression techniques capable of handling high dimensional input spaces. The proposed multistep approaches will be tested against actual data from semiconductor manufacturing industry.

This work has been done in collaboration with Infineon Technologies AG, Austria and it is an extended version of the work firstly presented in Pampuri et al. (2012).

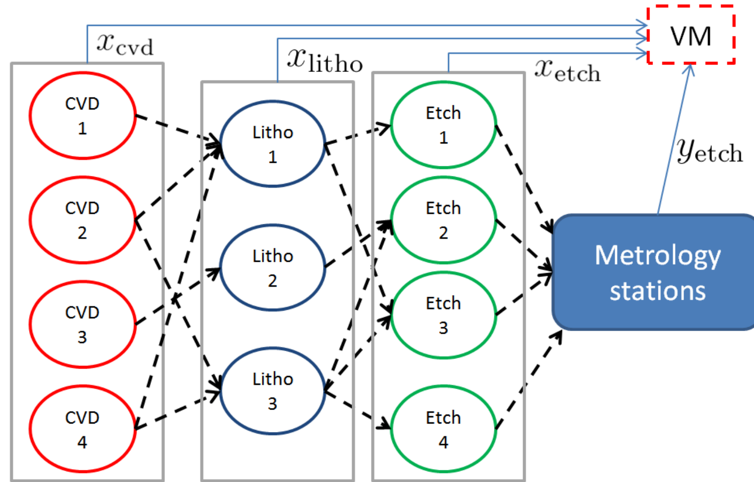


Figure 6.1: Example of process flow in semiconductor manufacturing: the black dashed lines represent wafer dispatching events, while the solid blue lines represent information flows. The Virtual Metrology (VM) block collects process data (x) for several consecutive steps, and metrology data (y) for the latest step.

6.1 Introduction

From the point of view of data modeling, the semiconductor manufacturing environment poses some serious challenges; here we tackle among the most prominent, the following:

- the already mentioned high-dimensionality, related to the number of process parameters that is usually quite high and may lead to ill-conditioned problems (Friedman, 1997).
- the multi process causes of variability, where the information regarding the outcome of a process may be not related just on the process itself, but also on previous steps.

It is quite straightforward that both problems can strongly affect prediction quality. We have already discussed in the previous Chapters the high dimensionality issue; The second issue, namely the lack of information in a single-process dataset, is somewhat more subtle: at a first glance, it may look like a data collection problem - if that would be the case, an information merge would suffice to make the problem fall back to the usual VM case. From the point of view of data analysis, though, the collected multistep data is often too fragmented to be used. The realistic example depicted in Figure 6.1, concerning the relationships between a total of 11 equipment performing CVD, Lithography and Etching process, shows why: since different process tools can perform the same process step for a specific wafer, the number of possible routes grows exponentially with the number of

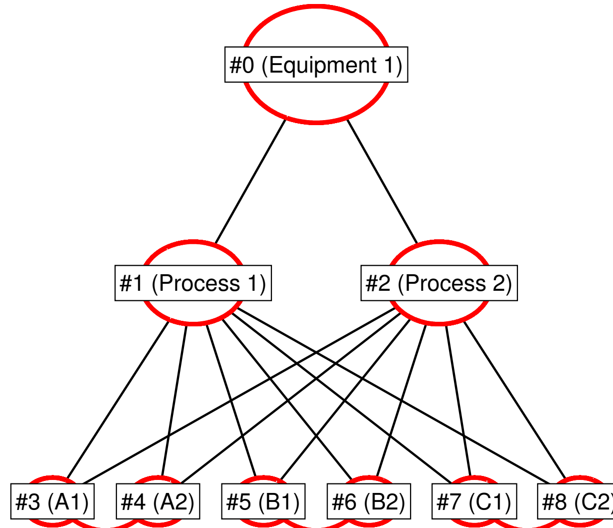


Figure 6.2: Tree representation of a CVD (Chemical Vapor Deposition) equipment with three chambers (A, B, C) with two subchambers each (1 and 2), involved in two processes (Process 1 and Process 2). Therefore, for the processed wafers, twelve distinct logistic configurations (i.e., paths) are possible.

considered steps. As a consequence, collecting an homogeneous dataset referred to a specific path would yield an insufficient number of observations.

It is interesting to note that the issue of concurrent data sources, especially in realities with highly mixed production, proves problematic even in the single step virtual metrology: indeed, several equipment types are composed of different chambers (and sub-chambers), whose behavior varies significantly with respect to each other. In order to overcome such issues (Figure 6.2), multilevel methodologies have been developed to model commonalities and differences in a tree-structured logistic representation.

In this Chapter, a novel approach in performing Multistep Virtual Metrology is presented, relying on regularized machine learning methodologies and a multilevel transformation of the input space: the aim is to estimate the quality indicators of a wafer that has undergo several processes, considering all its historical process data. The remainder of the Chapter is organized as follows:

- Section 6.2 describes the proposed approach in terms of dataset preparation and model assumptions

- Section 6.3 validates the proposed methodologies through semiconductor manufacturing datasets

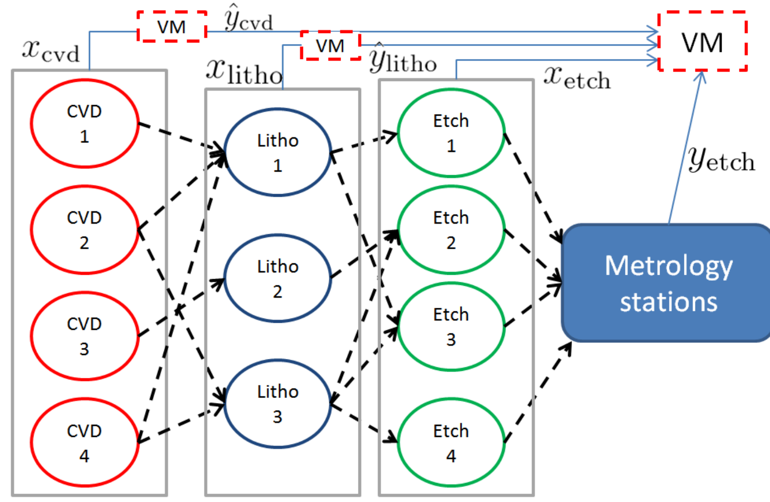


Figure 6.3: In the 'cascade VM' scenario, single-step Virtual Metrology modules are producing information for some of the previous process steps: the predicted values are then incorporated in the input space.

6.2 Multistep Virtual Metrology

In this section, three main strategies of Multistep VM are presented. First of all, with reference to Figure 6.1, we define a standard setting:

- A production flow is defined as a sequence of steps; each step represents an operation that must be performed on a wafer to obtain a specific results. Examples can include a deposition step, lithography and an etching operations, as depicted in Figure 6.1, but also intermediate steps such as coating and thermal oxidation.
- Each step is performed by different equipments and the knowledge of which equipment processed a specific wafer is available. Furthermore, each equipment might be composed of different chambers.
- Each equipment provides information about the processed wafer, including sensor readings and recipe set points. It is assumed that all the equipments that deal with a certain step (for instance, all the involved CVD equipments) are able to yield a compatible set of readings.
- On some equipments a "single step" VM system is already in place; that allows to have an estimated measure for each processed wafer (Figure 6.3).

In the following subsections, the different multistep philosophies are grouped by the type of information and previous knowledge that they require. The standard assumption

is that for the last step of the considered production flow, whose metrology values are the targets to be predicted, all relevant information is available. In the following, let us consider a process consisting of L sequential steps; the i-th step can be performed by η_i different equipments, while the total number of equipments involved in the dataset is η. Furthermore, let $X_i \in \mathbb{R}^{n \times p_i}$ be the input matrix related to a specific process step, and let $Y \in \mathbb{R}^n$ be the target array of measurement values. The matrix

$$X \in \mathbb{R}^{n \times \bar{\eta}}, \qquad \text{where } \bar{\eta} = \sum_{i=1}^{\eta} p_i \tag{6.1}$$

that serves as input for the learning problem is then obtained by means of the Multilevel Transform.

The Multilevel Transform

Given the data fragmentation problem described in Section I, additional assumptions are needed in order to obtain a feasible learning problem. Following the Multilevel paradigm defined in Schirru et al. (2011), the aim is to fit a Generalized Additive Model of the form

$$f(X) = \sum_{k=1}^{\eta} f_k(X_k) \tag{6.2}$$

that is, the prediction arises as the sum of independent effects connected to all the logistic entities involved in the process. In (6.2), X_k represents the input space associated to the j-th entity for the generic wafer x; that is,

$$X = [X_1 \ \ldots \ X_\eta]$$

where the k-th line of X_i contains either the process parameters collected from the i-th equipment (if it did process the k-th wafer), or is otherwise padded by zeros. It should be noted that the implied linear effect superposition is a strong assumption, and one that is asymptotically suboptimal. Such assumption has, however, proven to be effective when performing VM in small datasets situations.

The next subsections cover the different options to create the matrices X_i, according to the various philosophies of Multistep Virtual Metrology.

Cascade Multistep

Assuming that each equipment in the production flow of interest have a ”single step” VM system (Figure 6.3), it is possible to incorporate the already existing predictions as

input variables while creating the prediction model. In order to exploit such information in the model, the input matrix X_i is defined as follows:

$$\begin{cases} X_i = \hat{Y}_i & i \neq \eta \\ X_i = \text{process data} & i = \eta \end{cases}$$

That is, the input matrices are populated with previous Virtual Metrology predictions for equipments that do not belong to the target step, and process data for the rest. This approach is called *cascade Virtual Metrology* as it would allow to build a pipe system in which the predictive information is propagated forward to concur to further model estimations. The main advantage of this methodology is the small overhead appended to the input space: this can be an important factor, both from the computational point of view and to ease the model selection process. Conversely, the two main drawbacks of this approach are:

- Virtual Metrology systems must already be in place for steps that precede the target step
- The output of a VM system is essentially a weighted combination of some process parameters optimized to predict a predefined measure; therefore, there might be some information loss between two or more steps.

Logistic Multistep

Given the predominance of logistic information (for instance, chamber position) in prediction models that deal with stable processes, this technique allows a sensitive reduction of the input space size by employing modal variables to indicate the position of a certain wafer, at least for some of the non-target process steps. Once the possible logistic paths are fixed, it is possible to embed this information in our model(s) with a binary indicator function: X_i is a column whose k-th entry is 1 if the wafer has been processed in the i-th production tool and 0 otherwise.

This approach allows to plug in the model the logistic information of previous steps with relatively small increasing in input matrix's size (depends on the granularity of logistic in which we are interested); anyway, the matrix block added to the standard input matrix is very sparse and will be easily handled by the algorithms presented in Chapter 2. Of course, this strategy strongly relies on the assumption that the variability in the process outcome is related more to logistic information than sensor readings; as this is not often the case, experiments would be needed to assess the appropriateness of such solution.

Process-based Multistep

With this approach, all the relevant process and recipe information from all the considered steps is included in the input set. In this case, the generation of X_i fully follows the above described Multistep paradigm. From the theoretical point of view, this approach presents a series of benefits:

- It allows to include data from steps for which no measurements are available, or whose measurements are devoid of meaning with respect to the target step.
- It provides all the available information to the learning algorithms

On the other hand, the input space dimension is significantly increased by this approach; it is likely that a higher number of observations will be needed in order to estimate the predictive model.

6.3 Results

In order to validate the proposed Multistep VM approach, a dataset from the semiconductor manufacturing industry (courtesy of the Infineon Technology Austria facility in Villach) is employed as a benchmark. Such dataset has been collected considering the following production flow:

- **Chemical Vapor Deposition** (CVD): as described in the previous Chapter, it is a process in which a thin films of solid material is produce on the surface of a wafer. The deposit is usually evaluated by measuring the thickness (THK) and the uniformity (typically the standard deviation of various measurements performed at different coordinates on the wafer), and comparing them with the desired values.
- **Thermal Oxidation**: this process consists on heating multiple wafers (usually in a furnace) in order to force an oxidizing agent to react with the wafer materials. This allows to produce a thin layer of oxide.
- **Coating**: the wafer is covered by a viscous solution of photoresist that is rapidly removed in order to produce a thin layer.
- **Lithography**: this process allows to remove predefined parts of the wafer substrate by means of photomasks; this way, geometric patterns are transferred on the photoresist. The results of this operation is evaluated measuring geometric features (e.g. height, width, depth) of the created pattern on the wafer; such features are named Critical Dimensions (CDs) (Ito and Okazaki, 2000).

In this scenario, we evaluated the performances of three different Multistep VM systems with both algorithms described in Section 2. The aforementioned systems are setting up as follows:

- **CVD-Litho Cascade**: a single step VM tool predicts the Thickness value after the CVD process; that is used as additional parameter to estimate the Critical Dimension post lithography process.
- **Full Logistic**: context data, about relevant logistic information through the whole process flow, are considered to perform VM on litho CD.
- **CVD-Litho Process and Full Logistic**: this scenario is the most complete. It takes in account process data of CVD and Litho plus logistic information of all the four steps.

The dataset, consisting of 583 wafers, is anonymized and randomly split between by means of 10-fold crossvalidation. The hyperparameter λ is then tuned for both Ridge Regression and LASSO to minimize the Root Mean Squared Error (RMSE) of the validation predictions. In order to evaluate also the improvements of the Multi Step approach with respect to standard techniques, we compared the previous mentioned RMSE with the one obtained by Single Step approach, that is taking in account just process data and logistic data of Lithography step.

The results of the analysis, presented in Figures 6.4 to 6.9, highlight several interesting points:

- The proposed Multistep VM approach allows to improve the performances of Virtual Metrology: the overall best results, method-wise, are obtained this way (Figures 6.5 and 6.7).
- The process data of the target process alone still performs fairly well (Figures 6.4, 6.5, 6.7, 6.8): intuitively, excluding data from the target step yields the worst results.
- Ridge Regression outperforms LASSO in most cases for the dataset at hand; this might be due to the presence of important correlations in the input space, where the natural averaging properties of the Ridge Regression can act as a noise-mitigating filter.
- For both algorithms, the best overall performances were obtained considering all process steps (CVD, Coating, Oxidation and Lithography).

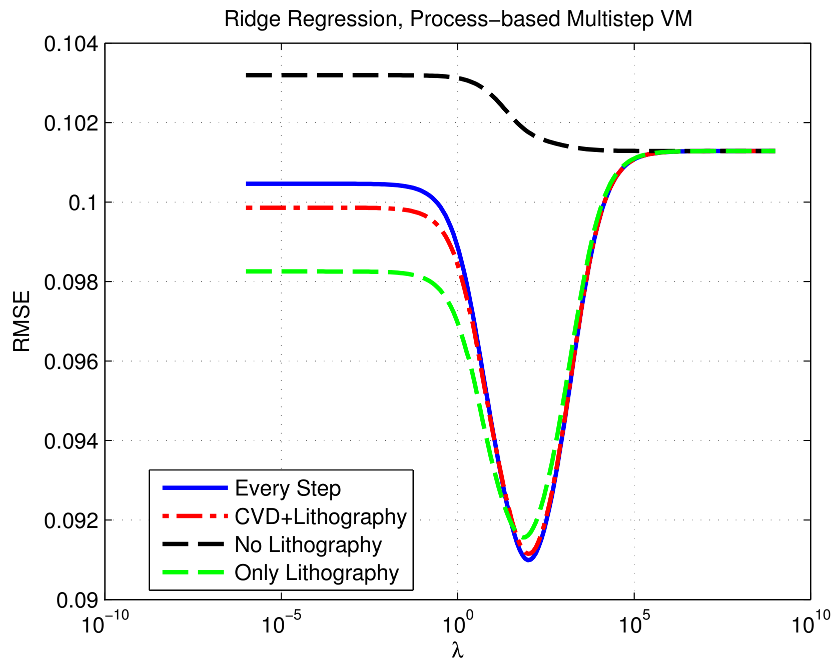


Figure 6.4: Validation RMSE results for Ridge Regression: it is apparent how the full step choice allows to improve the predictive performances.

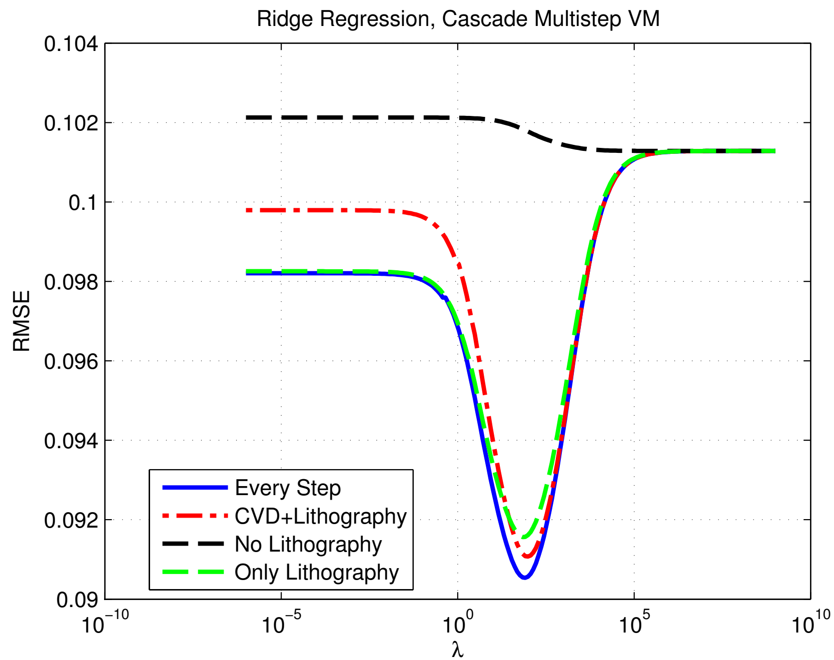


Figure 6.5: The cascade VM allows to further improve the VM performances using Ridge Regression. This somewhat counterintuitive result might be related to the additional hidden knowledge provided by the intermediate CVD metrology prediction.

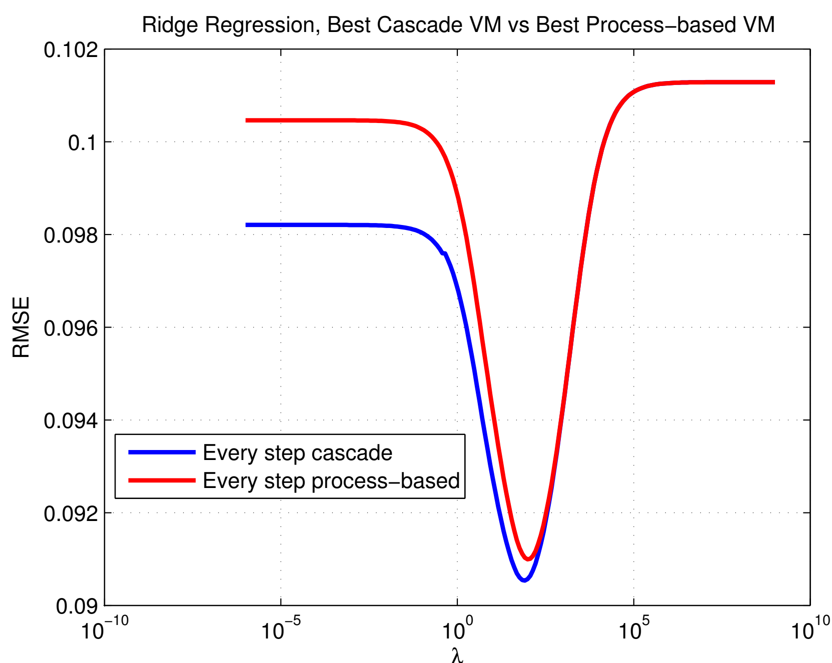


Figure 6.6: The best overall results for Ridge Regression are obtained with the cascade approach and by considering all the process steps.

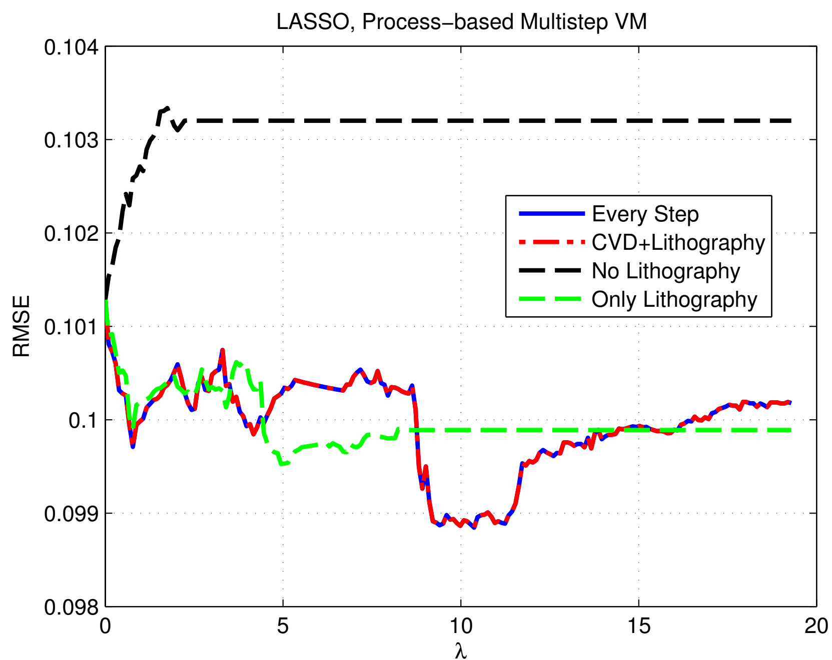


Figure 6.7: LASSO is consistently outperformed by Ridge Regression in the dataset that was used for the experiment; nevertheless, the extended input space proves to be fruitful also in this case, with respect to the Lithography-based approach.

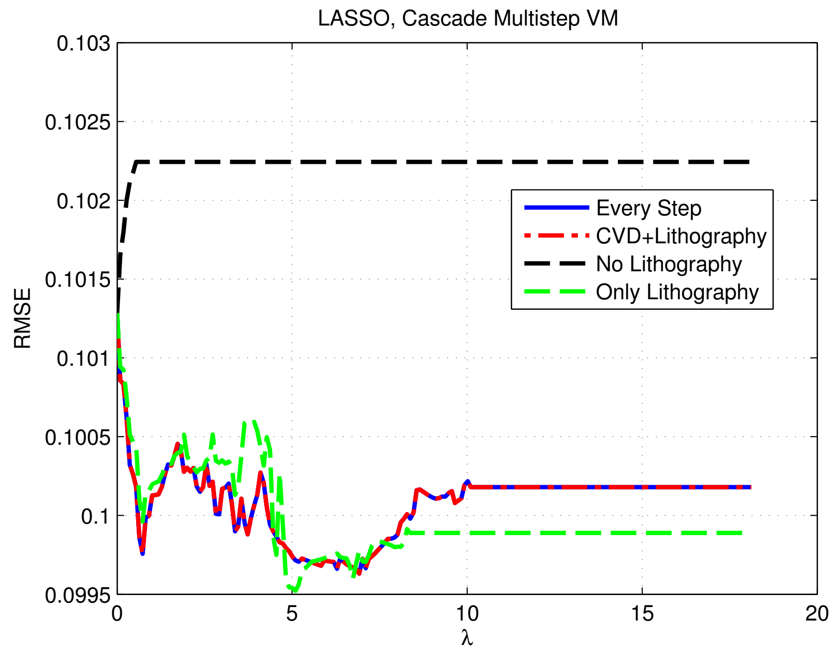


Figure 6.8: The cascade approach performs worse with the LASSO. It should be noted that this is the only case in which the extended input space does not improve the predictive performances.

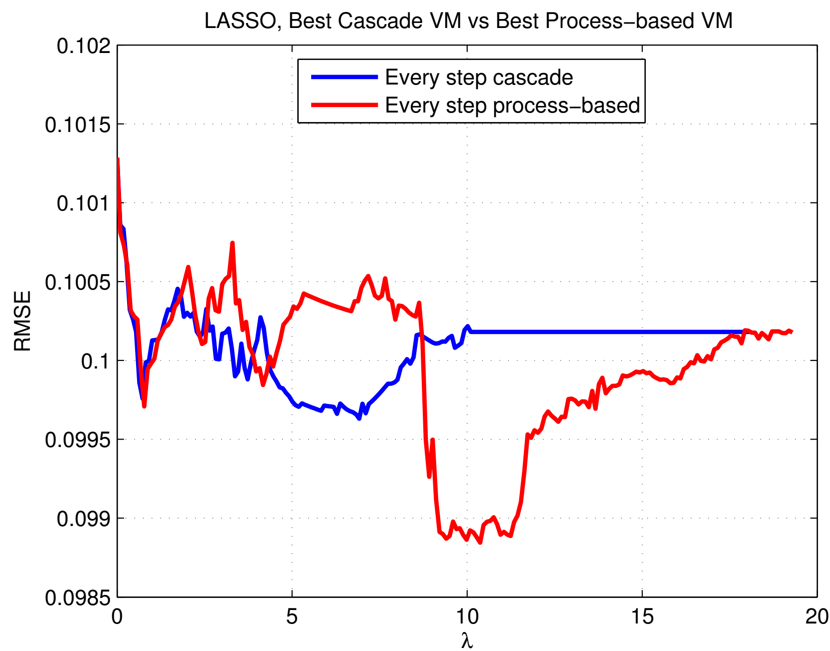


Figure 6.9: For the LASSO, the best overall results are obtained by considering the extended process values for all the involved steps.

As expected, the Single Step approach obtained worst results then Multi Step, that shows a significantly improvement in terms of accuracy on test sets. Surprisingly, the 'Cascade' scenario leads to better results, compared to most complete approach: this peculiar outcome might be attributed to the small sample size with respect to the input space of the 'Process and Full Logistic' system.

6.4 Conclusions

In this Chapter, a novel strategy for Virtual Metrology in semiconductor manufacturing was proposed. The proposed approach can be named 'Multi Step Virtual Metrology' and it consists in using information about previous process step(s), as process data, logistic data, and virtual and actual measurement values, jointly to the current process information, to improve the precision and the accuracy of the Virtual Metrology system. Furthermore, this strategy allows to taking in account processes without measurements, and it is highly customizable to suit any use cases. This method was tested in a specific production flow consisting of four steps, three different Multi Step strategies were compared, and two machine learning algorithms were used to build the VM models.

The tests, on dataset of semiconductor manufacturing industry, show promising results; however, the strategy to be implemented must be carefully designed: sample size and relevance of the steps are fundamental criteria to obtain the best performances.

The proposed methodologies were also tested against standard 'Single Step Virtual Metrology' approach, showing significant performances improvement.

7 Virtual Metrology with Time Series Data

Many modeling problems require to estimate a scalar output from one or more time series; VM problems usually belongs to this category.

Such problems are usually tackled by extracting a fixed number of features from the time series (like their statistical moments), with a consequent loss in information that leads to suboptimal predictive models. Moreover, feature extraction techniques usually make assumptions that are not met by real world settings (e.g. uniformly sampled time series of constant length), and fail to deliver a thorough methodology to deal with noisy data.

To overcome the aforementioned problems it is illustrated here a methodology, firstly presented in Schirru et al. (2012b) and Schirru, Susto, Pampuri, and McLoone (2012c), based on functional learning; the Supervised Aggregative Feature Extraction (SAFE) approach allows to derive continuous, smooth estimates of time series data (yielding aggregate local information), while simultaneously estimating a continuous shape function

yielding optimal predictions. The novel feature extraction framework will be defined and presented for dataset consisting of time series input spaces and scalar target variable. The research is originally motivated by real-life datasets representing industrial processes (where the input is represented by sensor readings and the output is a quantitative indicator of product quality), but the presented results are applicable to any time series-intensive learning environment.

The SAFE paradigm enjoys several properties like closed form solution, incorporation of first and second order derivative information into the regressor matrix, interpretability of the generated functional predictor and the possibility to exploit Reproducing Kernel Hilbert Spaces setting to yield nonlinear predictive models. The proposed methodology derives from a functional learning setting in which the time series input space is reconstructed by means of Gaussian process inference, and the unknown shape function is parametrized as a weighted sum of Gaussian functions. This setup allows for a number of interesting properties, including closed form solution and the possibility of using the extracted information as input for any machine learning methodology.

In the following, simulation studies are provided to highlight the strengths of the new methodology with respect to standard unsupervised feature selection approaches.

7.1 Introduction and Problem Statement

The mathematical settings for the regression problems described in chapter 2 in real life applications data is rarely (if ever) organized in a convenient $n \times p$ matrix ready to serve as input for a machine learning procedure: for many relevant learning problems, obtaining a mathematical representation of the input space is not trivial.

Indeed, the transition from a real life object to its mathematical representation will necessarily destroy part of the original information. A notable example of this fact occurs in text data mining (Dai, Chang, R.T.-H., and Tsai, 2010), where the goal is to understand the meaning of written text: providing a compact mathematical representation with little information loss is one of the biggest challenges in the field.

In this chapter, we consider the learning problem where the input information is conveyed in the form of time series; more specifically, every observation of the phenomenon is described by p time series, that we know through an array of irregularly sampled measurements whose size can vary observation-wise. This setting relates to a common problem in predicting process results in an industrial setting (Schirru et al., 2011), where the input space is often represented by non-uniformly sampled sensor readings. The challenge is to aggregate the information contained in each time series so that summary

features are produced that are good predictors of the target value.

Assuming the existence of a continuous process underlying such sensor readings, we adopt a functional learning paradigm in order to tackle the presented problem: in the following a suitable estimation technique to reconstruct the original continuous time series and derive a feature extraction technique that can be employed with regular machine learning techniques will be discussed, which we refer to as Supervised Aggregative Feature Extraction (SAFE). Furthermore, it will be shown the advantage of the proposed methodology with respect to other approaches by means of numerical simulations.

We now define mathematically the problem.

Given n observations consisting of p time series, where the i-th observation $\mathcal{X}_i$ is defined as

$$\mathcal{X}_i = [x_i^{(1)}(t) \ \dots \ x_i^{(j)}(t) \ \dots \ x_i^{(p)}(t)], \ t \in [0,1], \forall \ j$$

and a scalar target variable y_i, let

$$\mathcal{S} = \{\mathcal{X}_i, \ y_i\}_{i=1}^n$$

be the training set. The goal is then to learn, relying on $\mathcal{S}$, a predictor function f. Such a predictor must be optimal in the sense that, given a new input $\mathcal{X}_{\text{new}}$, $f(\mathcal{S}, \mathcal{X}_{\text{new}})$ will be close (in the sense of a normed distance) to the unobserved y_{new}.

In practice, the continuous time series $x_i^{(j)}(t)$ are most often not available: instead it is necessary to rely on a set of discrete observations (samples)

$$\left\{t_{i,s}^{(j)}, \ z_{i,s}^{(j)}\right\}_{s=1}^{\mathcal{N}_{i,j}}$$

where $t_{i,s}^{(j)}$ and $z_{i,s}^{(j)}$ are the time and value of the s-th sampled point from the j-th time series of the i-th observation. In general, the series may have difform length (such that $\mathcal{N}_{i,j} \neq \mathcal{N}_{i,m}$, $\mathcal{N}_{i,j} \neq \mathcal{N}_{k,j}$) and sampling timestamps ($t_{i,s}^{(j)} \neq t_{i,s}^{(m)}, t_{i,s}^{(j)} \neq t_{k,s}^{(j)}$). Furthermore, the noise of the channel needs to be taken into account:

$$\begin{aligned} z_{i,s}^{(j)} &= x_i^{(j)}(t_{i,s}^{(j)}) + v_{i,s}^{(j)} \\ v_{i,s}^{(j)} &\sim N(0, \rho_j^2) \end{aligned}$$

In order to employ machine learning techniques to find f, two main issues must be addressed: **(i)** it is in general necessary to extract a homogeneous set of features from every observation, and **(ii)** it is not possible to know in advance what part of the time series (if any) has an impact on the target variable. This lack of information must be taken into account when choosing a feature extraction methodology: indeed, a representation

based solely on the global features of a dataset is likely to yield suboptimal predictions.

In the next section, some of the most common feature extraction techniques for time series are presented and discussed.

7.2 Feature extraction

It is to be noted that, in general, the extraction of a set of features from an observation will result in the loss of some information. This is especially true when the format of such information is expected to show inter-example differences, such as in the presented case where the feature extraction procedure needs to deal with difform sampling times and length.

The goal is to build a regressor matrix $\Phi \in \mathbb{R}^{\overline{n} \times \overline{p}}$, whose entry (i, j) represents the j-th feature of the i-th observation that can be subsequently used, along with the target variable vector $Y \in \mathbb{R}^{\overline{n}}$, to train a predictor using a machine learning algorithm.

One of the simplest approaches is to rely on statistical moments: given p time series, let us build Φ as

$$\Phi = [\Phi_1 \ \dots \ \Phi_j \ \dots \Phi_p]$$

where the $[i, k]$ element of $\Phi_j \in \mathbb{R}^{\overline{n} \times k_{\max}}$ is

$$\Phi_j[i, k] = m^{(k)} \left(\left\{ z_{i,s}^{(j)} \right\}_{s=1}^{\mathcal{N}_{i,j}} \right)$$

Here $k_{\max}$ is the highest considered moment order and $m^{(k)}(\cdot)$ is the k-th sample moment of the input time series; a common choice is to build the matrix up to the fourth moment (kurtosis). It is immediately evident that this approach suffers from a major drawback, namely the inability to consider the dependency between information and time. Furthermore, it should be noted that the sample estimators of statistical moments are consistent for *independent* data points: it follows that, in the quite common case of autocorrelated time series, such estimates bear very little statistical meaning.

A more sophisticated approach consists of a systematic sampling of the input time series: specifically, the interval $[0, 1]$ is divided into $\mathcal{N}$ segments $[\tau_1 \ \dots \ \tau_{\mathcal{N}}]$. The regressor matrix is then populated with the segment-wise averages, as

$$\Phi_j[i, k] = \text{Avg}[z_{i,s}^{(j)} : t_{i,s}^{(j)} \in \tau_k].$$

When using this approach it is necessary to select the number of segments, $\mathcal{N}$, in advance: this usually translates to a trade-off decision between locality (temporal resolution)

and stability of information (robustness to noise). Furthermore, in the case of difform sampling, different features are likely to be computed from a different number of values, as the distribution of sampling points would privilege some segments over the others: this can potentially lead to data reliability issues.

In order to overcome such instabilities, it is possible to project the rows of the Φ matrix obtained using the sampling approach on their direction of main variance. This yields the Principal Component Analysis (PCA) (Hastie et al., 2009) transformation of the sampled input space (Section 5.3).

Other feature extraction techniques rely on the general concept of populating Φ with the optimal coefficient estimates for a model of the time series whose structure is fixed in advance: for instance, the coefficients of an AutoRegressive Moving Average (ARMA) (Box, Jenkins, and Reinsel, 1964) model of fixed order. It should be noted that ARMA modeling is not suited to the problem at hand, as it requires the time series to be uniformly sampled; in general, though, the main issue with this approach is selecting the "right" model structure among infinite possibilities.

With all the aforementioned approaches, once the regressor matrix Φ is obtained, it is possible to employ a machine learning technique to find a predictor model f as described in Chapter 2. In the following section we will show a new approach to cast the time-series information in a regressor matrix Φ that will be fed to machine learning algorithm.

7.3 SAFE: Supervised Aggregative Feature Extraction

In this section the proposed supervised aggregative feature extraction (SAFE) methodology is presented and motivated from a theoretical point of view.

In order to introduce SAFE, we consider an ideal case, in which the continuous functions $x_i^{(j)}(t)$ are known and available. Employing the functional regression paradigm, consider the following definition of f:

$$f(\mathcal{X}_i) := \sum_{j=1}^{p} \left\langle x_i^{(j)}(t), \beta^{(j)}(t) \right\rangle_{L^2} \tag{7.1}$$

where $\langle f, g \rangle_{L^2}$ is the L^2 inner product of real functions f and g, defined as

$$\langle f, g \rangle_{L^2} = \int_{-\infty}^{\infty} f(t)g(t)dt$$

It is apparent how the predictor defined by (7.1) assumes that the continuous phenomenon x influences the target variable y through a weighted integration with an unknown shape

function β (Figure 7.1).

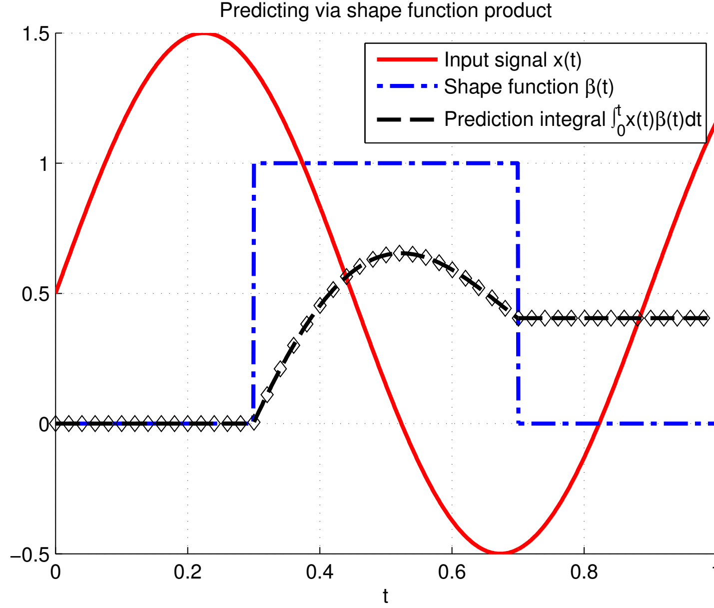


Figure 7.1: The input signal (solid line) times the shape function (dash-dotted line) is integrated to obtain the target value. The final value ($t = 1$) of the prediction integral (squares) is the prediction output.

In the following we focus on the sum of squared residuals approximation error term, defined as

$$\mathcal{F}(\beta) = \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \int_{-\infty}^{\infty} \beta^{(j)}(t) x_i^{(j)}(t) dt - y_i \right)^2 \tag{7.2}$$

It is then possible to introduce the functional learning optimization problem:

$$\beta^* = \arg\min_{\beta} \mathcal{F}(\beta) + \lambda \mathcal{R}(\beta) \tag{7.3}$$

$$\beta = \left[\beta^{(1)}(t),\ \beta^{(j)}(t),\ \beta^{(p)}(t) \right] \tag{7.4}$$

where $\mathcal{F}(\beta)$ is defined in (7.2) and $\mathcal{R}(\beta)$ is a regularization term that penalizes the variability of β: for example,

$$\mathcal{R}(\beta) = \sum_{j=1}^{p} \left\langle \beta^{(j)}, \beta^{(j)} \right\rangle_{L^2}$$

It is apparent that the shape functions $\beta^{(\cdot)}(t)$ are functional parameters of the optimization problem (7.4). It is to be noted that it is not possible to directly handle

(7.2) for two reasons: **(i)** the functions $x_i^{(j)}(t)$ are observed only through a finite number of noisy, irregularly sampled data points, and **(ii)** the generic functions $\beta^{(j)}(t)$ have infinite degrees of freedom. In order to overcome such issues and solve (7.4), the next sections present a Gaussian process estimation of the unobserved time series and propose a parametrization for the shape functions β.

7.4 SAFE: Time Series Approximation

Consider an approximation of the fitness function $\mathcal{L}$, defined as

$$\hat{\mathcal{L}} = \hat{\mathcal{F}} + \lambda \mathcal{R}$$

where the approximated loss function is defined as

$$\hat{\mathcal{F}} = \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \int_{-\infty}^{\infty} \beta^{(j)}(t) \hat{x}_i^{(j)}(t) dt - y_i \right)^2$$

and $\hat{x}_i^{(j)}(t)$ is an estimate of the unobserved $x_i^{(j)}(t)$. In order to obtain this estimate we consider the expected value of a monodimensional Gaussian process posterior distribution. According to Riesz's representation theorem (Rudin, 1966) a continuous interpolation of $x_i^{(j)}(t)$ from its samples is given by

$$\hat{x}_i^{(j)}(t) = \sum_{s=1}^{\mathcal{N}_{i,j}} \mathcal{K}(t, t_{i,s}^{(j)}) c_{i,s}^{(j)}$$

where $\mathcal{K}$ is a suitable positive definite kernel function (see Section 2.2). The vector $c_{i,\cdot}^{(j)}$ is obtained as

$$c_{i,\cdot}^{(j)} = (\mathbf{K} + \xi_j I)^{-1} x_{i,\cdot}^{(j)}$$

where the $[w, z]$ entry of the kernel matrix $\mathbf{K}$ is

$$\mathbf{K}_{[w,z]} = \mathcal{K}\left(t_{i,w}^{(j)}, t_{i,z}^{(j)}\right)$$

and $x_{i,\cdot}^{(j)}$ is the column vector of the available observations. It is immediately evident how every coefficient of $c_{i,\cdot}^{(j)} \in \mathcal{N}_{i,j}$ depends on all the observed points. Considering the radial

basis function kernel and the Gaussian density, such that

$$\mathcal{K}(t_1, t_2) := e^{-\dfrac{(t_1 - t_2)^2}{2\omega^2}} \tag{7.5}$$

$$G(a, b; x) := \frac{1}{\sqrt{2\pi}b} e^{-\frac{(a-x)^2}{2b^2}} \tag{7.6}$$

it follows that

$$\begin{aligned} \mathcal{K}(t_1, t_2) &= \sqrt{2\pi}\omega G(t_1, \omega^2; t_2) \\ \hat{x}_i^{(j)}(t) &= \sqrt{2\pi}\omega_{(j)} \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t). \end{aligned} \tag{7.7}$$

Hence, the continuous-time approximation of $x_i^{(j)}(t)$ is obtained as a weighted sum of Gaussian densities. It should be noted that, to obtain such approximation, it is necessary to select two hyperparameters for each time series, namely the regularization term ξ_j and the kernel bandwdith $\omega_{(j)}^2$.

7.5 SAFE: Shape Function Parametrization

Let us consider a linear combination of Gaussian densities as parametrization for $\beta^{(j)}$, such that

$$\begin{aligned} \beta^{(j)}(t) &= \sum_{k=1}^{\gamma} \alpha_k^{(j)} G(\mu(k), \sigma^2; t) \\ \mu(k) &= \frac{k-1}{\gamma - 1} \end{aligned}$$

where the parameter γ controls the number of base Gaussian components, and σ^2 is the bandwidth of the Gaussian density.

The approximate loss function $\hat{\mathcal{F}}$ takes the following form:

$$\begin{aligned} \hat{\mathcal{F}} &= \sum_{i=1}^{n} \left[\sum_{j=1}^{p} \int_{-\infty}^{\infty} \left(\sum_{k=1}^{\gamma} \alpha_k^{(j)} G(\mu(k), \sigma^2; t) \times \sum_{s=1}^{\mathcal{N}_{i,j}} \sqrt{2\pi}\omega_{(j)} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) c_{i,s}^{(j)} \right) dt - y_i \right]^2 \\ &= \sum_{i=1}^{n} \left[\sqrt{2\pi} \sum_{j=1}^{p} \omega_{(j)} \sum_{k=1}^{\gamma} \alpha_k^{(j)} \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} \times \int_{-\infty}^{\infty} \left(G(\mu(k), \sigma^2; t) G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) \right) dt - y_i \right]^2 \end{aligned}$$

We now consider the following Theorem:

Theorem 7.5.1. *Let $a, b, x \in \mathbb{R}^p$ and $A, B \in \mathbb{R}^{p\times p}$. It holds that*

$$\int_{-\infty}^{\infty} G(a, A; x)G(b, B; x)dx = G(a, A + B; b)$$

where G is the Gaussian density as in (7.6).

In order to prove Theorem 7.5.1, let $G(b, B; x)$ be the univariate Gaussian probability distribution function of expected value b and variance B as in (7.6). By applying derivative rules, it follows that

$$\frac{\partial G(b, B; x)}{\partial x} = -\left(\frac{x-b}{B}\right) G(b, B; x) \quad (7.8)$$

$$\frac{\partial^2 G(b, B; x)}{\partial^2 x} = \frac{G(b, B; x)}{B^2}((x-b)^2 - B) \quad (7.9)$$

Furthermore, in the following we consider the theorem first proposed in Miller (1964)

Theorem 7.5.2. *Let $\mathbf{A} \in \mathbb{R}^{s\times s}$, $\mathbf{a} \in \mathbb{R}^s$, $\mathbf{B} \in \mathbb{R}^{t\times t}$, $\mathbf{b} \in \mathbb{R}^t$ and $\mathbf{Q} \in \mathbb{R}^{s\times t}$. Let $\mathbf{x} \in \mathbb{R}^t$ be an input variable. It holds that*

$$G(\mathbf{a}, \mathbf{A}; \mathbf{Qx})G(\mathbf{b}, \mathbf{B}; \mathbf{x}) = G(\mathbf{a}, \mathbf{A} + \mathbf{QBQ}'; \mathbf{b}) \times \\ \times G(\mathbf{d}, \mathbf{D}; \mathbf{x})$$

with

$$\mathbf{D} = (\mathbf{Q}'\mathbf{A}^{-1}\mathbf{Q} + \mathbf{B}^{-1})^{-1}$$
$$\mathbf{d} = \mathbf{b} + \mathbf{DQ}'\mathbf{A}^{-1}(\mathbf{a} - \mathbf{Qb})$$

in the special case for which $s = t = 1$ and $\mathbf{Q} = 1$.

Proof. of Theorem 7.5.1. Let

$$\chi = \int_{-\infty}^{\infty} G(a, A; x)G(b, B; x)dx$$

By applying Theorem 7.5.2,

$$\begin{aligned}\chi &= \int_{-\infty}^{\infty} G(d,D;x)G(a,A+B;b)dx \\ &= G(a,A+B;b)\int_{-\infty}^{\infty} G(d,D;x)dx\end{aligned}$$

Since by definition $\int_{-\infty}^{\infty} G(d,D;x)dx = 1$ it holds that $\chi = G(a,A+B;b)$.

■

Theorem 7.5.1 allows $\hat{\mathcal{F}}$ to be rewritten as

$$\hat{\mathcal{F}} = \sum_{i=1}^{n}\left(\sqrt{2\pi}\sum_{j=1}^{p}\omega_{(j)}\sum_{k=1}^{\gamma}\alpha_k^{(j)} \times \sum_{s=1}^{\mathcal{N}_{i,j}} c_{i,s}^{(j)} G(\mu(k),\sigma^2+\omega_{(j)}^2;t_{i,s}^{(j)}) - y_i\right)^2.$$

Defining the parameters

$$\begin{aligned}\delta_{i,s}^{(j)}(k) &= \sqrt{2\pi}c_{i,s}^{(j)}\omega_j G(\mu(k),\sigma^2+\omega_{(j)}^2;t_{i,s}^{(j)}) \qquad (7.10)\\ \overline{\delta}_i^{(j)}(k) &= \sum_{s=1}^{\mathcal{N}_{i,j}}\delta_{i,s}^{(j)}(k), \qquad (7.11)\end{aligned}$$

yields the compact version of $\hat{\mathcal{F}}$ as

$$\hat{\mathcal{F}} = \sum_{i=1}^{n}\left(\sum_{j=1}^{p}\sum_{k=1}^{\gamma}\alpha_k^{(j)}\overline{\delta}_i^{(j)}(k) - y_i\right)^2 \qquad (7.12)$$

Equation (7.12) can be expressed in matrix form as

$$\hat{\mathcal{F}} = ||\Phi\theta - Y||^2$$

with $\Phi = \Delta$, where Δ is defined as

$$\Delta = \begin{bmatrix} \overline{\delta}_1^{(1)}(1) & \dots & \overline{\delta}_1^{(1)}(\gamma) & \overline{\delta}_1^{(2)}(1) & \dots & \overline{\delta}_1^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\delta}_i^{(1)}(1) & \dots & \overline{\delta}_i^{(1)}(\gamma) & \overline{\delta}_i^{(2)}(1) & \dots & \overline{\delta}_i^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\delta}_n^{(1)}(1) & \dots & \overline{\delta}_n^{(1)}(\gamma) & \overline{\delta}_n^{(2)}(1) & \dots & \overline{\delta}_n^{(p)}(\gamma) \end{bmatrix}$$

$$\theta = \begin{bmatrix}\alpha_1^{(1)} & \alpha_2^{(1)} & \cdots & \alpha_k^{(j)} & \cdots & \alpha_\gamma^{(p)}\end{bmatrix}' \qquad Y = \begin{bmatrix} y_1 & \dots & y_n\end{bmatrix}'$$

Since $\hat{\mathcal{F}}$ is a quadratic form of the coefficients α, it is convex. If $\mathcal{R}$ is convex as well, the solution of the problem can be found by solving

$$\frac{\partial \hat{\mathcal{L}}}{\partial \theta} = 0$$

with respect to θ. For instance, the Ridge Regression solution follows from Equation (2.11).

7.6 SAFE: Derivatives Basis Expansion

In this section, the convenient properties of the proposed approximation (7.7) are exploited to expand the regressors matrix to include information about its derivatives. The theory behind first- and second-order derivative expansion is covered and the corresponding formula are provided. Let us consider the first derivative of $\hat{x}_i^{(j)}(t)$

$$\frac{\partial \hat{x}_i^{(j)}(t)}{\partial t} = -\frac{\sqrt{2\pi}}{\omega_{(j)}} \sum_{s=1}^{\mathcal{N}_{i,j}} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) c_{i,s}^{(j)} (t - t_{i,s}^{(j)})$$

By exploiting the following

Theorem 7.6.1. *Letting all the quantities be as in Theorem 7.5.1, it holds that*

$$\int_{-\infty}^{\infty} G(a, A; x) \frac{\partial G(b, B; x)}{\partial x} dx = \Omega G(a, A + B; b)$$

with

$$\Omega = \left(\frac{b - a}{A + B} \right)$$

Proof. of Theorem 7.6.1

$$\begin{aligned}
\chi &= \int_{-\infty}^{\infty} G(a, A; x) \frac{\partial G(b, B; x)}{\partial x} dx \\
&= -\frac{1}{B} \int_{-\infty}^{\infty} (x - b)\, G(a, A; x) G(b, B; x) \\
&= -\frac{G(a, A + B; b)}{B} \int_{-\infty}^{\infty} (x - b)\, G(d, D; x) dx \\
&= -\frac{G(a, A + B; b)}{B} \left(\int_{-\infty}^{\infty} x G(d, D; x) dx - b \right) \\
&= -\frac{G(a, A + B; b)}{B} (d - b)
\end{aligned}$$

Since, following Theorem 7.5.2,

$$d - b = DA^{-1}(a-b) = \frac{A^{-1}}{A^{-1} + B^{-1}}(a-b)$$

it holds that

$$-\frac{1}{B}\frac{A^{-1}}{A^{-1} + B^{-1}}(a-b) = -\frac{a-b}{A+B}$$

and therefore $\chi = -\left(\frac{a-b}{A+B}\right) G(a, A+B; b)$. ■

it is possible to define

$$\tau_{i,s}^{(j)}(k) = -\left(\frac{\delta_{i,s}^{(j)}(k)}{\omega_{(j)}^2}\right)\left(\frac{\mu(k) - t_{i,s}^{(j)}}{\sigma^2 + \omega_{(j)}^2}\right) \tag{7.13}$$

$$\overline{\tau}_i^{(j)}(k) = \sum_{s=1}^{\mathcal{N}_{i,j}} \tau_{i,s}^{(j)}(k) \tag{7.14}$$

and use

$$T = \begin{bmatrix} \overline{\tau}_1^{(1)}(1) & \dots & \overline{\tau}_1^{(1)}(\gamma) & \overline{\tau}_1^{(2)}(1) & \dots & \overline{\tau}_1^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\tau}_i^{(1)}(1) & \dots & \overline{\tau}_i^{(1)}(\gamma) & \overline{\tau}_i^{(2)}(1) & \dots & \overline{\tau}_i^{(p)}(\gamma) \\ \vdots & & \vdots & \vdots & & \vdots \\ \overline{\tau}_n^{(1)}(1) & \dots & \overline{\tau}_n^{(1)}(\gamma) & \overline{\tau}_n^{(2)}(1) & \dots & \overline{\tau}_n^{(p)}(\gamma) \end{bmatrix}$$

as a basis expansion for Δ, such that $\Phi = [\Delta\ T]$. Similarly, the second derivative of $\hat{x}_i^{(j)}(t)$ is

$$\frac{\partial^2 \hat{x}_i^{(j)}(t)}{\partial^2 t} = \frac{\sqrt{2\pi}}{\omega_{(j)}^3} \sum_{s=1}^{\mathcal{N}_{i,j}} G(t_{i,s}^{(j)}, \omega_{(j)}^2; t) c_{i,s}^{(j)}((t - t_{i,s}^{(j)})^2 - \omega_{(j)}^2)$$

and, using the following

Theorem 7.6.2. *Letting all the quantities be as in Theorem 7.5.1, it holds that*

$$\int_{-\infty}^{\infty} G(a, A; x)\frac{\partial^2 G(b, B; x)}{\partial^2 x}dx = \Gamma G(a, A+B; b)$$

with

$$\Gamma = \frac{(a-b)^2 - (A+B)}{(A+B)^2} = \Omega^2 - \frac{1}{A+B}$$

where Ω *is as defined in Theorem 7.6.1.*

Proof. of Theorem 7.6.2.

$$
\begin{aligned}
\chi = & \int_{-\infty}^{\infty} G(a,A;x) \frac{\partial^2 G(b,B;x)}{\partial^2 x} dx \\
= & \frac{1}{B^2} \int_{-\infty}^{\infty} G(a,A;x) G(b,B;x) \left((x-b)^2 - B \right) dx \\
= & \frac{G(a,A+B;b)}{B^2} \int_{-\infty}^{\infty} G(d,D;x) \left((x-b)^2 - B \right) dx
\end{aligned}
$$

Since $(x-b)^2 = (x-d)^2 + b^2 - d^2 - 2bx + 2dx$ and

$$
\int_{-\infty}^{\infty} G(d,D;x)(x-d)^2 dx = D
$$

it holds that

$$
\int_{-\infty}^{\infty} G(d,D;x)((x-b)^2 - B)dx = D + (b-d)^2 - B
$$

and therefore

$$
\begin{aligned}
\chi &= \frac{D + (b-d)^2 - B}{B^2} G(a,A+B;b) \\
&= \frac{(a-b)^2 - (A+B)}{(A+B)^2} G(a,A+B;b).
\end{aligned}
$$

■

the second derivative basis expansion elements read

$$
\begin{aligned}
\eta_{i,s}^{(j)}(k) &= \left(\frac{\delta_{i,s}^{(j)}(k)}{\omega_{(j)}^4} \right) \left(\frac{(\mu(k) - t_{i,s}^{(j)})^2 - (\sigma^2 + \omega_{(j)}^2)}{(\sigma^2 + \omega_{(j)}^2)^2} \right) \\
\overline{\eta}_i^{(j)}(k) &= \sum_{s=1}^{\mathcal{N}_{i,j}} \eta_{i,s}^{(j)}(k)
\end{aligned}
\tag{7.15}
$$

The matrix H of the elements $\overline{\eta}_i^{(j)}(k)$ is then similarly used to expand the matrix Φ, as $\Phi = [\Delta \;\; T \;\; H]$.

7.7 Experimental results

In this section, the capabilities of the SAFE feature extraction technique are tested against similar methodologies: three synthetic datasets are described and used as benchmarks.

The proposed methodology was tested against the feature extraction techniques defined in Section 7.2, namely

- Statistical moments
- Systematic sampling
- PCA

The input matrices resulting from such methodologies are employed to build an optimal Ridge Regression model; 500 instances of every synthetic dataset were created, each one composed of a training set (100 examples) and a test set (50 examples). Each example consists of a single input time series (available through a number of sampling points uniformly distributed between 35 and 45) and an output target value. Gaussian distributed white noise with expected value 0 and standard deviation 0.1 was imposed on every sampled time series value and on every target value. The methodologies were evaluated using the Root Mean Squared Error (RMSE) on the test data as a performance metric. For every experiment, the SAFE technique was tested with and without the inclusion of the time series first and second derivative expansions.

In the next subsections, the three synthetic datasets are described and the comparison results presented.

The sinusoid dataset - The purpose of the sinusoid dataset is to reproduce a situation in which only an unknown part of the input time series influences the target variable. In mathematical terms, the input time series is defined as follows:

$$\begin{aligned} x(t) &= \sin(t\omega + \delta) \\ \omega &\sim \mathcal{U}(0.01, 10) \\ \delta &\sim \mathcal{U}(0, 2\pi) \end{aligned}$$

while the target variable is computed as

$$y = \int_{0.3}^{0.7} x(t)dt = \frac{\cos(0.3\omega + \delta) - cos(0.7\omega + \delta)}{\omega}$$

Figure 7.2 shows the results for the sinusoid dataset: it is apparent that, while the statistical moment-based feature extraction is not able to learn a correct model, all the other techniques yield almost the same performances. This is quite unsurprising, since the statistical moment extraction relies exclusively on global features, and is therefore unable to select the correct range in the input time series. Figure 7.3 is a zoomed version

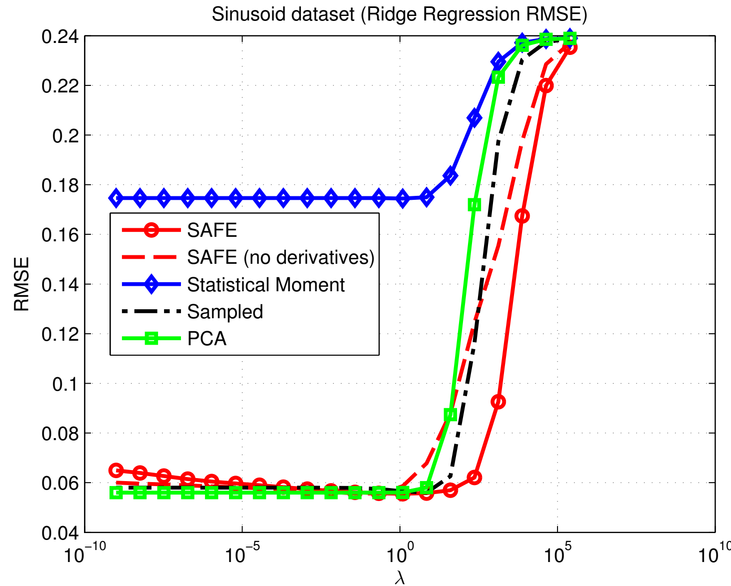


Figure 7.2: Sinusoid dataset results (average over 500 simulations)

of Figure 7.2: the SAFE methodology yields marginally better results with respect to sampling- and PCA-based feature extraction.

The ramp dataset - The purpose of the ramp dataset is to highlight the advantages of including time series derivative information in the extracted features. The input time series is generated as

$$x(t) = \begin{cases} n_1\sqrt{2t} & t < 0.5 \\ n_1 + n_2(t - 0.5) & t \geq 0.5 \end{cases}$$

$$n_1 \sim \mathcal{U}(0, 1), \quad n_2 \sim \mathcal{U}(1, 4),$$

while the output variable reads

$$y = n_2.$$

In other words, the slope of the second part of $x(t)$ (for $t \geq 0.5$) is the target variable. Figure 7.4 shows the test results for the ramp dataset. As expected, the incorporation of derivative information in the input dataset allows expanded SAFE to outperform the other methodologies.

The exponential dataset - The purpose of the exponential dataset is to test the SAFE methodology when the target variable is entirely explained by global features of the input

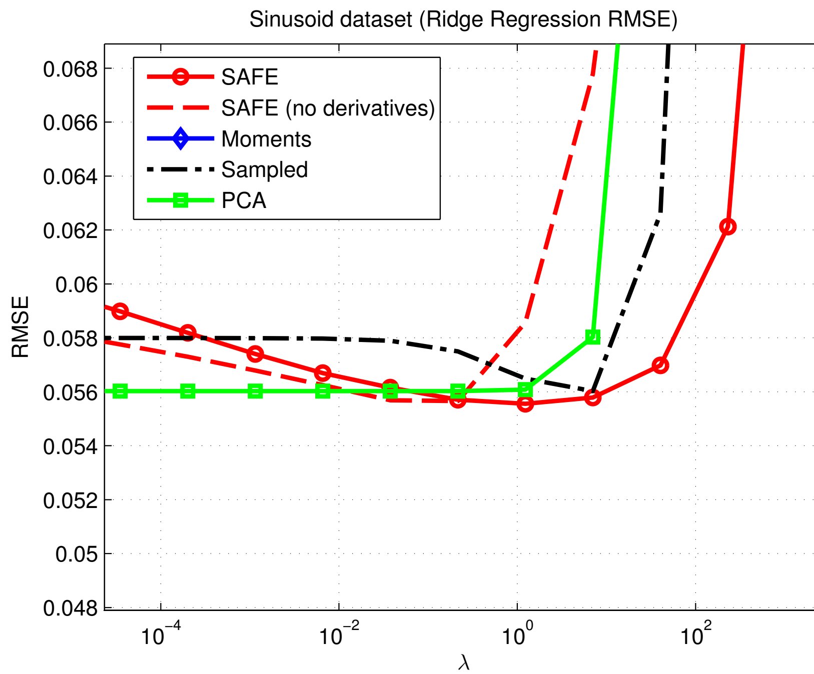


Figure 7.3: Sinusoid dataset zoomed results (average over 500 simulations)

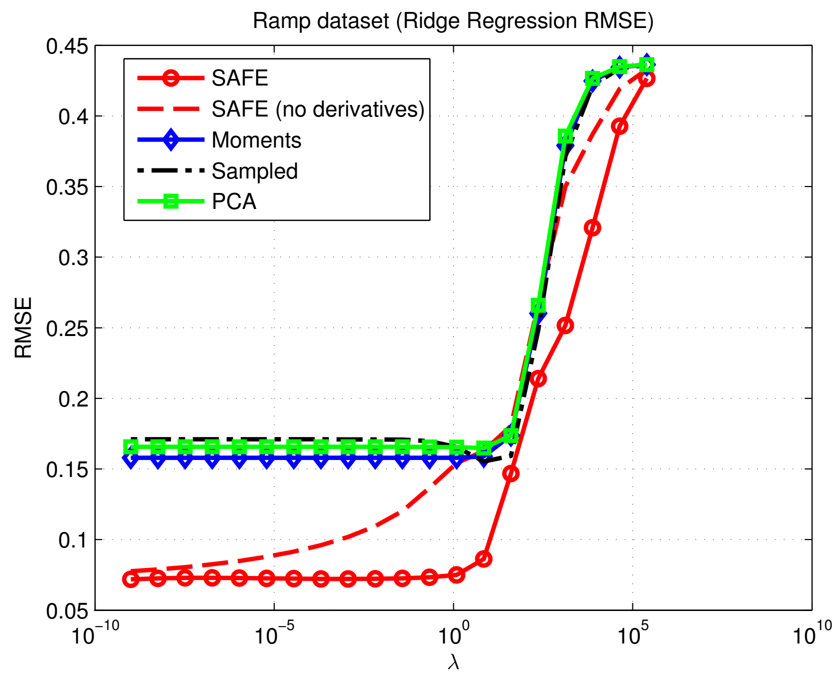


Figure 7.4: Ramp dataset results (average over 500 simulations)

time series. In this dataset, let the input series be

$$x(t) = ae^{-bt}$$
$$a \sim \mathcal{U}(8, 12) \quad b \sim \mathcal{U}(0.5, 1.5)$$

and the target variable result from

$$\begin{aligned} y &= \int_0^1 \left(x(t) - \int_0^1 x(t)dt \right)^2 dt \\ &= \frac{a^2 \left(1 - e^{-2\,b}\right)}{2\,b} - \frac{a^2 \left(e^{-2\,b} - 2e^{b} + 1\right)}{b^2} \end{aligned} \tag{7.16}$$

It is apparent how Equation (7.16) is the expected value of the second-order sample statistical moment of the observed data (sample variance): in this setting, the statistical moment extraction is expected to achieve better results with respect to all the other techniques. While this is true with respect to sampling- and PCA-based extraction, Figure 7.5 shows how the SAFE technique is again able to yield the best predictive performances. This somewhat counterintuitive phenomenon (the only relevant feature is a global one, and yet the best performances are obtained with a set of local features) is explained by the variance of the sample second-order central moment estimator: Figure 7.6 shows that, in the interested sample size (35 to 45), the estimate might suffer of important imprecision. In such sense, this dataset shows how the SAFE methodology, albeit relying exclusively on local features, is able to outperform global feature extraction even when the targeted phenomenon is global by its very definition.

Summarizing, the capabilities of the SAFE methodology have been assessed by means of the previous simulated examples, with the purpose of testing the novel framework against similar techniques and realistic situations. Such benchmarks yield promising preliminary results, as the proposed methodology is able to obtain in general better results than its competitors, including situations where the target output is determined by global features of the input time series.

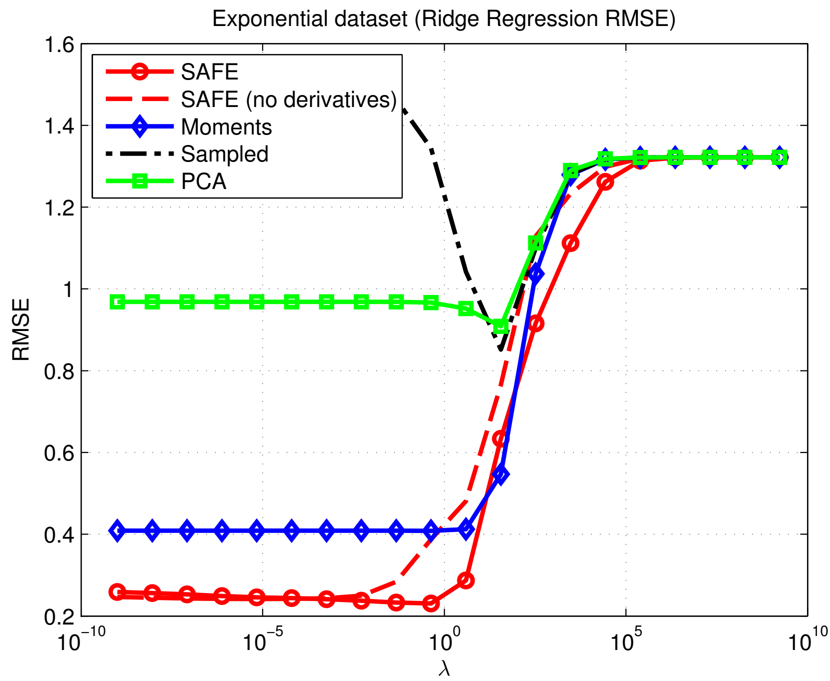


Figure 7.5: Exponential dataset results (average over 500 simulations)

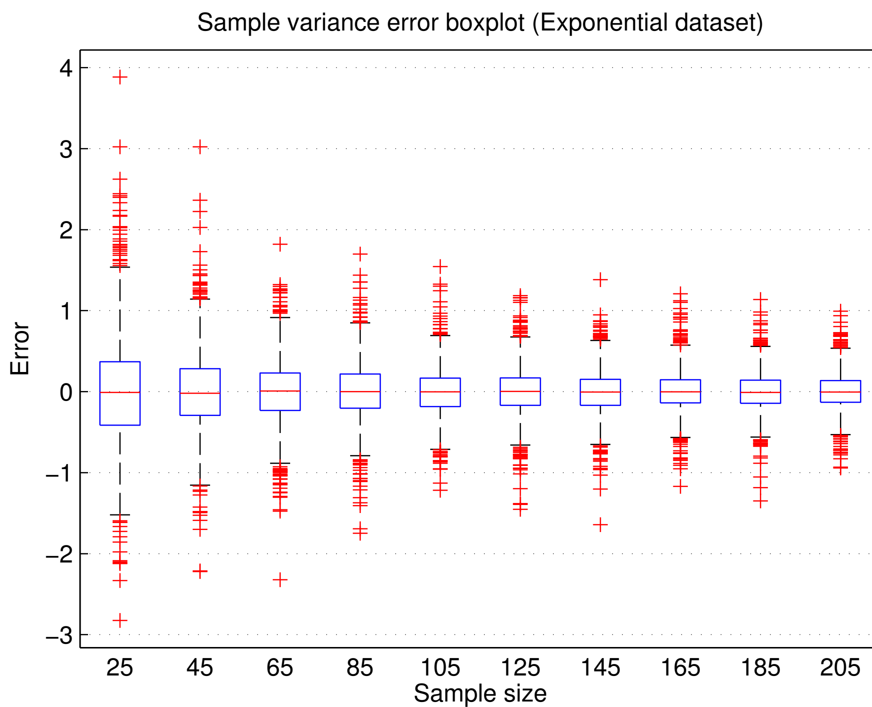


Figure 7.6: Exponential dataset: sample variance error boxplots for different sample sizes.

8

Run-to-Run Control and VM

VM tools are nowadays commonly used in semiconductor plants. However few scientific works have so far investigated interactions between VM and Run-to-Run (R2R), the most common control approach in the field. In this Chapter, a novel strategy aimed at integrating VM and R2R relying on Information Theory measure is presented and motivated. The proposed control method penalizes statistical measurements based on their informative distance from real metrology data. This approach is also able to cope with the virtual loop control, in which the R2R runs for several process iterations without actual measurements, relying only on VM predictions. The results are compared with the current state-of-the-art by means of simulation studies based on realistic assumptions.

The results contained in this Chapter have been partially published in Susto et al. (2012e).

8.1 Introduction

Run-to-Run (R2R) has become the standard approach for process control in Semiconductor plants Boning et al. (1996); Toprac et al. (1999) in the last decades. Despite its simplicity, R2R control presents several advantages such as improved process and device performance, decreased tool downtime, improved process throughput, reduction of defective wafers and early detection of process drifts Anderson and Hanish (2007).

R2R techniques are based on physical measurements of quality parameters (such as layer thickness or Critical Dimensions). Considering the common sampling practices of measuring a small number of wafers for lot (usually 1 or 2 for 25 wafers), it is apparent how R2R controllers operate on a Lot-to-Lot (L2L) control policy that allows for corrective actions to be taken once for lot Toprac et al. (1999). With the development and diffusion of Virtual Metrology (VM) systems in the last years, this scenario has changed as control systems have the possibility to incorporate this new information source in their calculations.

The presence of statistical measurements for each wafer and the reduction of physical measure should be taken into account when implementing a control strategy. The research on the field has been particularly focused on the modeling part of VM, while not so much has been proposed concerning the interaction between VM and control systems. The present work has been mainly inspired by Cheng et al. (2008), that represents the current state-of-the-art in the field; in Cheng et al. (2008) VM and physical measurements are treated differently in dependence of their probabilistic distributions. In this Chapter, a section is dedicated to review such work and identify possible improvements and alternatives.

The main contribution of the work presented in this Chapter regards a novel approach for dealing with the mixture of actual and virtual measurements in a R2R loop; we take into consideration the distances, in an Information Theory framework, between the two types of measurements to penalize the VM predictions in a flexible way through a small set of configuration parameters. Furthermore, the proposed approach can also take in consideration the growing risks associated to unobserved change events: specifically, it is assumed that unrecognized state changes in the process tool can impact the overall prediction quality of the VM module.

The Chapter is organized as follows; as said, in Section 8.2 we briefly review the state-of-the-art of VM and control in a R2R loop. In Section 8.3 we present the statistical distance-based R2R control. In Section 8.4 we illustrate the comparison of different control approaches against the one proposed in this Chapter. Finally, Section 8.5 concludes the Chapter with some remarks on the presented work and some further developments.

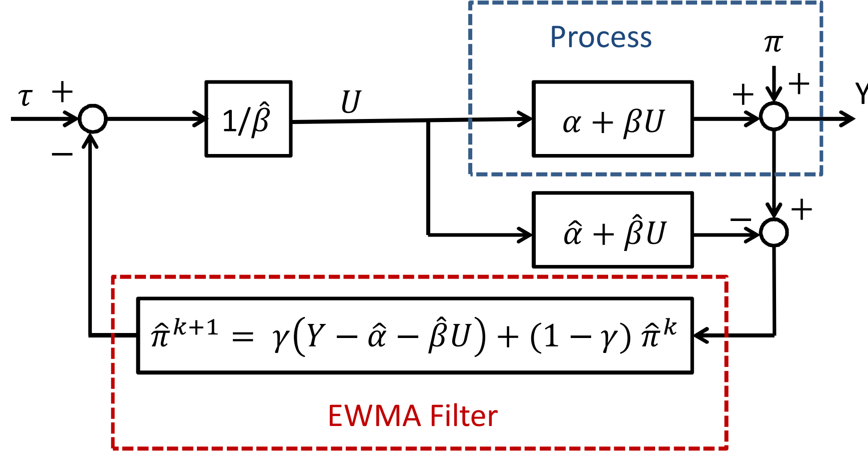


Figure 8.1: Scheme of the Run-to-Run control loop in exam.

8.2 Review on R2R and VM

The typical R2R controller in semiconductor manufacturing is based on Exponential-Weighted-Moving-Average (EWMA) filtering Patel and Jenkins (2000); for the sake of the presentation of the EWMA approach we consider the following linear process for the k-th iteration Wu et al. (2008)

$$Y^k = \alpha + \beta U^k + \pi^k, \tag{8.1}$$

where Y^k and U^k are process output and input, α and β are process bias and gain and π is the model offset. Let the process model estimate be

$$\hat{Y}^k = \hat{\alpha} + \hat{\beta} U^k, \tag{8.2}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the estimated process bias and gain and $\hat{Y}$ is therefore the estimated process output.

The EWMA filter estimates the model offset of the $k+1$-th run as

$$\hat{\pi}^{k+1} = \gamma \left(Y^k - \hat{\alpha} - \hat{\beta} U^k \right) + (1-\gamma)\, \hat{\pi}^k, \tag{8.3}$$

where $\gamma \in [0,1]$ is the EWMA coefficient. The optimal control action taken at iteration $k+1$ is

$$U^{k+1} = \hat{\beta}^{-1} \left(\tau - \hat{\alpha} - \hat{\pi}^{k+1} \right), \tag{8.4}$$

where τ is the process target. Such system is depicted in Figure 8.1.

With the classical approach, in which only actual metrology operations (performed either via scanning electron microscopes or in-situ sensors) concur in updating the controller, control actions are performed only after an entire lot is processed. Furthermore, it is well known that the delay induced by real metrology operations can make the EWMA controller less effective Wu et al. (2008). The introduction of VM measurements in the control loop can improve such scenario enhancing the control performances. As pointed out in Khan et al. (2007, 2008) in the control approach (8.4) the mixture of physical measurements Y_r and statistical predictions Y_s should be taken into account to achieve maximum accuracy. For this reason, (8.4) is modified by choosing different EWMA coefficients depending on the source of the measurement:

- if $Y^k = Y_r^k$ then

$$\hat{\pi}^{k+1} = \gamma_r \left(Y_r^k - \hat{\alpha} - \hat{\beta} U^k \right) + (1 - \gamma_r)\, \hat{\pi}^k; \tag{8.5}$$

- if $Y^k = Y_s^k$ then

$$\hat{\pi}^{k+1} = \gamma_s \left(Y_s^k - \hat{\alpha} - \hat{\beta} U^k \right) + (1 - \gamma_s)\, \hat{\pi}^k. \tag{8.6}$$

Given the fact that physical measurements are more reliable than statistical measurements, it is reasonable to choose $\gamma_r > \gamma_s$ Khan et al. (2007, 2008). Reasonably, the choice of γ_s should depend on the accuracy of the VM measurements: as such, the need for a criterion to evaluate the reliability of the statistical measurements arises.

In Cheng et al. (2008), such criterion takes the name of *Reliance Index* (RI). The RI is computed by considering the distributions of physical and statistical measurements. The computational system is based on two models:

- a *conjecture model*, the VM model that provides the statistical measurements;
- a *reference model* that simulates the physical measurements.

Under the hypothesis of normally distributed physical and statistical measurements, RI is defined as the integral of the intersection area between the Gaussian distribution $G(\mu_{Y_s}, \sigma_{Y_s}; x)$ of the virtual measurement Y_s and the Gaussian distribution $G(\mu_{\hat{Y}_r}, \sigma_{\hat{Y}_r}; x)$ of the reference model $\hat{Y}_r$. After a variance standardization step, RI is computed as

$$\mathrm{RI} = \int_{-\infty}^{\infty} \min \left\{ G(\mu_{Y_s}, 1; x), G(\mu_{\hat{Y}_r}, 1; x) \right\} dx. \tag{8.7}$$

It immediately follows that $\mathrm{RI} \in [0, 1]$, where the virtual metrology information is considered more and more reliable as RI approaches its upper bound (Figure 8.2).

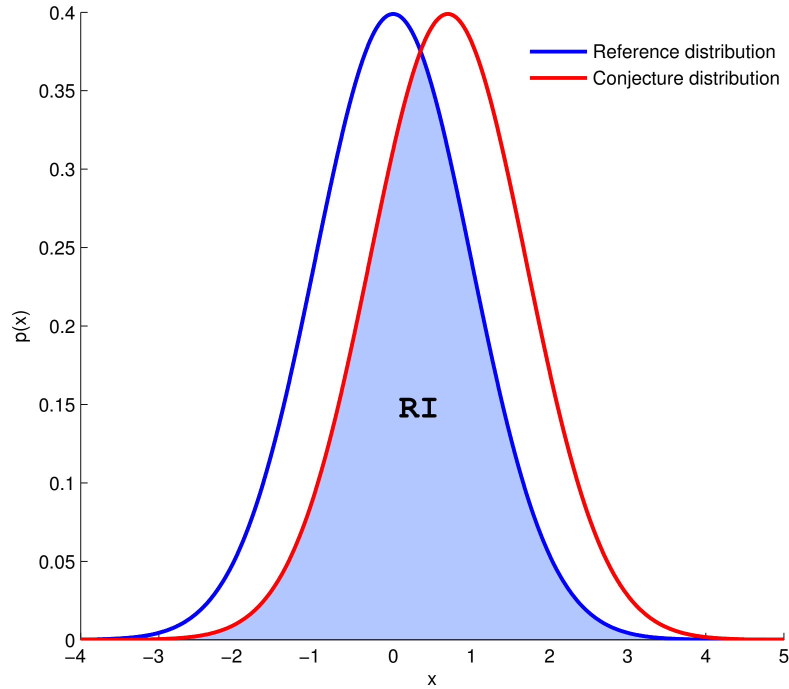


Figure 8.2: Univariate example of Reliance Index calculation

Once RI is computed, the relationship between γ_r and γ_s is obtained, as defined in Kao, Cheng, and Wu (2011), as

$$\gamma_s = \mathrm{RI}\gamma_r. \tag{8.8}$$

The coefficient RI is accompanied with *Similarity Indexes* (based on Mahalanobis distance) that allow to establish whether a new set of process observations is statistically similar to training dataset employed for VM (Kao, Cheng, Wu, Kong, and Huang).

While the described Reliance Index approach enjoys a number of appealing properties (above all, its parameterless nature), it also presents some weaknesses that leave room for further proposals: namely,

1. The Reliance Index RI arises from a static relation with respect to the difference between virtual measurement and reference models (Figure 8.3). In other words, the uncertainty information associated to both models is not taken in account.

2. The relationship between γ_s and γ_r is fixed after RI is calculated; in this way, new available information is not considered nor weighted appropriately against outdated data.

3. The considered metric, arising as the integral of shared area between two Gaussian

distribution functions, lacks an interpretation that can easily be linked to well established probabilistic or information metrics. While this is not necessarily an issue with respect to actual applications of the presented technique, the authors feel that investigating novel Reliance Indexes based on well known metrics would prove an advantage from the point of view of robustness.

Furthermore, it is known that change events can happen in a process tool and affect the quality of VM predictions (Figure 8.4). For this reason, it would be appealing to define an evolving dynamic to model the increasing risk of relying on a long series of virtual measurements. In the remainder of the Chapter, this phenomenon will be referred to as *model degradation*.

In the next Section, a new dynamic Reliance Index approach is proposed to cope with the above stated requirements.

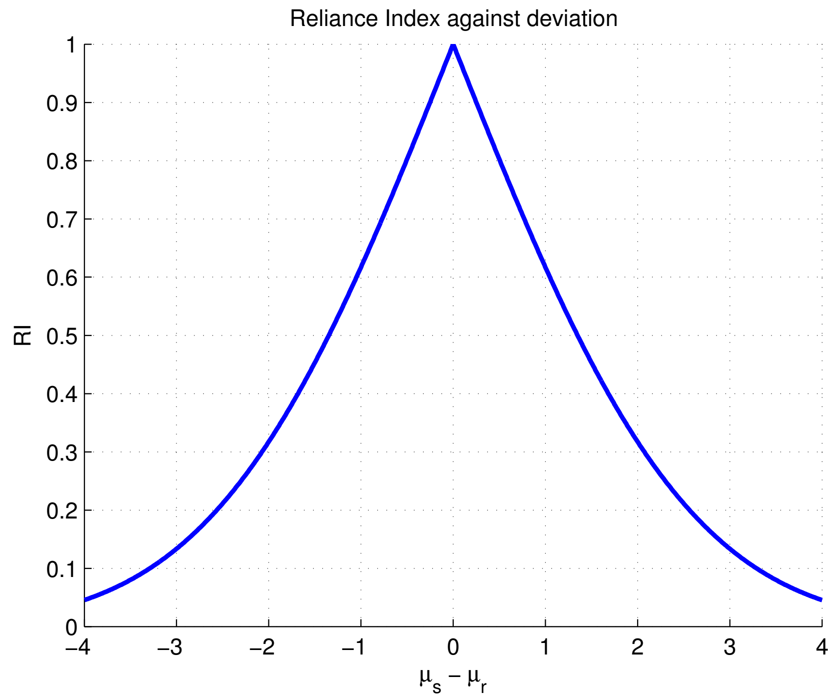


Figure 8.3: Evolution of RI in dependence on different accuracy levels

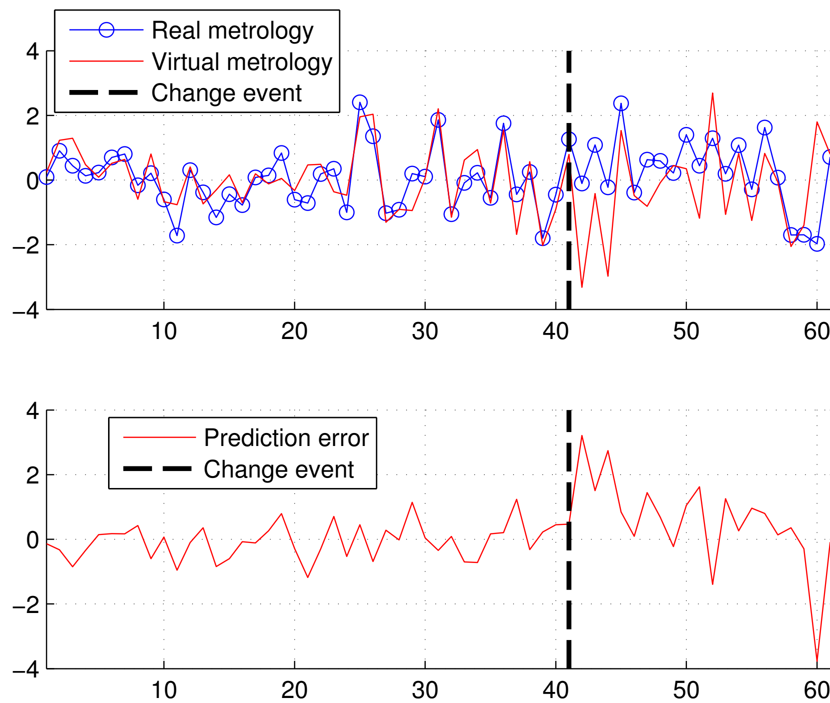


Figure 8.4: Example of VM degradation over time: the change event happening at $t = 40$ increases the error variability of the VM module.

8.3 The proposed approach

In this section, a continuous update strategy for γ_s is described and motivated. Assuming that a virtual measurement will be provided for every processed wafer, let us consider a sequence of time points in which new metrology information is available. Such events can belong to two mutually exclusive categories:

- Both a virtual and a real measurements are available (closed loop or, in short, $\mathcal{C}_c$)
- Only a virtual measurement is available (virtual loop or $\mathcal{C}_v$)

In the proposed approach, an event of type $\mathcal{C}_c$ will cause a *closed loop* update of the model parameters, while an event of type $\mathcal{C}_v$ will trigger a *virtual loop* update. At a generic time k, let the virtual measurement be

$$v^k \sim \mathcal{N}(\mu_v^k, \Sigma_v^k) \tag{8.9}$$

where all the quantities are provided by the existing VM tool, and the real measurement (whenever available) be

$$r^k \sim \mathcal{N}(\mu_r^k, \Sigma_r^k) \tag{8.10}$$

For (8.9) and (8.10), all the dimensions are compatible with p-variate quantities. It should be noted that the real measurement model usually results from a gage study performed on the real measurement tool - in most cases, it would be safe to simplify such model as

$$r^k \sim \mathcal{N}(\mu_r^k, \sigma_r^2 I_p)$$

where σ_r^2 is the estimated measurement error variance and $I_p \in \mathbb{R}^{p \times p}$ is the identity matrix. Nevertheless, the more generic form (8.10) will be used in the next subsections to derive the update rules.

Information Theory Measures

By relying on Information Theory (IT) metrics, it is possible to obtain meaningful indexes of the probabilistic distance between observations, as well as the amount of information entropy that is associated to a specific random variable; it is well known that such measures yield a more complete information with respect to statistical moments and other probabilistic measures. For these reasons, and thanks to the increasing

computational capabilities, IT instruments are becoming more and more used in model identification Principe, Xu, and Fisher (2000), control systems Touchette and Lloyd (2000) and decision making tasks Moreno, Ho, and Vasconcelos (2003). Since the amount of available information is different between $\mathcal{C}_c$ and $\mathcal{C}_v$ updates, it is necessary to employ two different metrics to penalize bad VM performances. Specifically, the selected criteria should enjoy the following properties:

- Low VM quality should result in a decrease of the weighting parameter γ_s.
- Underestimation and overestimation of confidence boundaries should be penalized in a similar fashion.

The Kullback-Leibler Divergence (KLD) is a non-symmetric measure, based on information theory, of the difference between two distributions f and q, and is defined as

$$D_{KL}(f||q) = \int_{\mathcal{S}} f(x) \log \frac{f(x)}{q(x)} dx \tag{8.11}$$

where $\mathcal{S}$ is the common support of $f(x)$ and $q(x)$. In (8.11), $f(x)$ and $q(x)$ are the probability distributions of f and q. In such formulation, f represents the "real" model, while q is an estimated model - with reference to the Virtual Metrology evaluation problem, the role of q is filled by the virtual measurement v, while the real model f is represented by the actual measurement by r. Notably, the following equivalence holds for D_{KL}:

$$D_{KL}(f||q) = H(f,q) - H(f)$$

that is, the Kullback-Leibler distance is the difference between the cross-entropy of f and q and the Shannon entropy of the real model f. Furthermore, $H(f)$ is defined as

$$H(f) = -\int_{\mathcal{S}} f(x) \log f(x) dx$$

and represents the expected amount of information contained within the random variable f; notably, both $D_{KL}(f||q)$ and $H(f)$ are always positive. At a given time $k+1$, the Kullback-Leibler Divergence between r^{k+1} and v^{k+1} can be computed in closed form as

$$\begin{aligned} D_{KL}(r^{k+1}||v^{k+1}) = &\frac{1}{2}\mathrm{tr}\left(\Sigma_v^{-1}\Sigma_r\right) + \\ &+ \frac{1}{2}\left((\mu_v - \mu_r)^\top \Sigma_v^{-1}(\mu_v - \mu_r) - \ln\left(\frac{|\Sigma_r|}{|\Sigma_v|}\right) - p\right) \end{aligned} \tag{8.12}$$

while the Shannon entropy associated to v^{k+1} is

$$H(v^{k+1}) = \frac{1}{2}\log|\Sigma_v| + \frac{p}{2}(1+\log(2\pi)) \tag{8.13}$$

Note: the time apices are omitted in (8.12) and (8.13) for the sake of readability.

The update rules

Given such definitions, it is possible to define the following update rules: when $\mathcal{C}_c$,

$$D^{k+1} = (1-\lambda_D)D^k + \lambda_D D_{KL}(r^{k+1}||v^{k+1}) \tag{8.14}$$

$$\delta_s^{k+1} = 0 \tag{8.15}$$

and when $\mathcal{C}_v$,

$$D^{k+1} = D^k \tag{8.16}$$

$$\delta_s^{k+1} = (1-\lambda_s)\delta_s^k + \lambda_s H(v^{k+1}) \tag{8.17}$$

At a given time $k+1$ the smoothing parameter γ_s^{k+1} is then obtained as

$$\gamma_s^{k+1} = \underbrace{e^{-(\epsilon_D D^{k+1} + \epsilon_s \delta_s^{k+1})}}_{\mathrm{RI}_{\mathrm{new}}^{k+1}} \gamma_p$$

The update rules described in (8.14)-(8.17) enjoy some interesting properties:

1. The reliability assigned to the VM model is updated every time a comparison is available ($\mathcal{C}_c$) by using (8.14)

2. The virtual loop ($\mathcal{C}_v$) dynamic is handled by a weighted update of the Shannon Entropy associated to the virtual measurements. It is easy to see that the penalization term δ will tend to a regime value if the uncertainty associated to virtual measurements is fixed in the predictive model.

3. The virtual loop reset equation (8.15), that is triggered whenever a closed loop update occurs, allows to restore the reliance index $\mathrm{RI}_{\mathrm{new}}$ to a point in which only the performances of the VM module are considered for penalization. In other words, the Reliance Index $\mathrm{RI}_{\mathrm{new}}^{k+1}$ will have an instantaneous jump towards higher values if the prediction quality is confirmed by the D_{KL} test.

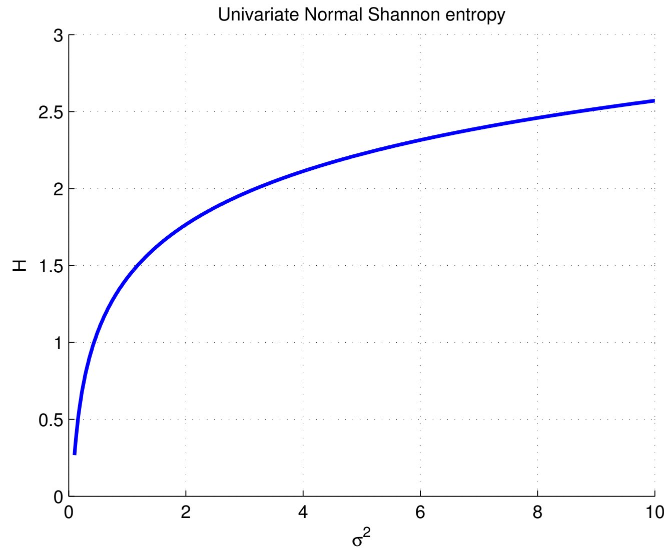


Figure 8.5: Evolution of the univariate Shannon entropy H for a Gaussian random variable, in dependence on the considered variance

It is worth noting that the penalizations induced by the two different set of updates rules go in opposite directions: this is needed in order to penalize every type of inaccurate self-assessment coming from the VM module. For instance, the Kullback-Leibler distance tends to penalize more, as the difference between real and virtual measurement grows, an under-assessment of the VM module's prediction error (Figure 8.6); on the other hand, the overestimated prediction covariance will be penalized during the Virtual Loop iterations (Figure 8.5). The bottom line is that the VM tool needs, to grant minimal coefficient penalization, to provide a realistic assessment of its own uncertainty as well as precise punctual prediction values.

Parameters Tuning

In order to provide maximum flexibility, a set of 4 parameters can be tuned to adapt the proposed model to a specific penalization setting. The parameter set consists of:

- the forgetting factors $\lambda_D \in [0, 1)$ and $\lambda_S \in [0, 1]$;
- the penalization coefficients $\epsilon_D \in \mathbb{R}^+$ and $\epsilon_s \in \mathbb{R}^+$.

While a full tuning of such parameters should arise as the result of a DOE, in this subsection we provide some intuitive guidelines.

- The ratio ϵ_s/ϵ_D should be chosen to enhance the penalization contribution D^{k+1} or δ_s^{k+1}. The trade-off between the relevance in $\mathrm{RI}_{\mathrm{new}}$ of the two terms D^{k+1}

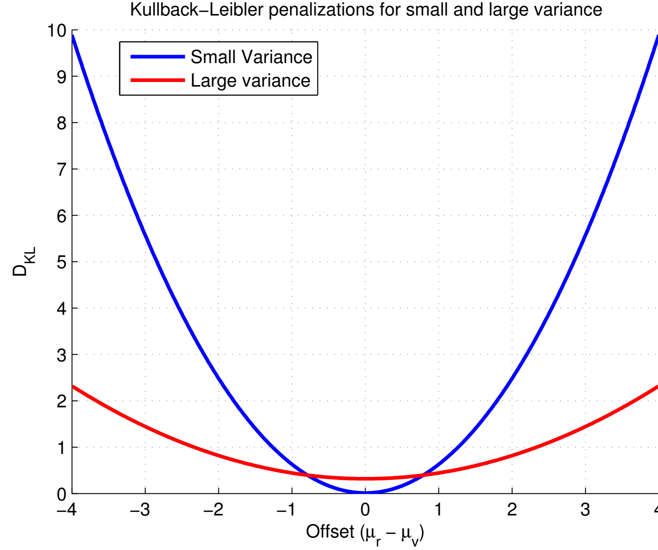


Figure 8.6: Calculated Kullback-Leibler distances for different levels of accuracy; hereby an underestimated (blue) and overestimated (red) variances are considered

and δ_s^{k+1} must be tailored to the process variability. For example, if the VM prediction accuracy tends to deteriorate after some virtual metrology loops (this can be detected if the control action is usually strongly changed after the arriving of a new measure), then the ratio ϵ_s/ϵ_D should probably be high. Once the ratio is tuned ϵ_D or ϵ_s is chosen in order to make $\mathrm{RI}_{\mathrm{new}}$ small enough when inaccurate VM predictions are provided.

- λ_S should depend on the number of unmeasured wafer and should be low if predictive model degradation is an issue.
- λ_D should be designed to react properly upon unexpected drifts in process quality; hence, it should be tuned to be compatible with the time constants of drifts observed in the past.

8.4 Experimental Results

In this Section we test the proposed Information Theory (IT) based approach against other R2R control philosophies, namely

- *Lot-To-Lot* (L2L), where the control is update every time a new physical measure is available (in common practice this happens once or twice per lot).

- *Complete Virtual Metrology* (VM), where the statistical measures are trusted as the physical ones and the control law is updated at every step with $\gamma_r = \gamma_s$.
- *Reliance Index-based* (RI), as described in Section 8.2.

We consider a single-input single-output system (SISO), with the same structure depicted in Section 8.2 with the following parameters[1]

$$\alpha = 1 \qquad \beta = 1$$
$$\hat{\alpha} = 1 \qquad \hat{\beta} = 1.2$$

In the following simulations the target of the CD is $\tau = 10$; we consider a typical semiconductor production where

−25 wafer are processed for each lot,

−1 wafer is measured per lot (the one in slot 1).

To perform a fair comparison between the various approaches, the VM module that provides the statistical measures is the same. We consider here a VM system whose prediction error has a Gaussian distribution with variance $\sigma_{\text{VM}} = 3$ and bias $\mu_{\text{VM}} = 0.3$, if not differently stated.

We performed for each experiment N simulations, where we let the system evolves for K iterations. The performances of the various algorithms are compared by considering the Root Mean Square Error (RMSE)

$$\text{RMSE} = \frac{1}{NK} \sum_{j=1}^{N} \sum_{k=1}^{K} \left(\tau - y^j(k)\right)^2 ,$$

where $y^j(k)$ is the CD for the k-th process step for j-th simulation.

The following experiments differ for the modeling of the output noise π.

Non-correlated, biased noise

In the first experiment we choose an i.i.d. (independent and identically distributed) output noise ($\pi_s \perp \pi_t, \forall s \neq t$) with Gaussian distribution of variance $\sigma_\pi = 2$ and bias $\mu_\pi = 1.3$. We have performed $N = 1000$ simulations, each one of $K = 5000$ steps long (200 lots of 25 wafers).

[1]Given the control scheme in exam, there is not difference in introducing a mismatch in α coefficient and its estimation $\hat{\alpha}$ or introducing noise bias. There is therefore no loss of generalization in considering $\hat{\alpha} = \alpha$.

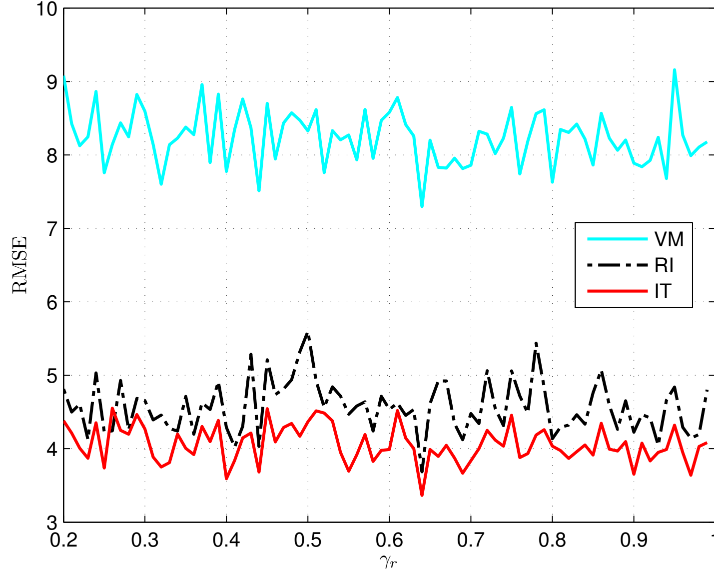


Figure 8.7: Evolution of RMSE with different approaches (VM, RI e IT) for different values of the parameter γ_r (The RMSE for the L2L is always above 13 and it is not reported here to enhance figure interpretability).

In Figure 8.7 the results of the experiment are summarized, reporting the RMSE associated to different choices of the forgetting factor γ_r. It can be seen how RI and IT-based control approaches outperforms the other methods.

Auto-correlated, unbiased noise

In this second experiment we model the noise in a different way, as an auto-correlated noise

$$\pi_k = \pi_{k-1} + \bar{\pi}_k,$$

where $\bar{\pi}_k \sim \mathcal{N}(\mu_\pi, \sigma_\pi)$.

The previous equation describes physical processes in which trends are present; this is typical of semiconductor processes where few maintenance operations are performed and the tool behavior tends to drift until the machine is again properly maintained.

Figure 8.8 reports the evolution of one simulation with the aforementioned settings. It can be clearly appreciated how in this case the production stays closer to the target with RI and IT approaches than complete VM and L2L (as shown in the first panel of Figure 8.8); this is due to a more stable and accurate estimation of π (second panel of Figure 8.8).

Figure 8.9 depicts a new experiment, with the same process and noise settings as

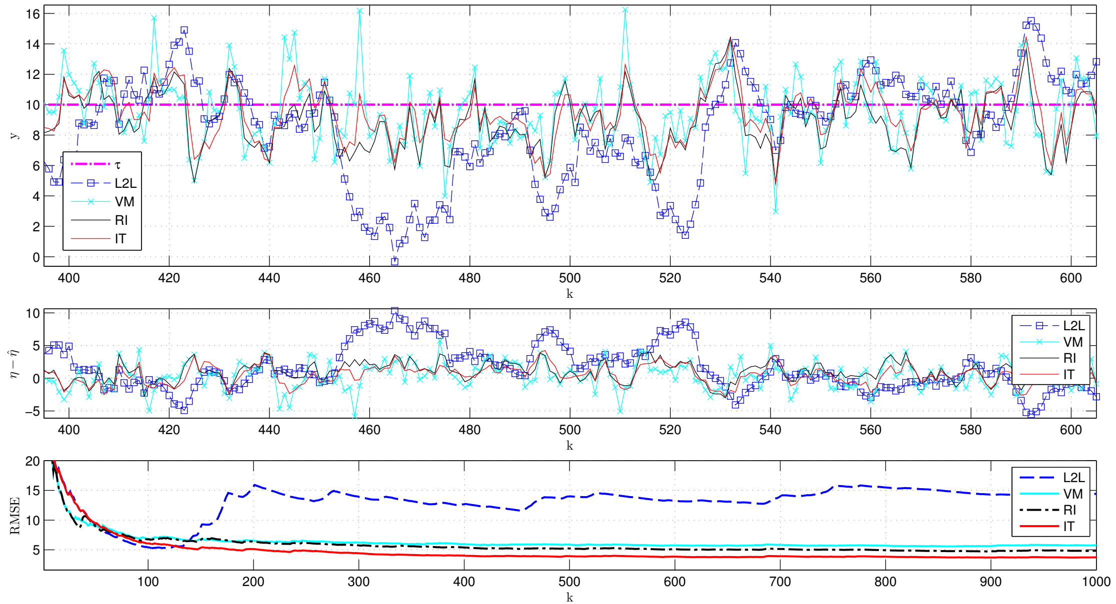


Figure 8.8: Example of the behavior of different approaches. In the three panels are depicted, respectively, the evolution of the CD y, of the noise estimation error $(\pi - \hat{\pi})$ and of the RMSE. In this particular case the RMSE at the end of the simulation were $\mathrm{RMSE_{L2L}} = 14.455$, $\mathrm{RMSE_{VM}} = 5.754$, $\mathrm{RMSE_{RI}} = 4.83$ and $\mathrm{RMSE_{IT}} = 3.743$.

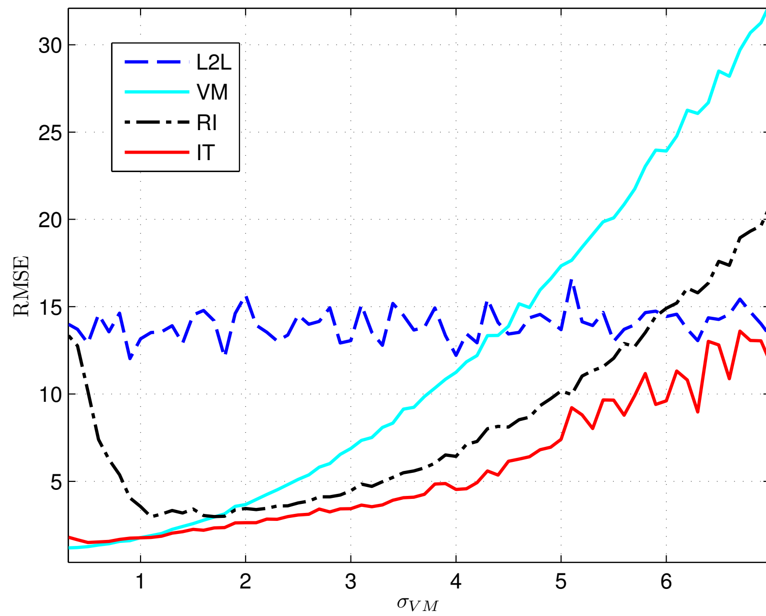


Figure 8.9: Performances of the various R2R policies at the change of Virtual Metrology prediction accuracy variance σ_{VM}.

described before, for different values of VM variance σ^2_{VM}. In this case, regardless of the quality of VM predictions, the IT algorithm outperforms the other methods. It can be seen how the full VM approach depends too much on VM quality, and, for high variance VM predictions, it behaves worse than a simple L2L approach, accordingly to the qualitative principle that no information is better than inaccurate information. It can be seen how the RI approach accuracy is not just proportional to VM precision; in the case of really accurate VM predictions (σ_{VM} low), the RI approach is penalized by the mismatch between the conjecture model distribution (VM model) and the reference model. However, it should be remarked that for standard values of the VM variance ($2 < \sigma_{\mathrm{VM}} < 5$), the RI approach has close performances to the IT algorithm without the parameters tuning required by the IT approach.

The robustness shown in this experiment is, in our consideration, the most appealing quality of the proposed IT algorithm.

8.5 Conclusions

In this Chapter, a new approach firstly presented in Susto et al. (2012e) for penalizing VM predictions with respect to in actual measurements in a R2R control loop has been described. The proposed approach is based on two well known Information Theory metrics, namely the Kullback-Leibler Divergence and the Shannon Entropy. Specifically, the former is used to assess the prediction quality of a specific VM tool, while the latter allows to reduce the risk of relying only on virtual measurements for a long period. The proposed methodology has been tested against the state-of-the-art approach based on the Reliance Index, and it has been shown to guarantee better performances at the cost of tuning a set of parameters.

It has in fact been shown that the proposed approach outperforms other R2R control policy with different critical settings, namely

- autocorrelated noise, typical of semiconductor processes where few cleanings on a machine are performed;
- poor VM predictions, situations that may happens when products with few historical data are available or when fault events happen on the tool.

The next developments of this work will include the information regarding the statistical distance of the input data, as well as a more systematic way of tuning the four parameters that define the behavior of the proposed algorithm. Furthermore, a test of the proposed control system in a semiconductor environment will be performed to validate the presented approach in a real environment.

Part IV

Predictive Maintenance

9

Predictive Maintenance for Epitaxy

Predictive Maintenance (PdM) systems have already been introduced in Section 1.4; this part of the thesis will be dedicated to some innovative PdM modules developed:

- this Chapter will be dedicated to a PdM system for Epitaxy developed in collaboration with Infineon Technologies Austria, AG;
- in Chapter 10 a PdM system for Ion-Implantation done in collaborations with STMicroelectronics will be discusses.

Silicon Epitaxial Deposition is a process strongly influenced by wafer temperature behavior, that has to be constantly monitored to avoid the production of defective wafers. However, temperature measurements are not reliable and the sensors have to be appropriately calibrated with some dedicated procedure. A Predictive Maintenance (PdM) System is proposed here with the aim of predicting process behavior and scheduling control actions on the sensors in advance. Two different prediction techniques have been

employed and compared: the Kalman predictor and the Particle Filter with Gaussian Kernel Density Estimator. The accuracy of the PdM module has been tested on real industrial production datasets.

The Chapter is organized as follows. In Section 9.1 a more detailed introduction of the maintenance problem in exam is provided, while in Section 9.2 the epitaxy equipment and the fab data are described. In Section 9.3 the problem is formalized mathematically and the PdM module is presented. A comparison between the performance of the filtering and prediction approaches tested is given in Section 9.4 where the application of the PdM algorithms to fab data is discussed. A regression approach to estimate the best control action to the maintenance issue is described in 9.5. Concluding remarks are given in Section 9.6.

The results discussed in this Chapter have been adapted and extended from Susto et al. (2011a, 2012b) and Susto et al. (2012a).

9.1 Introduction

In this Chapter we describe how statistical and filtering techniques can be employed to develop a PdM tool with application to Silicon Epitaxial Deposition (Epitaxy) processes.

Epitaxy is the process of growing, usually through Chemical Vapour Deposition (CVD), a thin layer of single-crystal silicon over a single-crystal silicon substrate. The crystal growth during Epitaxy strongly depends on the process temperature (Cheng, 1997).

The epitaxy process is divided into two main phases as illustrated in Fig.9.1:

- the *warm-up* phase, where the wafer and the supporting plate (susceptor) are heated until a suitable temperature is reached;
- the *deposition* phase, where the epitaxy layer is grown on the wafer surface.

The wafer temperature is driven by 4 groups of heating lamps (see Fig.9.2) and it is sensed by two infrared thermometers, called pyrometers. To avoid deformations on the wafer, it is important that during the warm-up phase the susceptor and the wafer temperatures are increased homogeneously. The susceptor is made of isostatic compressed graphite that has different heat conductivity characteristics with respect to silicon, and, for a given heating power, reaches lower temperatures with respect to silicon. For this reason, a control loop based on the reading of the two pyrometers is used to guarantee that the temperatures of the two sides of the wafer are equal during the warm-up phase by appropriately managing the heating power of the lower and upper lamps.

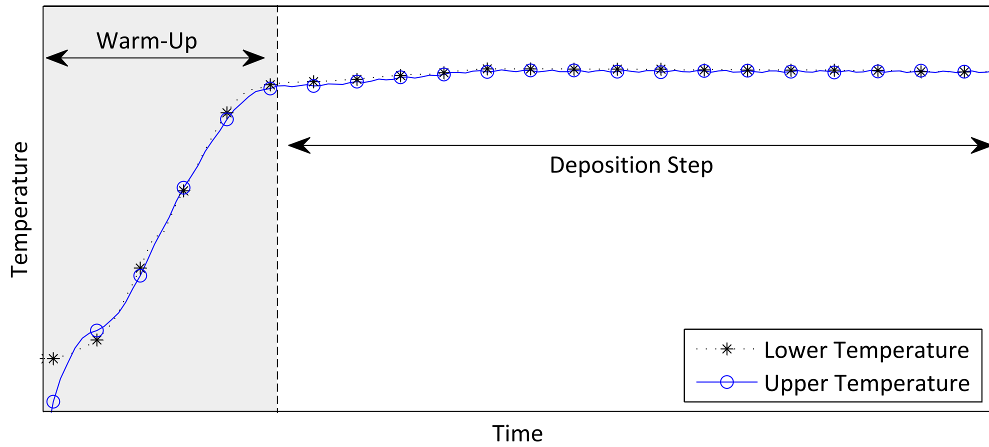


Figure 9.1: Wafer temperature evolution during the Epitaxy process

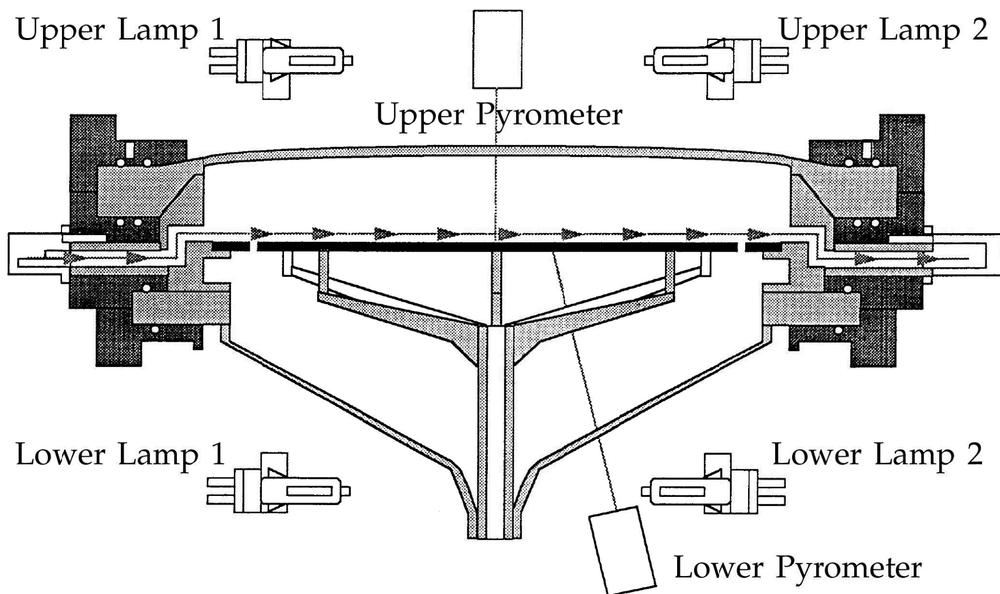


Figure 9.2: Lateral section of the Epitaxy machine

It is therefore straightforward to understand why temperature readings should be precise and reliable. However, the reading of the lower pyrometer drifts in time from the real value so that the pyrometer must be calibrated to ensure that the readings of the two pyrometers are the same. For this reason, the temperature difference reading at the end of the deposition step, when the temperatures on the two sides of the wafer are supposed to be equal, is constantly monitored. The reading of the temperature difference on the two sides of a wafer has to be kept inside given confidence thresholds otherwise control actions are taken when limits are violated, in a typical CBM approach. This is a tedious issue for process engineers who have to monitor the temperature difference and to frequently perform the calibration on the machine.

This procedure has several drawbacks:

- *lack of planning*: interventions on the equipment are performed in reaction to alarms only;

- *operator subjectivity*: actions are taken mostly on the basis of operators experience. An automatic procedure is desired to avoid such dependency on operator subjectivity;

- *lack of knowledge of machine state*: tools for visualizing and monitoring the behaviour of the machine in relation to the maintenance events are not available to process engineers. Furthermore, recurring adjustments prevent the operator to effectively detect drifts and time trends in the data.

To overcome the above mentioned problems it is necessary to move from CBM to PdM, so as to provide process engineers with a reliable schedule of maintenance actions. In this Chapter, we discuss how a PdM module can be designed on the basis of an algorithm for predicting wafer temperature behaviour. We propose here a tool that allows the end-user to predict out-of-control situations in the epitaxy equipment, in a probabilistic framework. The PdM module provides a list of machines potentially close to out-of-control situations, ordered in terms of a probabilistic index (health factor) associated with the outcome of the predictors, to better use the resources dedicated to the equipment monitoring activity.

Due to the lack of an *a priori* knowledge on the process evolution from a statistical point of view, the prediction of temperature behavior is tackled by taking two different approaches, based respectively on:

- the *Kalman Predictor* (Kalman, 1960), that, under Gaussian noise assumptions, is the optimal linear predictor in terms of minimum variance of the prediction error;

- the *Particle Filter* (Arulampalam et al., 2002) in combination with a *Gaussian Kernel Density Estimator*, that can be used in presence of arbitrary noise distributions but it is more computationally demanding than the Kalman Predictor.

Particle Filters have been recently employed with success in the PdM for semiconductor manufacturing processes. In Schirru et al. (2010b) Particle Filters have been used to estimate a Gamma process that, being non-negative and with non-negative derivative, is a suitable distribution for defining a health factor. In Butler and Ringwood (2010) Particle Filters are employed in combination with Gaussian Mixture Models to estimate the Remaining-Useful-Life of a semiconductor manufacturing equipment.

9.2 Equipment and Data Description

As anticipated in the Introduction, the Epitaxy (EPI) process strongly depends on wafer temperature (Jaeger, 2001). The epitaxial growth is kept stable by monitoring the median of the temperature differences of a batch of wafers (usually 8 wafers)

$$x = T_{\mathrm{DOWN}} - T_{\mathrm{UP}}, \tag{9.1}$$

where T_{DOWN} and T_{UP} are the temperatures on the two sides of the wafer at the end of the deposition step.[1], and, when necessary, using a control input u to keep the evolution of x between an Upper Control Limit (UCL) and a Lower Control Limit (LCL). The proposed PdM system predicts the behavior of x and suggests if and when a correction action via u has to be taken.

Inside a batch the sensed temperature difference from wafer to wafer changes drastically (as can be seen in Fig. 9.3). This is due to the fact that, after every batch, a small clean on the tool is performed and chamber conditions change. For this reason, wafer-to-wafer monitoring of the temperature difference, even if feasible, is not significative. A single value for each batch, that may be of different size (in Fig. 9.3 we have two batches with 8 wafers and 1 with 7), is therefore more adequate for monitoring purposes. The median of the temperatures in each batch is then chosen in accordance to fab practice.

The EPI tool considered here (depicted in Fig.9.4) is composed of three independent chambers that can be considered as different machines. Several recipes run on the same machine, causing different levels of stress.

[1] To avoid confusion, from now on the temperature difference reading is the value described by x (the temperature difference at the end of the deposition step) and not the FDC temporal profile (as depicted in Fig. (9.1)).

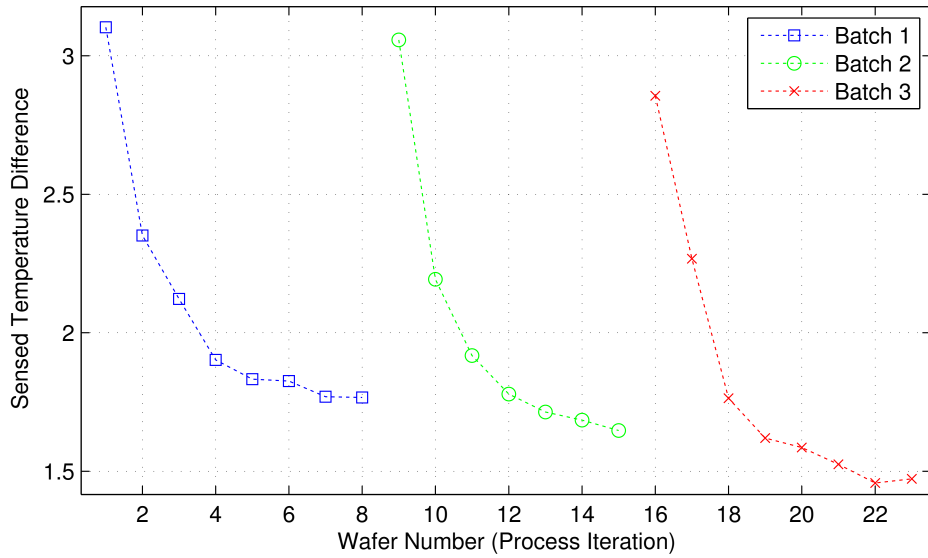


Figure 9.3: Example of the temperature differences sensed at the end of the EPI deposition step for different wafers

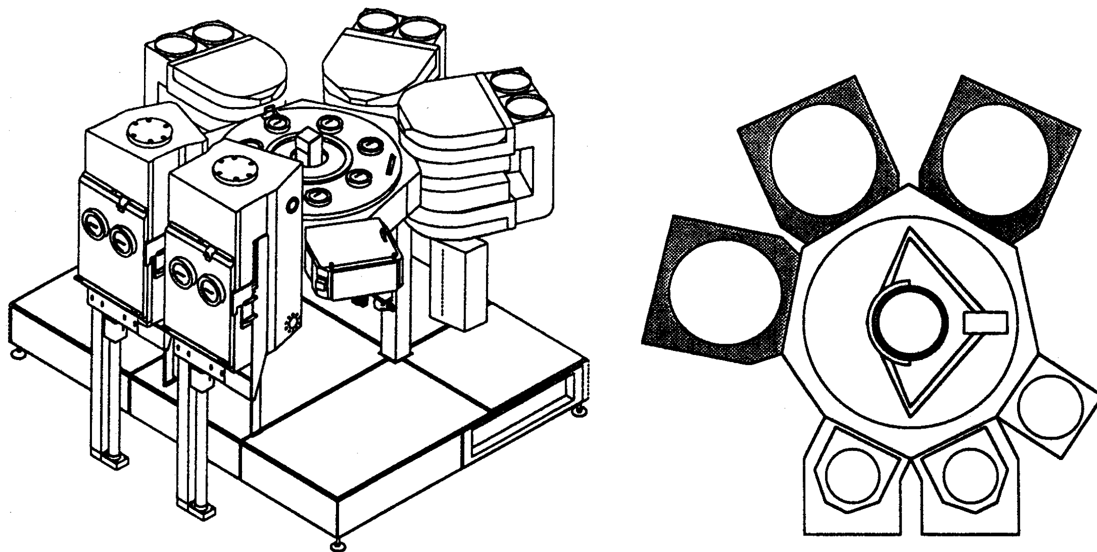


Figure 9.4: The Epitaxy equipment: scheme and horizontal section

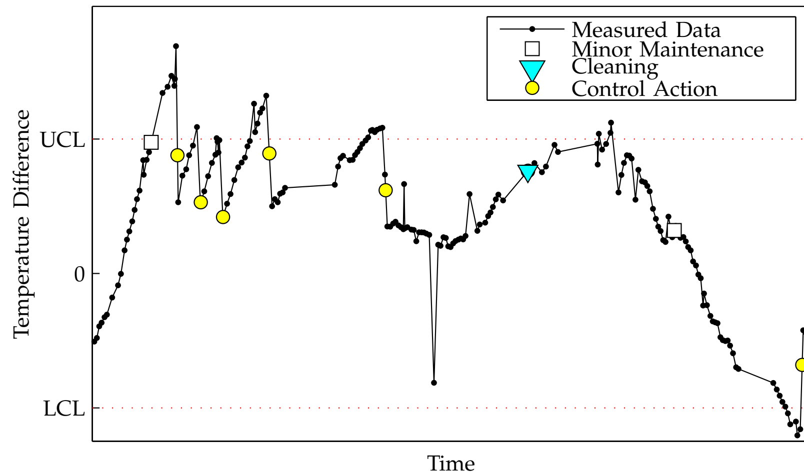


Figure 9.5: Example of trend of measured batch temperature difference at the end of the deposition step

For the sake of clarity, we consider in the following the case of only one equipment with only one recipe. However, the approach is scalable and can be employed also in presence of more recipes running on the same machine.

Available data from the equipment consist of measurements of $x = T_{\text{DOWN}} - T_{\text{UP}}$ (although not at a constant sampling rate) and information on wet cleans, control actions, and other minor maintenance interventions. A typical time behaviour of the temperature difference variable is shown in Fig.9.5, where trends in the evolution can clearly be observed. The presence of such trends is associated by process experts with the asymmetric structure of the tool chambers.

Process evolution is characterized also by the presence of outliers and abrupt change-points as shown in Fig.9.6. Abrupt changes are defined in Basseville and Nikiforov (1993) as any change in the parameters of the system that occurs either instantaneously or at least very fast with respect to the sampling period of the measurements. Here we define *abrupt changes*, or *change points* (Barry and Hartigan, 1993), the drifts or jumps in data that are not clearly associated with known operations on the tool, like cleanings, emissivity adjustments or other maintenance actions.

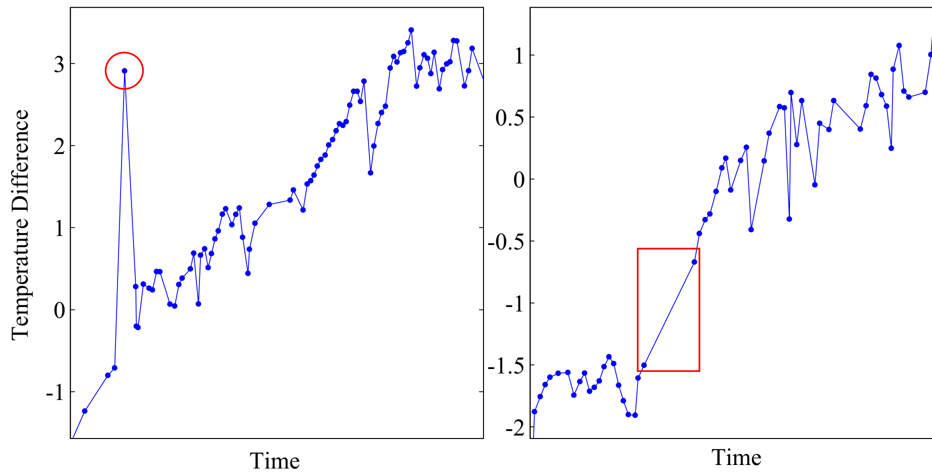


Figure 9.6: An outlier (left) and a change-point example (right) in measured batch temperature difference at the end of the deposition step evolution

9.3 Problem Formalization and Proposed Module

To move toward the definition of a PdM approach for the EPI process, it is necessary to deal with the problem of predicting, with some confidence level expressed in a probabilistic way, the evolution of the temperature difference x, so as to have indications on when such variable will be outside the given control limits and consequently take corrective actions.

In the following subsections we show how to tackle such problem with filtering and prediction algorithms. We present two approaches, based on the Kalman Predictor (Section 4.2) and the Particle Filter (Section 4.3), respectively.

We observe that it is a common approach in Semiconductor Manufacturing to tackle filtering and control problems by using EWMA-based algorithms (Chen and Guo, 2001). It is known that EWMA can also be used for prediction problems (Ramjee and N.Crato and, 2002), such as the one described in this Chapter. However, it is a classical statistical result (Cox, 1961) that the optimal linear predictor (as the Kalman Predictor described in Section 4.2) guarantees better prediction performance in terms of mean square errors than EWMA, with similar computational and implementation efforts, thus motivating the approach followed in this Chapter.

Assumptions and Modeling

We make two basic assumptions on the EPI process

Assumption 9.3.1. The evolution of x is event driven (i.e. it is determined by the equipment usage and it does not depend on the continuous time variable). □

Under Assumption 9.3.1 it is possible to consider the evolution of x as a discrete-time system x_k, where the index k indicates the number of batches processed on the tool.

Assumption 9.3.2. The evolution of x can be approximated by

$$x_{k+1} = x_k + \Delta x_k + v_k + b u_k, \tag{9.2}$$

where Δx_k indicates the difference between x_k and x_{k-1}, v_k is the model noise, $v_k \sim g(x)$, $g(x)$ is an unknown continuous probability density function (pdf) on $\mathbb{R}$, u_k is the change of the emissivity coefficient in the pyrometers and the coefficient b represents the relationship between the temperature reading and the emissivity change. The value of b has been provided by engineers on the basis of on-the-field experience and it is $b = 1000$.

The relationship between emissivity adjustments and jumps in temperature readings for the tool in exam has been established during the years by trial and error done by process engineers. This relation has been further investigated in Susto et al. (2012b) with multi-tasking regression techniques and will be discussed in Section 9.5.

The observed temperature difference Y can be defined by

$$Y_k = x_k + w_k, \tag{9.3}$$

where $w_k \sim \mathcal{N}(0, R)$. We consider the model noise v_k and the sensor noise w_k to be uncorrelated, i.e. $\mathbb{E}[v_t w_t] = 0$. □

Assumption 9.3.2 reflects the usual behavior of x of drifting in a given direction (see Fig. 9.5). The evolution of the system can therefore be described by the following linear model

$$z_{k+1} = \underbrace{\begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}}_{A} z_k + \underbrace{\begin{bmatrix} b \\ 0 \end{bmatrix}}_{B} u_k + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{G} v_k \tag{9.4}$$

$$Y_k = \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} z_k + w_k, \tag{9.5}$$

where $z_k = \begin{bmatrix} x_k & x_{k-1} \end{bmatrix}^T$.

As can be seen from equations (9.4)-(9.5), the process is Markovian of order one in the variable z as required by the Bayesian approach described in Section 4.1.

The simple linear model (9.2) has been derived on the basis of the analysis of process data characteristics. However, a different data-driven modeling approach can been

adopted by discarding Assumption 9.3.2 and assuming that matrices A, B, C and G do not have an a priori structure.

In order to estimate elements and dimensions of the unknown matrices A, B, C and G from the available historical time series of $Y(\cdot)$ and $u(\cdot)$, several identification algorithms can be employed. Among these, *subspace methods* (Larimore, 1990) are based on the idea that at time t the state space can be constructed as the space spanned by the oblique projections of the future outputs $\{y_{t+k}; k = 0, 1, \ldots\}$ onto the joint past inputs $\{u_s, v_s \text{ with } s < t\}$ and the future joint inputs $\{u_s, v_s \text{ with } s \geq t\}$, see Chiuso and Picci (2005) for further details. The aforementioned class of algorithms also goes under the acronym of CCA to underline that the state reconstruction is performed by using Canonical Correlation Analysis (Chiuso, 2007).

Standard algorithms to perform subspace identification are N4SID and MOESP, we refer the interested reader to VanOverschee and Moor (1996) for details regarding their implementation. Standard algorithms to perform subspace identification are N4SID (VanOverschee and Moor, 1994) and MOESP (Verhaegen, 1995), we refer the interested reader to VanOverschee and Moor (1996) for details regarding their implementation.

By using subspace identification methods, we derived a completely data-driven model, as opposed to the approach used to derive (9.2) that relies on a priori physical assumptions. A comparison of the performances obtained with the two approaches is reported in Section 9.4.

Gaussian Kernel Density Estimator

Based on whether the distribution $g(x)$ is Gaussian or not, different approaches may be used to solve the filtering and prediction problem for the linear system (9.4)-(9.5). Particle Filtering (Section 4.3) is a technique that guarantees good prediction performance for any distribution of the model noise, however it is computationally demanding. If the Gaussian Assumption hold, we suggest to employ the Kalman Predictor that guarantees good prediction performances with less computational effort. Here we propose an approach, described in Algorithm 8, where we estimate $g(x)$ by using a Gaussian Kernel Density Estimator (KDE) (Jones et al., 1996), see Chapter 4.4. Then, based on the outcome of the estimation, we decide which method is to be used.

To discriminate if a distribution is Gaussian or 'statistically close to a Gaussian', visual inspection is the first approach. Otherwise, a rough but systematic approach, is that of performing a *normality test*; one of the most popular is the *Kolmogorov-Smirnov Test* (Lilliefors, 1967). With a normality test we can assess if the assumptions for the Kalman Predictor are satisfied or if we need to employ a more sophisticated approach as

Algorithm 8: PdM Module for Epitaxy

Data: Measure data $Y_{k:1}$

(1) Obtain an Estimation ($\hat{g}(x)$) of $g(x)$ through Gaussian KDE

if $\hat{g}(x)$ *is Gaussian (or statistically close to a Gaussian)* **then**

  (2.a) use Kalman Predictor (Section 4.2)

else

  (2.b) use Particle Filter (Section 4.3) with $\hat{g}(x)$ computed in (1);

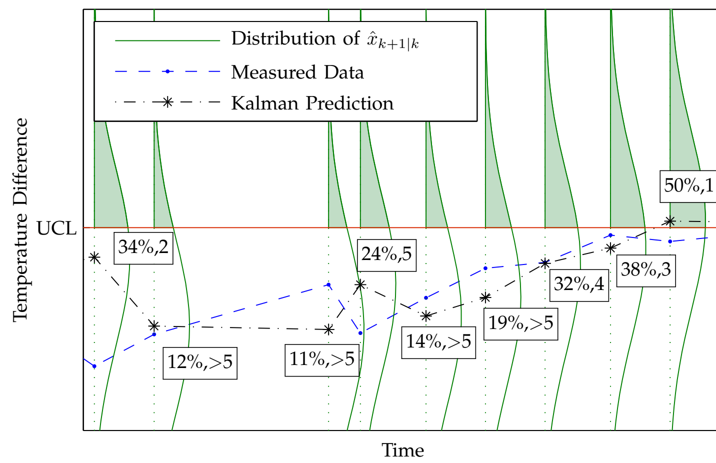


Figure 9.7: Screenshot from the data visualization tool: Each prediction is associated with the probability of maintenance need at next step (computed as integral of the shaded area) and with the estimated number of steps for being in a out-of-control state

Particle Filter.

In Fig. 9.8 and 9.7 the evolution of z_k is shown, together with its prediction $\widehat{z}_{k|k-1}$ and associated probability distribution. The probability of being in an out-of-control state, and therefore the probability of needing a control action, at next step is also obtained by integrating the part of the distribution that is outside the control limits (shaded area in Fig. 9.8). It is important to underline that, in a PdM perspective, this is actually the most useful outcome of the module.

The Kalman approach also allows to obtain predictions of z_k over several steps ($\hat{z}_{k+J|k}$, with $J > 0$). Exploiting (9.4) and (4.7) we get that

$$z_{k+J} = A^{J-1}\hat{z}_{k+1|k} + \sum_{j=1}^{J-1} A^{j-1}Bu_{k+J-j}, \tag{9.6}$$

with $A^0 = I$, the identity matrix.

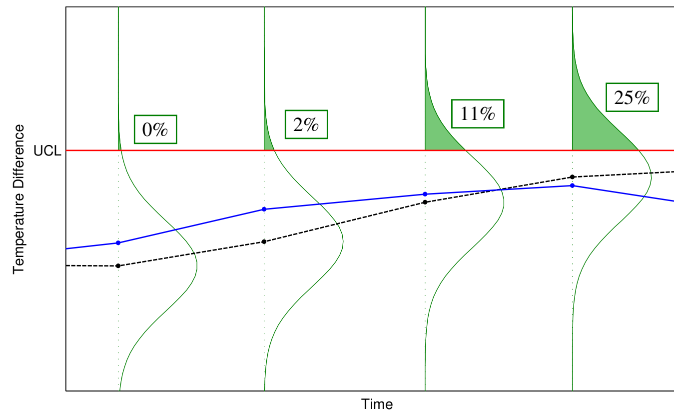


Figure 9.8: Evolution of the measured wafer temperature (solid, blue), 1-step Kalman prediction (dash, black), and associated probability distribution of the estimates (Gaussian curves). Percentage indices represent the control action need probabilities at every step

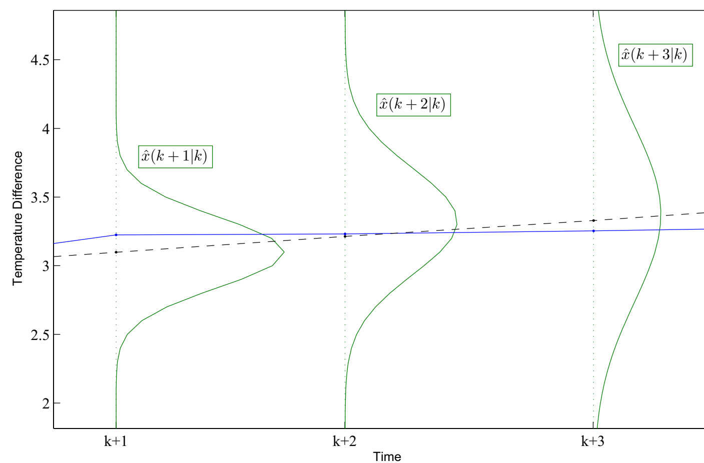


Figure 9.9: Evolution of the measured wafer temperature (solid, blue) along with Kalman predictor (dash, black) for multiple steps

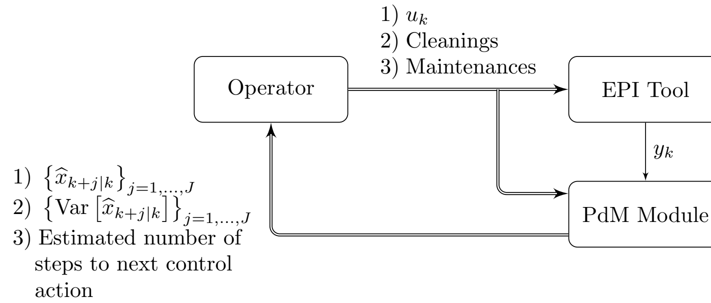


Figure 9.10: The PdM module action for each chamber at discrete time k in the control loop

In Fig. 9.9 a multiple-step prediction is shown. The prediction variance is increasing due to the fact that more variability is included in the model by increasing the number of steps.

As depicted in Fig. 9.10, the PdM module provides for each chamber:

1. an estimation of next values of temperature difference with confidence levels;
2. a health factor (HF) index that describes the probability of the machine to need a maintenance intervention at the moment;
3. an estimation of runs to be processed before a maintenance action is needed.

The same scheme is valid also for the Particle Filter approach, described in Section 4.3. In order to ease the priority of the interventions, the PdM module also provides a list of all the chambers ordered by their HFs.

9.4 Experimental Results

The proposed PdM approach has been tested on both simulation and fab data. To assess the quality of a Maintenance Management System we consider the following errors (Susto et al., 2012d):

- *Unnecessary Maintenance (UM), Type I Error* - an out-of-control state is predicted, and a maintenance is accordingly performed, but the evolution of x would not have exceeded the control limits even if the maintenance action was not performed (maintenance action predicted while it was not needed)[2]; the costs associated with an unnecessary maintenance are indicated with C_{UM} and they take into account

[2]Given the 'instability' of the dynamic system in exam, maintenances are not unnecessary, but in this case they are anticipated in time. To provide a fair evaluation of the PdM performances, we consider unnecessary maintenances those performed more than three steps before the actual out-of-control event.

the costs associated with the time spent by the process engineers to perform the maintenance and the time for which the machine is not operating. The amount of UM can be evaluated only if R2F data are available (otherwise it is not possible to evaluate if a maintenance was necessary or not).

- *Unprevented Out-of-Control state (UOC), Type II Error* - an out-of-control state that is not predicted in advance by the PdM system (maintenance action not predicted while it is needed). The costs associated with an UOC state are indicated with C_{UOC} and they take into account the costs associated with the decrease of production quality given by the out-of-control.

With a R2F policy the number of UMs (n_{UM}) is clearly always equal to zero (no maintenance action is taken in advance), while the number of UOC states (n_{UOC}) is equal to the number of process runs from one maintenance to the following. Usually $C_{\text{UM}} < C_{\text{UOC}}$.

We also have to define a criterion for triggering the maintenance action. The goal of a Maintenance Management system is that of minimizing the overall cost:

$$C = n_{UM} \times C_{UM} + n_{UOC} \times C_{UOC}, \tag{9.7}$$

therefore the conditions that trigger the maintenance action must be based on the values of C_{UM} and C_{UOC}. As stated before, the Filtering and Prediction algorithms proposed in this Chapter provides a probability indicator p_{MNT} of out-of-control state in the next k steps. We decide to perform a maintenance every time $p_{\text{MNT}} > k_{\text{T}}$, where k_{T} is a threshold that must be tuned accordingly to (9.7); qualitatively

- a conservative approach with k_{T} 'high' will perform few maintenances (if we want to decrease the number of n_{UM});
- a reactive approach where k_{T} is 'low' will perform more maintenances and avoid a high number of UOC states.

Simulation Data

The simulation data consist of 1000 process evolutions with a R2F maintenance policy, from a safe operational state to an out-of-control state. The data are generated by the same dynamical system described in (9.4) and (9.5). We first consider the case where the $v_k \sim G(0, 1; k)$ as in Fig. 9.11, $R = 1.1$, UCL $= 5$ and LCL $= -5$.

We first consider the 1-step ahead predictors, that are more accurate, given the fact that less process steps are taken into account and therefore, less variability. We compare

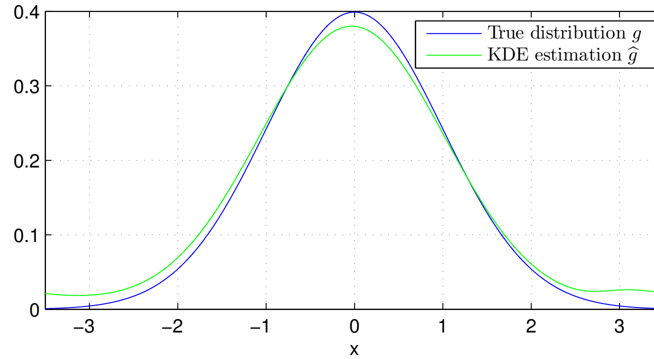


Figure 9.11: Gaussian $g(x)$ and PDF estimation $\hat{g}(x, \gamma^*)$ obtained through the Gaussian KDE

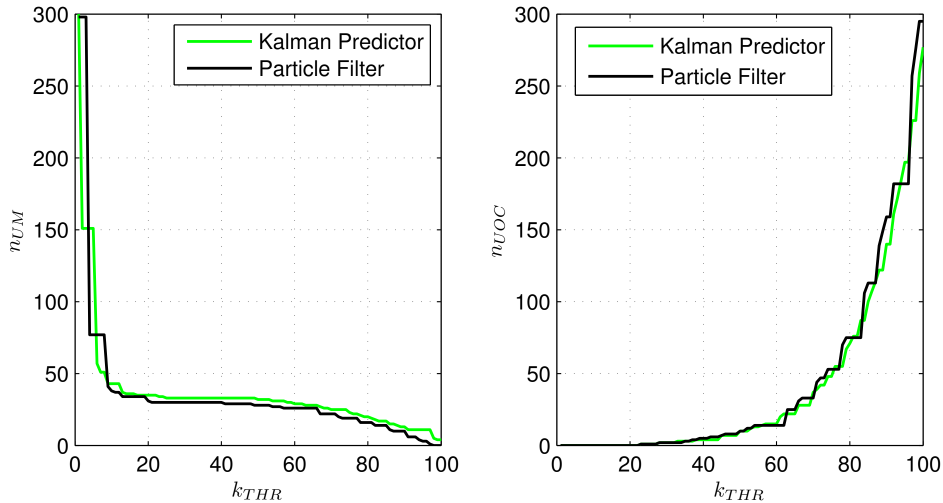


Figure 9.12: n_{UM} and n_{UOC} at the variation of the threshold k_{T} for the data with Gaussian distributed model noise

the performance of both Kalman and Particle Filter predictors. The dataset is split into a training dataset of 700 observations (for tuning the algorithm) and a validation dataset of 300 observations (for the performance evaluation). Several values of k_{T} are considered, and the corresponding results are reported in Fig. 9.12. It can be appreciated that the two approaches exhibit similar performance, so that the less computationally intensive Kalman Predictor can be preferred to the Particle Filter.

The scenario drastically changes if we consider the same system with a non-Gaussian distribution of the model noise. We considered $v \sim \frac{1}{2}G(0.1, 1; x) + \frac{1}{2}G(-2.3, 0.7; x)$, as in Fig. 9.13.

In Fig. 9.12 n_{UM} and n_{UOC} for different values of k_{T} are shown, for both approaches.

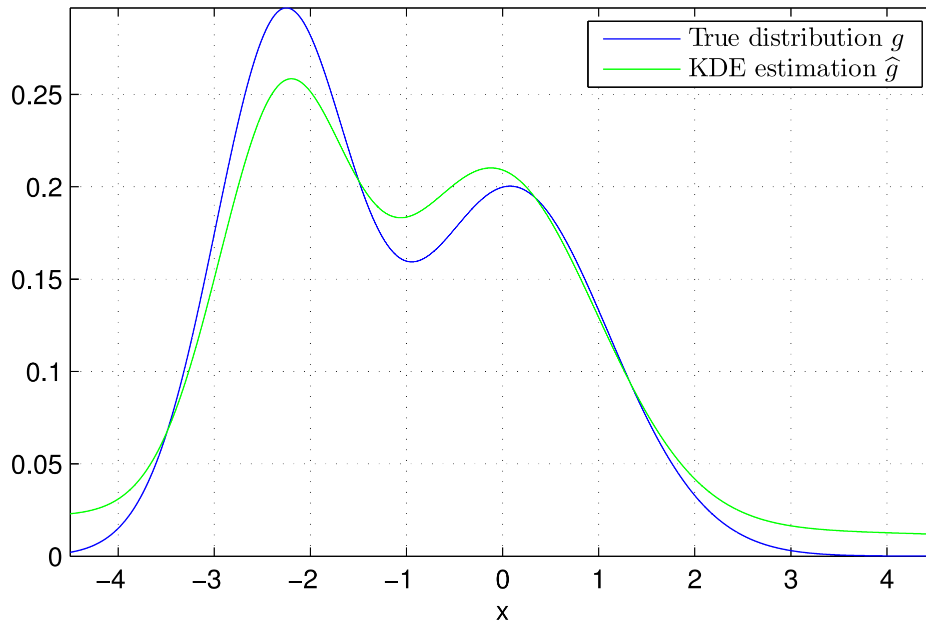


Figure 9.13: Non-Gaussian $g(x)$ and PDF estimation $\hat{g}(x, \gamma^*)$ obtained through the Gaussian KDE

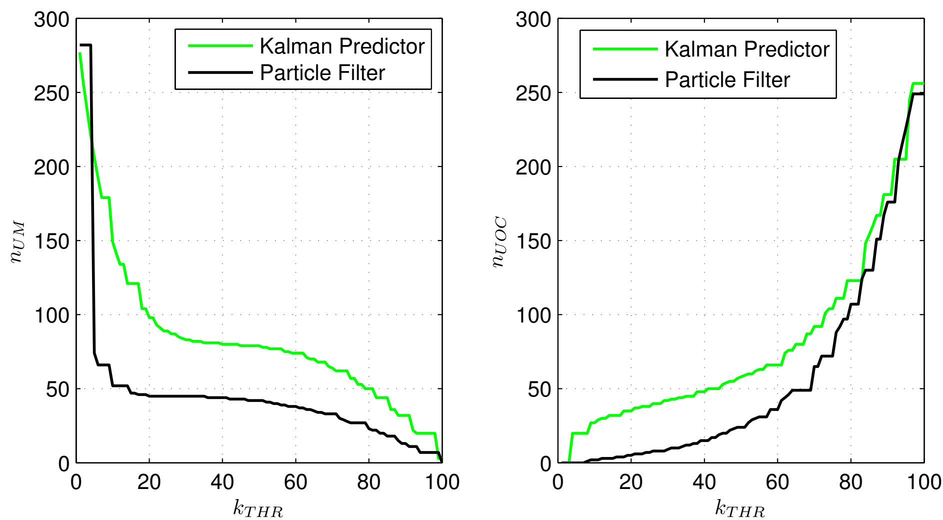


Figure 9.14: n_{UM} and n_{UOC} at the variation of the threshold k_{T} for the data with non-Gaussian distributed model noise

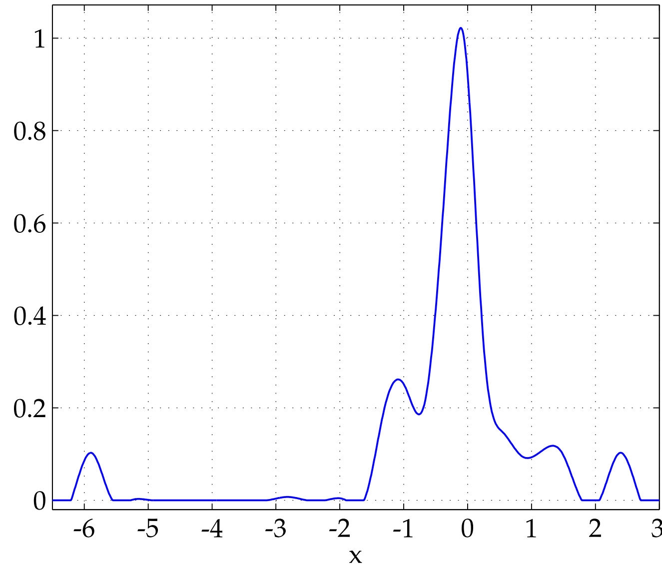


Figure 9.15: PDF estimation $\hat{g}(x, \gamma^*)$ obtained through the Gaussian KDE.

In this case the introduction of Particle Filter is justified by the fact that the Monte Carlo method outperforms the Kalman Predictor.

Fab Data

The proposed PdM module has also been tested on real fab data. The available data have been collected during approximatively 13 months of production where 2136 batches have been processed with 84 emissivity adjustments (maintenance interventions); we have split the data in order to have 70% of the maintenance interventions (59 emissivity adjustments) in the training dataset and 30% in the validation dataset (25 emissivity adjustments). For the equipment under consideration, the probability density function $g(x)$ of model noise has been estimated by using Gaussian KDE (more precisely with Algorithm 6 in Appendix), and it is shown in Fig. 9.15. The presence of smaller peaks for $x = 2.4$ and $x = -5.9$ proves that the Gaussian KDE correctly detects the presence of events associated with change points. Even at a visual inspection, the distribution cannot be considered as Gaussian, and accordingly, badly performs on the normality test. However, the estimated g in Fig. 9 is, like a Gaussian, concentrated around zero and has a fair distribution of outcome probability on both negative and positive part. Given this outcome and for illustrative purposes, we analyze the dataset by using both the Kalman Predictor and the Particle Filter, and compare the corresponding results.

We first illustrate an experiment to assess prediction accuracy of the two approaches.

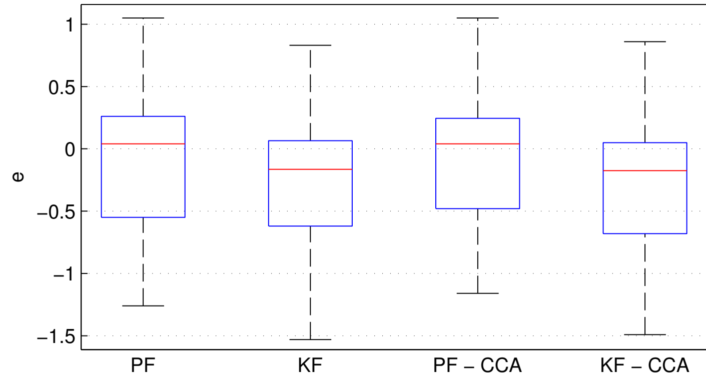


Figure 9.16: Boxplot: graphical representation of the Prediction error e distribution for the 1-step Kalman Filter (KF) and Particle Filter (PF).

In Fig. 9.16 the distributions of the 1-step prediction accuracy is reported in a boxplot

$$e_k = z_k - z_{k|k-1}, \tag{9.8}$$

for both Kalman Predictor and Particle Filter. The two filtering and prediction algorithms have been tested on two different dynamical models, the one described in equations (9.4)-(9.5) (KF and PF), where system matrices have been chosen after critical inspection of the data, and the one estimated by CCA (KF-CCA and PF-CCA), as described in Section (9.3).

We first observe that prediction error distribution with the CCA estimated model is slightly better (smaller variance) than the one obtained with system (9.4)-(9.5). However, the difference is quite small in the particular situation at hand, and given its easier applicability, we consider in the following system (9.4)-(9.5) only. However, we stress in other situations (e.g.,different tools with different dynamics), models derived by using data driven techniques such as CCA may be preferable. A second consideration is that Particle Filter prediction error distribution is preferable than the one of Kalman Predictor (less biased and with smaller variance). However, this better prediction accuracy doesn't translate in major advantages from a PdM point of view, as it will be shown with next experiments.

As stated in the previous Subsection, it is preferable to evaluate a maintenance management system in terms of n_{UM} and n_{UOC} than using the prediction error. We can also introduce a further element. Up to now, we have only considered the 1-step ahead predictor, however, this approach lacks in planning capability. For instance, if the process engineers are not available to perform the emissivity change before next batch, information given by the 1-step prediction may not be useful. Not only that, but it is

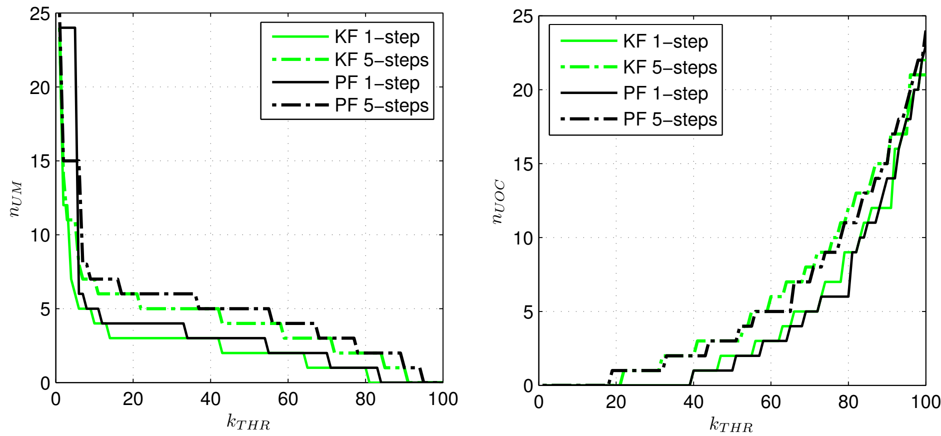


Figure 9.17: n_{UM} and n_{UOC} at the variation of the threshold k_{T} for the Fab data

usually convenient to perform more than one maintenance operation at the same time, to reduce tool down time. However, if we base the emissivity adjustments on the outcome of the 1-step ahead predictor only, there may be no opportunity of performing joint maintenance actions.

In Fig. 9.17 the performance of the Kalman Filter (KF) and Particle Filter (PF) are shown, in terms of n_{UM} and n_{UOC} for both 1-step and 5-steps predictions and various values of trigger threshold k_{T}. As expected, when we consider multiple step prediction, less accurate performances are available (more variability is taken into account in the predictions) w.r.t. 1-step ahead prediction, with the advantage of allowing more time and flexibility for process engineers to perform the maintenance. For example, with a 50% threshold maintenance policy based on the1-step PF, n_{UOC} decreases from 25 (R2F approach) to 1 at the cost of 3 unnecessary maintenances. No significant performance difference between KF and PF can be observed.

To evaluate the actual improvement (in economical terms) due to the introduction of the PdM system, we have to associate values to C_{UM} and C_{UOC}. We choose a reasonable set of possible values for C_{UM} and C_{UOC} (remembering that usually $C_{UM} < C_{UOC}$) and we then compare the performance of the PdM module to those of the classical R2F approach and of a PvM policy. The PvM policy introduced here is based on the computation of the mean process evolution from maintenance to maintenance:

$$\mu = \{\mathrm{mean}(m_i) \mid i \in \text{training dataset}\}\,, \tag{9.9}$$

where m_i is the number of batches processed from the previous emissivity adjustment to next one, for all emissivity adjustments i in the training dataset. The maintenance

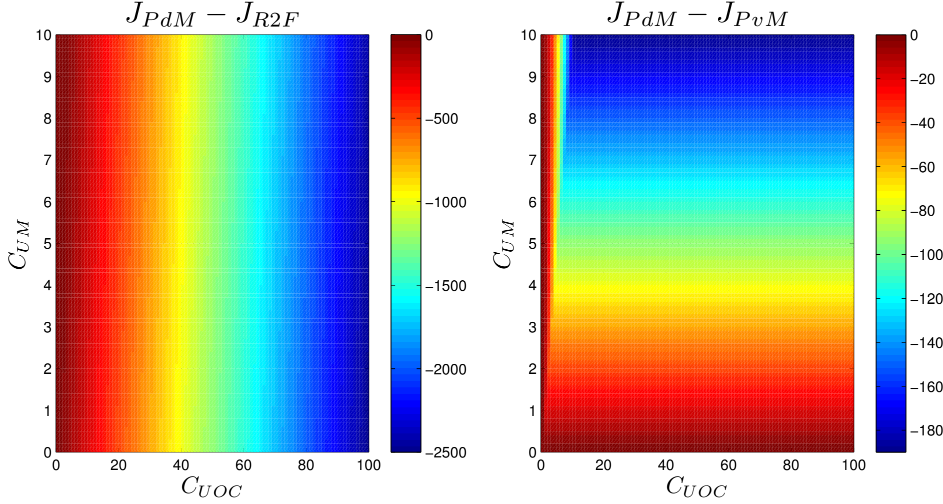


Figure 9.18: Performance comparison between PdM versus R2F and PdM versus PvM

action is then triggered after $\mu - k_{\mathrm{THR2}}$ batches have been processed from last emissivity adjustment, where the threshold k_{THR2} acts as k_{T} in the PdM module. Results of the comparison PdM versus R2F and PdM versus PvM are shown in Fig. 9.18. In both cases, the PdM module is based on the KF with 5-step prediction horizon. The overall cost $C_{PdM} - CR2F$ and $C_{PdM} - CPvM$ is shown for several pairs of $\{C_{UM}, C_{UOC}\}$ values. For each $\{C_{UM}, C_{UOC}\}$ pair, thresholds k_{T} and k_{THR2} are chosen so that C_{PdM} and C_{PvM} are minimized. It can be noticed that both $(C_{PdM} - CR2F)$ and $(C_{PdM} - CPvM)$ are always negative, that is, the proposed PdM reduces maintenance costs with respect to R2F and PvM. It can also be observed that R2F performances decrease badly if there are costs associated with Out-of-Control states (always present in real production environments) and that PvM performances are strongly affected by the cost of Unnecessary Maintenances.

9.5 Optimal Tuning of Epitaxy Pyrometers

As discussed, the procedure of recalibration of the pyrometers is done by changing their emissivity coefficient (Mizutani, 1988); after the deposition step of an EPI run if T is outside the control limits Upper Control Limit (UCL) and Lower Control Limit (LCL) the emissivity coefficient is changed in a typical *Run-to-Failure (R2F)* approach.

We have seen in the previous sections that with the use of filtering and prediction techniques (Ristic, Arulampalam, and Gordon, 2004) a system that predicts when control limits will be violated allowing the control engineers to know in advance when emissivity

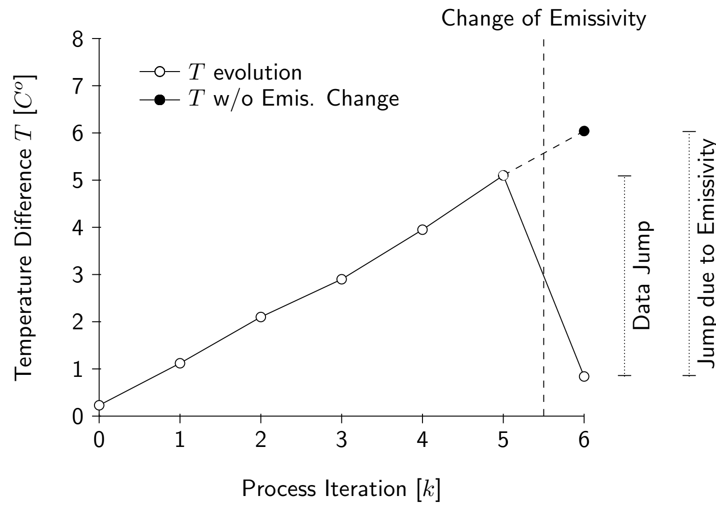


Figure 9.19: The actual jump on the temperature reading due to an emissivity change cannot be directly seen; this because the jump in the data it is also due to the natural trend of temperature difference

changes have to be performed can be implemented.

The aforementioned system does not completely resolve the maintenance issue in exam. Unfortunately, at the present state, the cause-effect relationship between emissivity coefficient changes and temperature reading changes is unknown and calibration of the emissivity coefficient are made according to a rule-of-thumb based on process engineers experience. The goal of this work is to present a statistical approach to compute reliable maps between emissivity and temperature reading changes for a real industrial environment.

There are several statistical techniques for regression that can be suitable to estimate our desired maps; for this issue, we have employed the LASSO (Tibshirani, 1996) and Smoothing Splines (Hastie et al., 2009), that are both regularization methods that allow a good trade-off between bias and variance of the estimator. The one that we are facing is a challenging task due to several issues:

(a) The effect of an emissivity adjustments on temperature readings are not directly observable from the data as can be appreciated by Fig. 9.19; 'jumps' in the temperature readings are due not only to emissivity changes but also to the natural trend of the temperature reading.

(b) Not all EPI chambers have an history usage where all the possible emissivity changes have been already performed on the tool; this is even more critical in the case some new equipment has just been recently installed. This complexity leads to few data

available to construct a reliable statistical model.

To overcome issue (a) we can employ the filtering and prediction techniques described previously in this Chapter; thanks to these techniques we can have an a priori estimation of what would have been the temperature difference without the emissivity correction and use this value to isolate the contribution of the emissivity change in the temperature jumps from the natural temperature trend.

To cope with (b) we used *Multi-Task techniques* (Evgeniou, Micchelli, and Pontil, 2005); the basic idea of this approach is that, if we have several similar functions to be estimated (like our emissivity change-temperature jump maps for different chambers of the same tool), we can obtained better estimation of our unknown functions by computing them jointly; this can be especially fruitful in the case of few data available, by exploiting the knowledge of the twin functions we can fill the gaps in the domain where few data are available.

The Regression Problem

We indicate with

- x the sensed temperature difference;
- u the change of emissivity in the lower pyrometer;
- s the temperature difference due to the emissivity adjustment u.

Let $\mathcal{I} = \{k \,|u_k \neq 0\}$; we consider the set of n couples

$$\{u_k, s_k\}\,, \quad \text{with } k \in \mathcal{I}. \tag{9.10}$$

From the observations (9.10) we want to infer the function

$$s = f(u) \tag{9.11}$$

in a typical regression framework.

At the moment, based on the fab tradition and process engineer expertise, (9.11) is considered as linear relationship between emissivity changes and temperature difference,

$$\hat{s} = \alpha u. \tag{9.12}$$

However, the current control policy based on (9.12) is not satisfactory and usually several emissivity adjustments are necessary to effectively correct the temperature readings. We propose here a model to estimate statistically this relationship from the historical data.

The three approaches used for this regression problem are:

1. Ordinary Least Squares (Section 2.1)
2. LASSO (Section 2.1)
3. Smoothing Splines

The last method employed for one-dimensional regression is *Smoothing Spline* (Hastie et al., 2009), a regularization approach that do not need to expand the basis of the input. Generally, with SS the estimated approximation $\hat{f}(\cdot)$ of $f(\cdot)$ is the one the minimize

$$\mathcal{F} = \frac{1}{n}\sum_{k=1}^{n}\Big(s_k - \hat{f}(x_k)\Big)^2 + \lambda\int_{\mathbb{R}} \hat{f}''(x)^2 dx. \tag{9.13}$$

SS are widely used thanks to the fact that are a flexible tool to obtain interpolation solutions with a desired degree of smoothness. For more detailed description of SS-based methods we refer the reader to Hastie et al. (2009).

Multi-Tasking

The framework described in the previous section is complicated by the complexity of a normal fab environment where several multi-chamber tool are present, each one with different products and *recipes* (tool settings). We indicate with N_T the number of tools in the fab. The generic tool j has N_C^j chambers and N_P^j products that can be run on that machine. Each chamber has its own behaviour and each product is processed at a given temperature, therefore the relationship $f(\cdot)$ should be modeled differently for every *logistic setting* (a particular configuration of tool, chamber and recipe). Therefore, a total of

$$N_F = \sum_{i=1}^{N_T} N_C^i N_P^i \tag{9.14}$$

different functions $\{f^i(\cdot)\}_i^{N_F}$ must be learned. This high-mixed scenario usually leads to few data available for some logistic settings and consequent poor prediction accuracy.

The previous issue can be overcome with the use of *Multi-Tasking* methods (Evgeniou et al., 2005). The basic idea of these approaches is that, instead of modeling each function separately, we learn all of them together with the assumptions that the similarities between the different functions can help to improve the overall modeling performances.

For the sake of the explanation we consider here that the functions f^i are linear

$$f^i(u) = \Phi(u)\beta^i \quad \forall i = 1, \ldots, N_F. \tag{9.15}$$

We will choose the coefficients $\{\beta^i\}_{i=1}^{N_F}$ as the minimizers of the regularization function

$$\mathcal{F}_{\mathrm{MT}} = \frac{1}{nN_F}\sum_{i=1}^{N_F}\sum_{k=1}^{n}\left(s_k^i - \Phi(x_k^i)^T\beta^i\right)^2 + \lambda\mathcal{J}(\beta), \tag{9.16}$$

where $\lambda \geq 0$ is the regularization parameter and $\mathcal{J}$ is a function that describes the relationship between the different tasks.

For particular choices of $\mathcal{J}$, (9.16) still learn the tasks independently; if $\mathcal{J}(\beta) = \sum_{i=1}^{N_F}|\beta^i|$ then (9.16) became

$$\mathcal{F}_{\mathrm{MT}} = \sum_{i=1}^{N_F}\mathcal{F}^i \tag{9.17}$$

with $\mathcal{F}^i$ the LASSO objective function, i.e. a sum of N_F separated LASSO problems. However, if we choose for example $\mathcal{J}(\beta) = \sum_{i,j=1,i\neq j}^{N_F}\left\|\beta^i - \beta^j\right\|^2$ we can favorite solutions where the tasks are close to each other. The previous approach allows to improve the prediction performances of those tasks for which few data are available, by taking advantage of the similarities with the tasks that have more observations.

In this work we will choose $\mathcal{J}(\beta)$ in order to keep close the functions that share the same tool, chamber or recipes, with the assumptions that those tasks have some similarities and will enjoy the benefits of this approach.

The relationship between LASSO and multi-task methods are further detailed in Pampuri et al. (2011b).

The problem we are facing, as described in Section 9.5 and 9.5, is furtherly complicated by the fact that, unfortunately, s quantities are not available[3]. Instead we will have to seek for an estimation of s. In order to do that, we will employ filtering and prediction techniques to decouple the jumps in temperature readings due to emissivity adjustment and the normal trend of temperature reading evolution.

As introduced previously in this section, for our analysis we need to obtain an a priori estimation of what would have been the temperature difference without emissivity correction; formally we want to estimate the conditional probability density

$$p(z_{k+1}|z_k) \tag{9.18}$$

given the measurements $Y_{1:k} = \{Y_1, Y_2, \ldots, Y_k\}$. As explained earlier in this Chapter, based on whether the distribution $g(x)$ in Assumption 9.3.2 is Gaussian or not, different approaches may be used to solve the filtering and prediction problem for the linear system (9.4)-(9.5). A general approach, suitable for every scenario, are Sequential Monte Carlo

[3]In this section we will omitted the apices indicating the logistic settings to lighten the notations.

Methods (SMCM), or Particle Filter.

To overcome the lack of observations of s, we can now employ the following quantities in our regression problem

- $t_k = (Y_k - Y_{k-1})$, the temperature jump observed after the emissivity adjustment at time k;
- $\tilde{t}_k = \left(Y_k - \widehat{z}_{k|k-1}(1)\right)$, the temperature difference with the trend compensated by considering the a-priori estimation $\widehat{z}_{k|k-1}(1)$ obtained with the Particle Filter.

Experimental Results

We consider the production data of two different EPI tool, $E1$ and $E2$, each one with 3 separated chambers (A, B and C). The same recipe $R1$ has been used in $E1$ and $E2$, while a second recipe $R2$ has been employed in $E2$. We therefore have:

$$\begin{aligned} &N_T = 2, \\ &N_C^{E1} = N_C^{E2} = 3 \\ &N_P^{E1} = 1, \quad N_P^{E2} = 2. \end{aligned}$$

According to (9.14), we have a total of $N_F = 9$ functions to be learned.

The data available regards 18 months of consecutive production; the number of observations available for each logistic settings are summarized in Table 9.1.

We compare the performances of the regression methods proposed in Section 9.5 with the actual policy (9.12) applied by process engineers. In order to compare those methods we compute the prediction error e that is computed as

$$e = t_k - \hat{f}(u_k). \tag{9.19}$$

The evaluation of methods' performances is done through Monte Carlo crossvalidation (MCCV), where M simulations are done by randomly splitting the n observations into a training dataset of $n_{\text{TR}} = qn$ observations and a validation dataset of $n_{\text{VAL}} = (1-q)n$

Tool-Chamber	E1-A	E1-B	E1-C	E2-A	E2-B	E2-C
Recipe 1	34	56	45	69	45	46
Recipe 2	-	-	-	53	32	34

Table 9.1: Number of observations for each Logistic settings

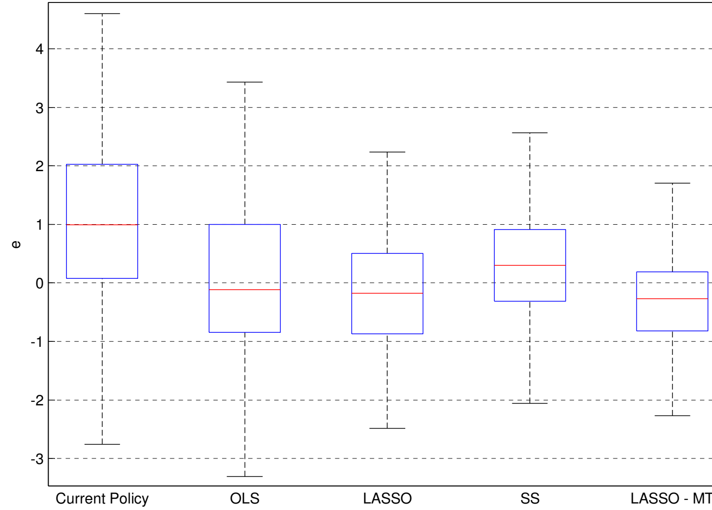


Figure 9.20: Boxplots of the prediction error e obtained with the actual policy (9.12), OLS, LASSO, Smoothing Spline (SS) and LASSO with Multi-Task (LASSO-MT)

observations, with $0 < q < 1$. In our experiments we have chosen $q = 0.7$ and we have collected data for $M = 3000$ simulations. Boxplots in Figure 9.20 represent the prediction error e distribution of the different approaches for the calibration of pyrometers[4] described in Section 9.5 and 9.14. The prediction errors are related to all the N_F possible logistic setting.

It can be clearly appreciated how the current policy for emissivity adjustments can be easily outperformed by any statistical approach; in fact, the current approach provides positively biased adjustments. On the other hand the approach that guarantees the best performance in terms of error variance is the LASSO with Multi-Tasking penalization (LASSO-MT). In these simulations we have chosen $\mathcal{J}(\beta) = \sum_{i,j=1,i\neq j}^{N_F} \delta_{i,j} \left\|\beta^i - \beta^j\right\|^2$ as the task regularization function, where

$$\delta_{i,j} = \begin{cases} \delta_T & \text{if } i \text{ and } j \text{ share the same tool} \\ \delta_C & \text{if } i \text{ and } j \text{ share the same chamber} \\ \delta_R & \text{if } i \text{ and } j \text{ regards the same recipe} \end{cases}$$

The three penalization parameters δ_T, δ_C and δ_R, alongside with the regularization parameter λ, are chosen, also in this case, through cross-validation. In the simulation performed, the best results were achieved when $\delta_C >> \delta_T$ and $\delta_C >> \delta_R$, underlying that each chamber can be modeled as a singular machine with its own peculiarities (Susto et al., 2011b) and that the recipe run on the machine do not change particularly the

[4]Outliers have not been depicted in Fig. 9.20 in order to improve readability.

Logistic Setting	Data jump t_k	Jump detrended $\tilde{t}_k$
E1-A, Recipe 1	1.1800	**0.9861**
E1-B, Recipe 1	0.9899	**0.7587**
E1-C, Recipe 1	1.0307	**0.8994**
E2-A, Recipe 1	0.9117	**0.7326**
E2-B, Recipe 1	**1.0636**	1.0714
E2-C, Recipe 1	1.0529	**0.9696**
E2-A, Recipe 2	0.9910	**0.8699**
E2-B, Recipe 2	**1.2236**	1.3544
E2-C, Recipe 2	1.2081	**1.1705**
Multi-Task	0.6446	**0.5505**

Table 9.2: RMSE(t) and RMSE($\tilde{t}$) obtained with LASSO

effect on the temperature reading of the emissivity adjustments.

We then compare the performances of the LASSO and LASSO with Multi-Tasking by employing the 'detrended' data $\tilde{t}$ instead of t. The accuracy of the LASSO is compared in terms of Root Mean Squared Error (RMSE)

$$\text{RMSE}(t) = \frac{1}{n}\sum_{k=1}^{n}\left(t_k - \hat{f}(x_k)\right)^2, \tag{9.20}$$

that is slightly modified for the Multi-Task case in

$$\text{RMSE}(t) = \frac{1}{nN}\sum_{j=1}^{N}\sum_{k=1}^{n}\left(t_k^j - \hat{f}^j(x_k^j)\right)^2. \tag{9.21}$$

In Table 9.2 are reported the performances of LASSO in terms of (9.20) and (9.21). The RMSE statistics reported are averages made on $K = 500$ MCCV simulations. In bold are indicated the dataset that achieves better prediction for each logistic setting. It can be appreciated that generally with the new dataset $\{x, \tilde{t}\}_i^n$ better accuracy is obtained in the prediction with respect to what is achieved with $\{x, t\}_i^n$. The new dataset is therefore more predictable than the previous, suggesting that inner temperature trend has been properly decoupled from the effect of the emissivity adjustment.

9.6 Fab Implementation and Conclusions

In this Chapter, a Predictive Maintenance System for an Epitaxy process has been introduced. The tool provides the process engineer with an estimate of when the next

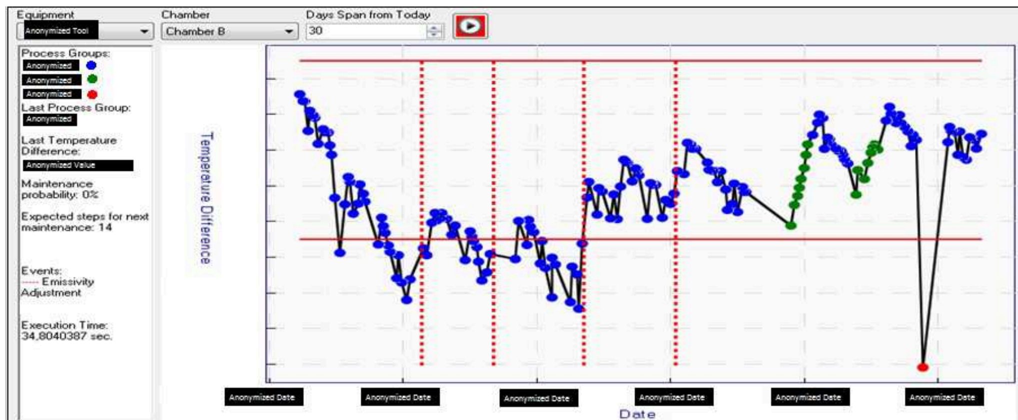


Figure 9.21: Screenshot of the PdM Graphic User Interface

control action has to be performed on the tool, with an associated confidence level. The PdM module is based on an algorithm for predicting the wafer temperature behaviour. To this aim, two different approaches have been compared, namely the Kalman Predictor and the Particle Filter with Gaussian Kernel Density estimator. In the particular application case considered here, both prediction approaches exhibit similar performance, as the probability distribution function of the model noise has, like a Gaussian, most of its probability around zero. In general, however, if noise distributions are not Gaussian, as may be the case for other tools and recipes, the Particle Filter may prove to be a more flexible and general tool for the filtering problem than the Kalman Predictor. Application of such tools to other fab case studies is under investigation.

Although the results presented in the Chapter have been obtained with one recipe only running on the tool, the module is completely scalable. Since data can be collected for each recipe run on the machine, the corresponding error distribution can be estimated, and prediction performed for the specific recipe running on the tool.

As suggested by Process Engineers, different recipes have different impacts on the tool and therefore, for enhanced prediction accuracy, we suppose it is better to have different distribution estimations for each recipes. The available dataset describes a chamber of the tool where just one recipe was run, so we don't have experimental results to support this claim, that is however reasonable from the physics of the tool point of view.

The proposed PdM system has been implemented in a C# program to be used by process engineers. In Fig. 9.21 a screenshot of the first MATLAB implementation of the Graphical Used Interface is reported. In such C# implementation, the user can have a general overview of the state of several equipments with the indication on the maintenance probability at next step, the estimated amount of lots to be processed before

an emissivity adjustment has to be performed and the suggested amount of emissivity correction.

A system to estimate the relationship between emissivity adjustment of pyrometers and temperature readings difference in an Epitaxy tool have been also proposed.

We have shown that with the statistical approach presented the tuning policy of the pyrometer can be greatly improved. Thanks to the increased accuracy in the emissivity adjustments, we estimate that with a Multi-Task LASSO-based estimation of the map (9.11) the number of emissivity adjustments to be performed can decrease of the 34%.

The use of Multi-tasking techniques guarantees good prediction even in the case of few observations available. The lack of homogeneous data is typical of semiconductor manufacturing modeling where high-mixed production realities are present and therefore Multi-Tasking schemes can enhance the prediction performances of several Advanced Process Control modeling modules, from Virtual Metrology (Schirru et al., 2011) to Predictive Maintenance.

10

Predictive Maintenance for Ion-Implantation

Ion Implantation is one of the most sensitive processes in Semiconductor Manufacturing. It consists in impacting accelerated ions with a material substrate and is performed by an Implanter tool. The major maintenance issue of such tool concerns the breaking of the tungsten filament contained within the ion source of the tool. This kind of fault can happen on a weekly basis, and the associated maintenance operations can last up to 3 hours. It is important to optimize the maintenance activities by synchronizing the Filament change operations with other minor maintenance interventions.

In this Chapter, a PdM system is proposed to tackle such issue; the filament lifetime is estimated on a statistical basis exploiting the knowledge of physical variables acting on the process. Given the high-dimensionality of the data, the statistical modeling has been based on the Regularization Methods described in Section 2.1: Lasso, Ridge Regression and Elastic Nets.

The predictive performances of the aforementioned regularization methods and of the

proposed PdM module have been tested on actual productive semiconductor data.

The work described in this Chapter has been partly presented in Susto et al. (2012d) and it has been developed in collaboration with STMicroelectronics in Catania.

10.1 Introduction

In this Chapter, we propose a PdM system for one of the most important processes in semiconductor manufacturing, Ion Implantation (McKenna, 2000). Ion Implantation consists in the imprinting of accelerated ions in the processed target wafer, thus altering the target elemental composition. In this way, it is possible to improve the conductivity of the semiconductor device. The major maintenance issue of the Implanter tool concerns the breaking of the tungsten filament in the Ion Source from which the electrons are emitted. Figure 10.1 depicts a new filament as well as a broken one. Several factors may negatively impact the operational lifetime of a filament, such as high pressure, voltage and filament current. Routine maintenance operations such as cleaning, installation and degasification can also fundamentally impact filament health. The filament breaking fault can happen on the tool on a weekly basis. Every time a filament is changed, it can take up to 3 hours to bring back the tool to a running state. As a consequence, an optimization of the maintenance activities that synchronizes filament changes with other minor maintenance operations is extremely desirable.

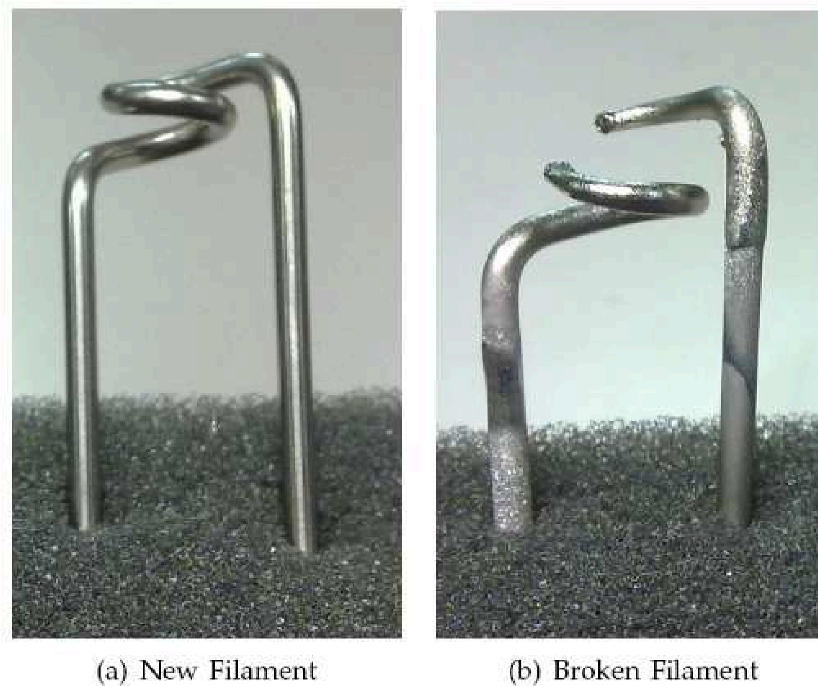
Policies enforced by most manufacturers rely on a fixed approach: the filament is changed every time the Implanter tool reaches a predefined amount of working hours. This is a typical *Preventive Maintenance (PvM)* approach and can suffer from two main drawbacks:

(i) filament is usually changed when it would still be usable;

(ii) filament faults can still take place.

By analyzing historical data, it is possible to optimize the PvM approach and obtain a good trade-off between undesired non-prevented faults and filament 'life' exploitation. Unfortunately, preventive maintenance methodologies do not take advantage of the current state of the machine: decisions are taken on the basis of past fault statistics like mean and median of filament lifetime.

In the hereby proposed Predictive Maintenance (PdM) approach, the lifetime of the filament is statistically estimated by relying on historical and current values of the physical variables acting on the process (such as pressures, voltages, filament currents, and so on). The provided lifetime prediction allows for maintenance and production



(a) New Filament (b) Broken Filament

Figure 10.1: Images of a tungsten filament before the installation on the Implanter tool and after a breaking.

activities to be appropriately performed, granting an increased equipment uptime and stability and a greater exploitation of the filament that will be changed just when needed. Several statistical approaches have been compared to have an estimation of the Remaining Useful Life (RUL) of the filament.

One of the major challenges in semiconductor manufacturing process modeling is that the number of tool variables is usually very high; in such setting, variable selection techniques often prove to be useful (Schirru et al., 2011). In the approach here proposed, the predictive models are obtained in a regularized machine learning framework (Hastie et al., 2009) that includes methodologies such as the LASSO (Tibshirani, 1996) and the Elastic Net (Zou and Hastie, 2005), that are quite adept in dealing with dimensionally demanding input spaces. Kernel-based methodologies (Scholkopf and Smola, 2001) have also been employed in order to model non-linear relationships and improve prediction accuracy.

While FDC systems have already been proposed for Ion Implanter (Lin, Hung, Lin, and Cheng, 2006), a PdM system aiming to predict failures is yet to come. It is worth noticing that the main difference between FDC and PdM systems is that FDC systems try to detect failures that already happened or are happening on a tool, while PdM

systems predict (and allow to prevent) possible failures in the future. Differently from Lin et al. (2006), where the FDC system is designed to deal with all possible failures on a tool, in this Chapter the focus is set on a specific fault, filament ruptures.

The rest of the Chapter is organized as follow: in Section 10.2 a brief description of the tool and the data are provided. In Section 10.3 the experimental results are illustrated, while the concluding Section 10.4 provides some summary comments on the present work and further developments.

10.2 Tool Description and Problem Formalization

Through Ion Implantation, it is possible to modify the electrical properties of the wafers by injecting doping atoms. Such step is considered a 'bottleneck' in semiconductor fabrication because of the high cost of the Implanter tool. For this reason, ion implantation is a critical operation for throughput (Lin et al., 2006).

Figure 10.2 depicts a scheme of the Implanter tool. The tungsten filament is part of the Ion Source, which is in charge of producing the ions. During the process, the filament is heated and electrons are 'boiled' off the heated filament; the electrons are then accelerated in the beamline area and then impinge on the wafers in the End Station area.

A change in the status of the filament can be detected by looking at the evolution of filament current (Fig. 10.3). A large variation from a low value of current to an high value means that a filament has been replaced. However, this is valid only if we have a dataset of *Run-to-Failure (R2F)* data. R2F is the simplest approach of maintenance management and consists in acting on the tool just after failure. The availability of R2F data (where all the replacements occur because of a broken filament) allows to observe the complete lifetime of the component. Intuitively, a dataset built out of Preventive Maintenance data would mask that information.

We indicate with

$$y = \text{batches missing before next filament breaking;}$$

y therefore represents the RUL of the filament in terms of remaining batches that can be processed on the tool before next filament change. Our aim is to tackle the prediction of y as a regression problem, starting from the tool data $X \in n \times p$, where n are the number of observations (batched processed in our case) and p are the number of regressors (the tool variables). It is useful to define also

$$z = \text{process iteration from last filament change}$$

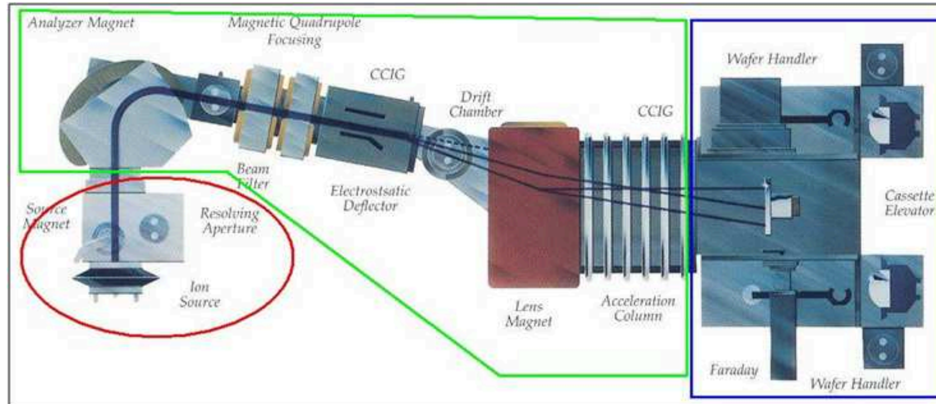


Figure 10.2: Scheme of the Ion Implanter tool. The tool can be divided in three parts: the Source ◯, the Beamline Area □, the End Station □.

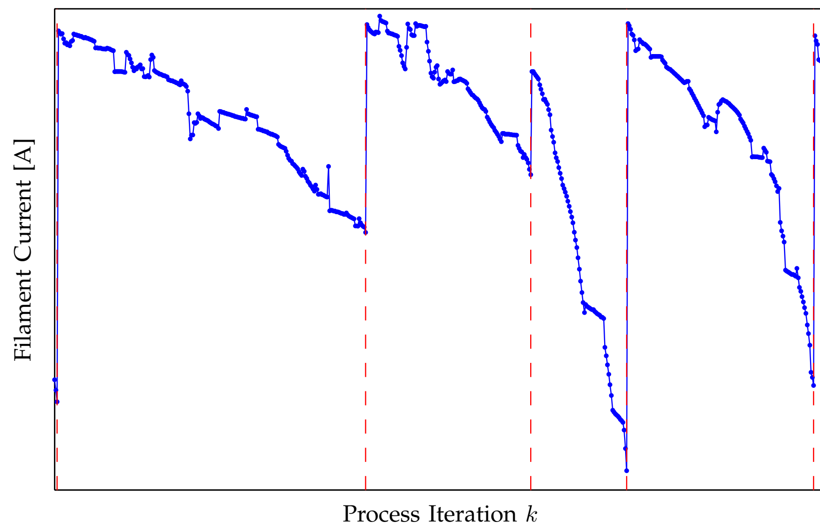


Figure 10.3: Evolution of the filament current over a month of productive data. Filament breakings (- - -) are in correspondence with jumps in the current.

and we indicate with z_{FB} the iteration of the process in correspondence with the filament breaks.

The available dataset consists of n observations (x_i, y_i) where $x_i \in \mathbb{R}^p$ are the tool variables value for process iteration i and $y_i \geq 0$ is the number of batches before next filament rupture. We suppose that there exists a relationship

$$y = f(x), \tag{10.1}$$

and we want to estimate $f(\cdot)$ form the set of observations $\{(x_i, y_i)\}_{i=1}^n$, in order to be able to predict the number of batches before the next fault; this is the framework of a typical regression problem.

10.3 Experimental Results

Prediction Accuracy

The available dataset consists of $N = 33$ *maintenance cycles* of a tool from maintenance to maintenance, with filament R2F policy (i.e. data of a tool from the installation of a new filament to filament breaking and tool stopped for maintenance), for a total of $n = 3671$ batches. The number of physical variables is $p = 125$.

We first build up experiments to asses which is the best prediction methods in terms of prediction accuracy, by computing the prediction error $e = y - s$. The evaluation of methods performances is done through Repeated Random Sub-Sampling Validation.

In Fig. 10.4 the prediction error $e = y - s$ distribution are represented in boxplots[1], with some of the regularization methods described in Chapter 2. It can be appreciate how Elastic Nets outperform all the other methods.

Comparison with R2F and PvM

Prediction accuracy is not an informative criterion to evaluate a maintenance management system. We evaluate the performances of the proposed maintenance policies in terms of two indicators:

(i) N_{UB} = average number of not prevented maintenances;

(ii) N_{BL} = average number of batches that may have been processed if the filament has not be preventively changed.

[1] Due to the enormous amount of data, outliers have been omitted in Fig. 10.4 in order to improve readability.

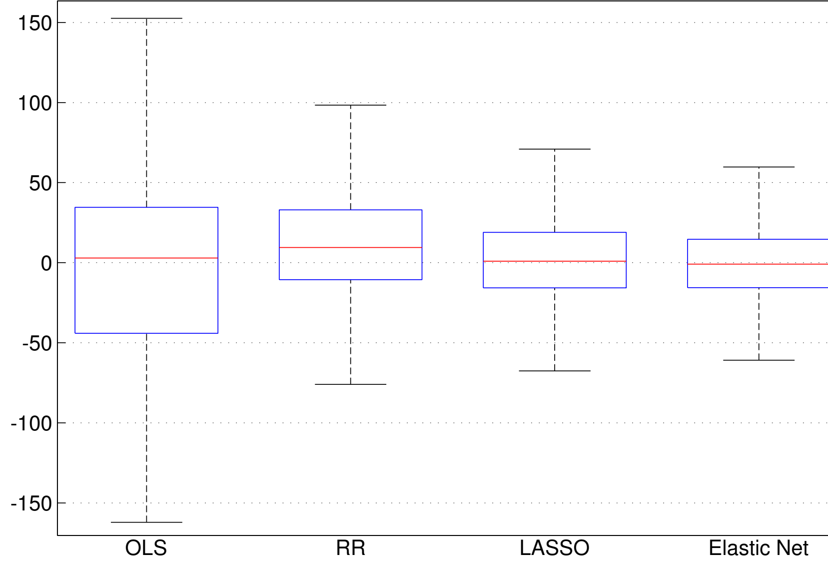


Figure 10.4: Boxplots of the prediction error $e = y - s$ obtained with OLS, Ridge Regression (RR), LASSO and Elastic Net.

The R2F approach clearly guarantees a total of $\mathsf{N}_{\text{UB}} = 1$ and $\mathsf{N}_{\text{BL}} = 0$ given the fact that a filament is never changed until it breaks.

We simulate here the performances of a Preventive Maintenance policy. As stated before, a PvM approach is usually based on the estimation of the mean or median batches processed with a newly installed filament until it breaks. We have done this by computing mean and median on the validation dataset for each experiment

$$\mu = \text{mean}\left\{z^i_{\text{FB}} | i \in \text{training dataset}\right\}, \quad (10.2)$$

$$\eta = \text{median}\left\{z^i_{\text{FB}} | i \in \text{training dataset}\right\}, \quad (10.3)$$

where z^i_{FB} is the iteration of the process in correspondence with the filament breaks for the i-th maintenance cycle.

Usually, once μ or $\eta(z_{\text{FB}})$ have been computed, a PvM approach is to change the filament once $\mu - k_{\text{THR}}$ (or $\eta - k_{\text{THR}}$) batches, where k_{THR} is a positive integer, have been processed until the filament has been installed. Depending on the choice of k_{THR}, the PvM system have different performances:

- values of k_{THR} 'large' allow to have N_{UB} low at the price of a considerable number of N_{BL};

- values of k_{THR} 'small' permits a small number N_{BL} at the price of high N_{UB}.

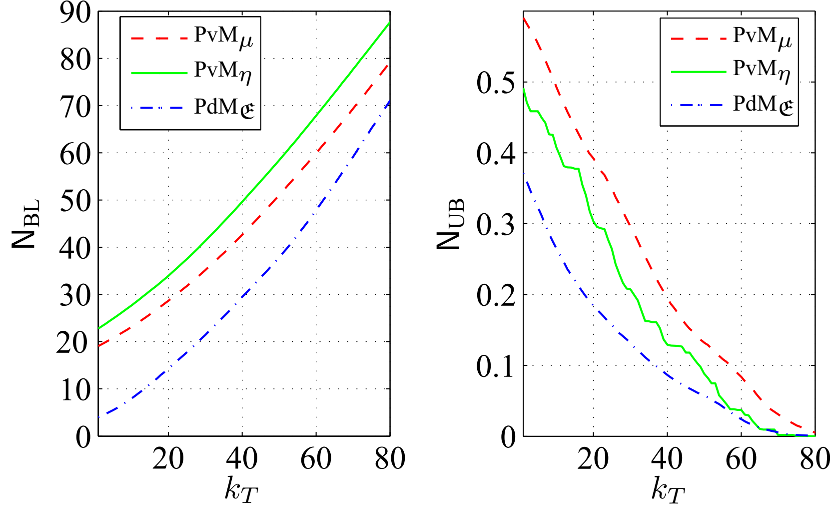


Figure 10.5: The performances of the PvM_{μ}, PdM_{η} and $\text{PdM}_{\mathfrak{E}}$ as a function of the threshold k_{THR}.

We indicate with PvM_{μ} the PvM policy based on the mean and with PvM_{η} the one based on the median.

The PdM approach, instead of being based on historical statistics, relies on the prediction $\hat{y}$ of the regression model (LASSO, $\mathfrak{L}$, or EN, $\mathfrak{E}$), and the filament is changed when $\hat{y} \leq k_{\text{T}}$ where

- values of k_{T} 'large' allow small N_{UB} and large N_{BL};
- values of k_{T} the contrary.

In this case we indicate the PdM policy with $\text{PdM}_{\mathcal{X}}$, where $\mathcal{X}$ is the method employed for the prediction.

In Figure 10.5 the performance in terms of N_{UB} and N_{BL} of PvM_{μ}, PdM_{η} and $\text{PdM}_{\mathfrak{E}}$ in terms of N_{UB} and N_{BL} as a function of of the threshold k_{THR} is shown. PdM based on Elastic Nets completely outperforms PvM approaches (besides for some really conservative choices of k_{THR} for which N_{UB} are almost the same for every approach).

Fab Environment Simulation

The results presented in the last section do not fully motivate the preference of a PdM w.r.t. a PvM approach. In fact, a PvM approach guarantees to know in advance the time when a maintenance action has to be performed on the machin. This is a great advantage because it is not always possible to act on the tool right after the maintenance

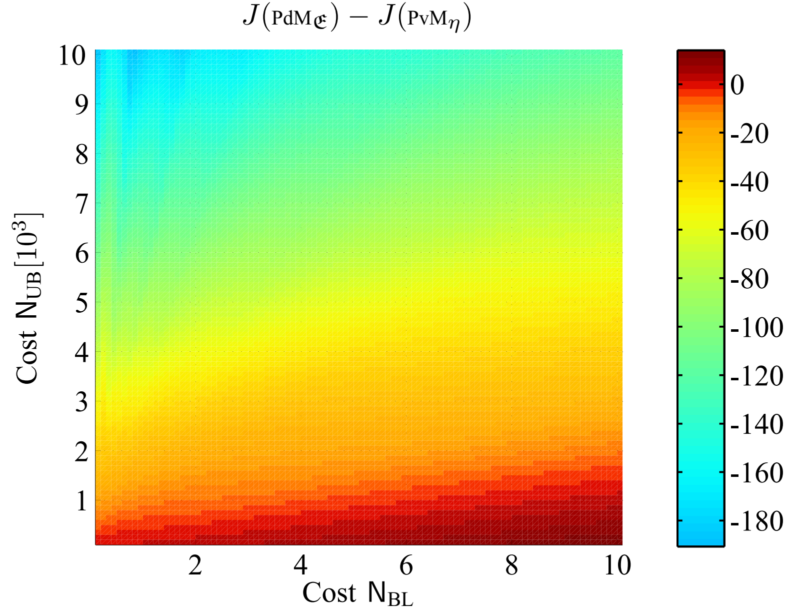


Figure 10.6: Matrix of $J_{\text{PdM}_{\mathfrak{C}}} - J_{\text{PvM}_{\eta}}$ at the change of (Cost N_{UB}) and (Cost N_{BL}). $J_{\text{PvM}_{\eta}}$ and $J_{\text{PdM}_{\mathfrak{C}}}$ are the minimum at the change of k_{THR}.

management system has suggested to do so (e.g., for issues associated with availability of maintenance personnel) .

To simulate this scenario, a delay δ between the suggestion by the PdM module to act on the tool and the effective performance on the machine is introduced. By doing so, the performance of the PdM approach when implemented in the fab environment is described more realistically and it is possible to better assess whether a PdM method can be preferable to the PvM one.

The delay δ has been modeled as a mixture of three Poisson distributions

$$\delta \sim \alpha_1 \text{Pois}(\gamma_1) + \alpha_2 \text{Pois}(\gamma_2) + \alpha_3 \text{Pois}(\gamma_3), \tag{10.4}$$

with $\alpha_i > 0$, $\sum_{i=1}^{3} \alpha_i = 1$ and $\{\gamma_i\}_{i=1}^{3} = \{2, 10, 35\}$ minutes to represents three kind of different delays during a working day (respectively delay due to a sort of 'reaction time' to the alarm, a break and a lunch). This delay is added to the answer time to a warning from the PdM tool to act on the machine. We compare PdM and PvM by computing the index

$$J = \mathsf{N}_{\text{UB}} \times (\text{Cost } \mathsf{N}_{\text{UB}}) + \mathsf{N}_{\text{BL}} \times (\text{Cost } \mathsf{N}_{\text{BL}}), \tag{10.5}$$

where (Cost N_{UB}) and (Cost N_{BL}) are the costs associated to unexpected breaks and batch for which the old filament has not been used. In Fig. 10.6 the difference $\Delta_J =$

$J_{\mathrm{PdM}_{\mathfrak{C}}} - J_{\mathrm{PvM}_{\eta}}$ is reported for various couples of values for (Cost N_{UB}) and (Cost N_{BL}) (the values have been chosen with the help of process experts). For each couple of $\{\text{Cost } \mathsf{N}_{\mathrm{UB}}, \text{Cost } \mathsf{N}_{\mathrm{BL}}\}$, $J_{\mathrm{PvM}_{\eta}}$ and $J_{\mathrm{PdM}_{\mathfrak{C}}}$ are choosen as the minimum of (10.5) at the variation of k_{THR}.

Negative values of Δ_J represent combination of costs for which the PdM approach outperforms the PvM and the contrary for positive values. We can appreciate from Fig. 10.6 how Δ_J is quite always negative, except for small penalizations of the unexpected breaks for which Δ_J is positive, but still really close to zero. Therefore we can conclude that the introduction of PdM policy for dealing with filament maintenances instead of PvM approaches is completely justified.

10.4 Conclusions

In this Chapter, a PdM system for a Ion-Implanter equipment that aims at predicting filament breaks in the source has been presented. The module is based on regularization statistical methods exploiting the knowledge at each process iterations of the tool variables.

The module described in this work has been shown to guarantee better performances than classical PvM approaches. The proposed approach can be extended to other maintenance problems where R2F historical data are available. Since costs related to unexpected breaks and equipment downtime may change during time, it is convenient to provide process engineers with the performances of the PdM module with all the thresholds and let them choose the action policy that minimizes the total cost at the moment.

As future developments of the presented work, the prediction of filament breaks can be tackled as a classification problem with techniques such as Support Vector Machines, with the aim of avoiding the need of predicting the amount batches missing before next breaks and focusing instead on estimating the 'health status' of the filament. To do this, some problems have to be faced:

(i) skewed data, lots of data available for functioning filament and few data for filament broken;

(ii) classification methods provide a distance of the current tool state from a faulty situation. Such concept of distance has to be translated into information on the time before a break that can be used by process engineers.

Part V

Conclusions

11
Conclusions

In this thesis an overview of the major applications of machine learning and automatic control for semiconductor manufacturing have been presented. This is a challenging research area of growing interest; as explained in Section 1.3 and in the various user cases presented during the thesis several of the problems of this area are still open or have just been recently tackled.

The focus of this work has been on Virtual Metrology and Predictive Maintenance modules. The description presented in this thesis of these two topics cannot be considered complete and the aim of this work was not to describe all the aspects related to these two technologies, but to describe the methodological challenges and results of the problems encountered when dealing with practical problems in the industries we had collaborated with during the doctoral studies this thesis summarized.

While Part III, the one related to VM, is quite complete in describing all the state-of-the-art solutions to VM, an equally comprehensive review of PdM methods is difficult to be presented given the fact that PdM problems are peculiar and have different definition and solutions.

Furthermore, several practical aspects of the introduction of APC modules in a real industrial environment were left out on this thesis: issues related to data (handling data extraction, data merging from different data sources, data format, etc.) and also issues related to the different expertises necessary to fully understand production, processes

and tool routines in the Fabs.

Several other APC works (for example Vincent, Stirton, and Poolla (2011) and Prakash, Johnston, Honari, and McLoone (2012)) in the area of statistical modeling/automatic control that cannot be included in the categories of Virtual Metrology/Predictive Maintenance/Fault Detection/Run-to-Run have been presented in recent years underlying the extent of the possibilities of machine learning and control systems in semiconductor manufacturing.

As expressed in Schirru (2011), statistical modeling has always played a major role in the industrial environment, and nowadays, as deterministic techniques struggle to keep track with technology advancements, complex statistical and machine learning methodologies are used to provide predictions of process results (as in Virtual Metrology) or residual equipment lifetime (Predictive Maintenance). The proliferation of these techniques in the Fab environment proves that the introduction of statistical modules for VM, PdM, control and generally APC systems are widely paid off in terms of Return of Investments (ROI).

References

Adamson T., Moore G., Passow M., J.Wong , and Xu Y. Strategies for successfully implementing fab-wide fdc methodologies in semiconductor manufacturing. In *AEC/APC Symposium XVIII*, 2006.

Aizerman A., Braverman E., and Rozoner L. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Ali S. and Silvey S. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28: 131–142, 1966.

Anderson M. and Hanish C. K. An evaluation of the benefits of integrating run-to-run control with scheduling and dispatching systems. *IEEE Transactions on Semiconductor Manufacturing*, 20:386–392, 2007.

Aronszajn N. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.

Arulampalam M., Maskell S., Gordon N., and Clapp T. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.

Baly R. and Hajj H. Wafer classification using support vector machines. *IEEE Transactions on Semiconductor Manufacturing*, 25:373–383, 2012.

Barron A. Chemical vapor deposition, July 2009. URL `http://cnx.org/content/m25495/1.2/`.

Barry D. and Hartigan J. A. A bayesian analysis for change point problems. *Journal of The American Statistical Association*, 88:309–319, 1993.

Basseville M. and Nikiforov I. *Detection of Abrupt Changes: Theory and Application.* Prentice-Hall, 1993.

Besnard J. and Toprac A. Wafer-to-wafer virtual metrology applied to run-to-run control. In *ISMI symposium on manufacturing effectiveness*, 2006.

Boning D., Moyne W., Smith T., Moyne J., Telfeyan R., Hurwitz A., Shellman S., and Taylor J. Run by run control of chemical-mechanical polishing. *IEEE Transactions on Components, Packaging and Manufacturing Technology, Part C: Manufacturing*, 19:307–314, 1996.

Botev Z. Kernel density estimation using matlab, Retrieved on 30th October 2012. URL www.mathworks.us/matlabcentral/fileexchange/14034.

Botev Z., Grotowski J., and Kroese D. Kernel density estimation via diffusion. *The Annals of Statistics*, 38:2916–2957, 2010.

Box G., Jenkins G., and Reinsel G. *Time Series Analysis: Forecasting and Control*. Prentice-Hall, 1964.

Buhmann M. *Radial Basis Functions: Theory and Implementations*, volume 12. Cambridge University Press, 2003.

Butler S. and Ringwood J. Particle filters for remaining useful life estimation of abatement equipment used in semiconductor manufacturing. In *Proc. IEEE Conf. on Control and Fault-Tolerant Systems*, pages 436–441, 2010.

Chang C. and Chao T. Wafer cleaning technology. In *USLI Technology*. New York: McGraw-Hill, 1996.

Chang C. *ULSI technology*. World Scientific, 1997.

Chen A. and Blue J. Recipe-indipendent indicator for tool health diagnosis and predictive maintenance. *IEEE Transactions on Semiconductor Manufacturing*, 22: 522–535, 2009.

Chen A. and Guo R.-S. Age-based double ewma controller and its application to cmp processes. *IEEE Transactions on Semiconductor Manufacturing*, 14:11–19, 2001.

Chen P., Wu S., Lin J., Ko F., Lo H., Wang J., Yu C., and Liang M. Virtual metrology: A solution for wafer to wafer advanced process control. In *IEEE International Symposium on Semiconductor Manufacturing*, pages 155–157. IEEE, 2005.

Cheng F.-T., Chen Y.-T., Su Y.-C., and Zeng D.-L. Evaluating reliance level of a virtual metrology system. *IEEE Transactions on Semiconductor Manufacturing*, 21: 92–103, 2008.

Cheng F.-T., Huang H.-C., and Kao C. Dual-phase virtual metrology scheme. *IEEE Transactions on Semiconductor Manufacturing*, 20:566–571, 2007.

Cheng H. Dielectric and polysilicon film deposition. In *USLI Technology*. New York: McGraw-Hill, 1996.

Cheng K.-Y. Molecular beam epitaxy technology of iiiŰv compound semiconductors for optoelectronic applications. *Proceedings of the IEEE*, 85:1694–1714, 1997.

Chiuso A. On the relation between cca and predictor-based subspace identification. *IEEE Transaction on Automatic Control*, 52:1795–1812, 2007.

Chiuso A. and Picci G. Prediction error vs. subspace methods in closed-loop identification. In *Proc. 16th IFAC World Congr., Prague*, 2005.

Chou P.-H., Wu M.-J., and Chen K.-K. Integrating support vector machine and genetic algorithm to implement dynamic wafer quality prediction system. *Expert Systems with Applications*, 37:4413–4424, 2010.

Cox D. Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 23:414–422, 1961.

Dai H.-J., Chang Y.-C., R.T.-H. , and Tsai W.-L. H. New challenges for biological text-mining in the next decade. *Journal of Computer Science and Technology*, 25: 169–179, 2010.

DiPalma F. *Algoritmi end-of-line per la diagnosi di processo nella fabbricazione di dispositivi a semiconduttore*. PhD thesis, (in Italian) Universitá degli Studi di Pavia, 2005.

Douchet A., deFreitas N., and Gordon N. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

Edgar T., Butler S., Campbell W., Pfeiffer C., Bode C., Hwang S., Balakrishnan K., and Hahn J. Automatic control in microelectronics manufacturing: Practices, challenges, and possibilities. *Automatica*, 36:1567Ű–1603, November 2000.

Efron B., Hastie T., Johnstone I., and Tibshirani R. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

ENIAC . Ju projects 120005 - improve project profile, Retrieved on 30th October 2012. URL `www.eniac.eu/web/communication/publications.php`.

Evgeniou T., Micchelli C., and Pontil M. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

Facco P., Doplicher F., Bezzo F., and Barolo M. Moving average pls soft sensor for online product quality estimation in an industrial batch polymerization process. *Journal of Process Control*, 19:520–529, 2009.

Ferreira A., Roussy A., and Conde L. Virtual metrology models for predicting physical measurement in semiconductor manufacturing. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 149–154, 2009.

Friedman J., Hastie T., and Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1–22, 2010.

Friedman J. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data mining and knowledge discovery*, 1(1):55–77, 1997.

Goodlin B., Boning D., Sawin H., and Wise B. Simultaneous fault detection and classification for semiconductor manufacturing tools. *Journal of the Electrochemical Society*, 150:778–784, 2003.

Hastie T., Tibshirani R., and Friedman J. *The Elements of Statistical Learning. Data Mining, Inference and Prediction.* Springer, 2009.

He Q. and Wang J. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. *IEEE Trans. on Sem. Manufacturing*, 20:345 – 354, 2007.

He Q. and Wang J. Large-scale semiconductor process fault detection using a fast pattern recognition-based method. *IEEE Transactions on Semiconductor Manufacturing*, 23:194–200, 2010.

Hecht-Nielsen R. Theory of the backpropagation neural network. In *International Joint Conference on Neural Networks*, 1989.

Himmel C. D., Kim B., and May G. S. A comparison of statistically based and neural network models of plasma etch behaviour. In *International Semiconductor Manufacturing Science Symposium*, pages 124–129, 1992.

Himmel C. D. and May G. S. Advantages of plasma etch modeling using neural networks over statistical techniques. *IEEE Transactions on Semiconductor Manufacturing*, 6:103–111, 1993.

Hoerl A. and Kennard R. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12:55–67, 1970.

Huang H.-C., Su Y.-C., Cheng F.-T., and Jian J.-M. Development of a generic virtual metrology framework. In *IEEE Conference on Automation Science and Engineering*, pages 282–287, 2007.

Huang Y.-T., Cheng F.-T., and Hung M.-H. Developing a product quality fault detection scheme. In *IEEE International Conference on Robotics and Automation*, pages 927–932, 2009.

Huang Y.-T., Huang H.-C., Cheng F.-T., Liao T.-S., and Chang F.-C. Automatic virtual metrology system design and implementation. In *IEEE Conference on Automation Science and Engineering*, pages 223–229, 2008.

Hung M.-H., Lin T.-H., Cheng F.-T., and Lin R.-C. A novel virtual metrology scheme for predicting cvd thickness in semiconductor manufacturing. *IEEE/ASME Transactions on Mechatronics*, 12:308–316, 2007.

Hyde J., Doxsey J., and Card J. The use of unified apc/fd in the control of a metal etch area. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, pages 237–240, 2004.

IMPROVE . Official website, Retrieved on 30th October 2012. URL `www.eniac-improve.eu`.

Ito R. and Okazaki S. Pushing the limits of lithography. *Nature*, 406(6799):1027–1031, 2000.

Jaeger R. *Introduction to Microelectronic Fabrication.* Prentice Hall, 2001.

Jones M., Marron J., and Sheater S. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11:337–381, 1996.

Kalir A. Segregating preventive maintenance work for factory performance improvement. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages x–x, 2012.

Kalman R. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

Kang P., Lee H., Cho S., Kim D., Park J., Park C.-K., and Doh S. A virtual metrology system for semiconductor manufacturing. *Expert Systems with Applications*, 36:12554–12561, 2009.

Kao C.-A., Cheng F.-T., and Wu W.-M. Preliminary study of run-to-run control utilizing virtual metrology with reliance index. In *IEEE Conference on Automation Science and Engineering*, 2011.

Kao C.-A., Cheng F.-T., Wu W.-M., Kong F., and Huang H. Run-to-run control utilizing virtual metrology with reliance index. *IEEE Transactions on Semiconductor Manufacturing*.

Karayiannis N. and Mi G. Growing radial basis neural networks: Merging supervised and unsupervised learning with network growth techniques. *IEEE Transactions on Neural Networks*, 8(6):1492–1506, 1997.

Khan A. A., Moyne J. R., and Tilbury D. M. An approach for factory-wide control utilizing virtual metrology. *IEEE Transactions on Semiconductor Manufacturing*, 20: 364–375, 2007.

Khan A. A., Moyne J. R., and Tilbury D. M. Virtual metrology and feedback control for semiconductor manufacturing processes using recursive partial least squares. *Journal of Process Control*, 18:961–974, 2008.

Khashei M. and Bijari M. An artificial neural network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications*, 37:479–489, 2010.

Konuma M. *Film Deposition by Plasma Techniques*. Springer-Verlang, 1992.

Kullback S. and Leibler R. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

Kurz D., Kaspar J., and Pilz J. Dynamic maintenance in semiconductor manufacturing using bayesian networks. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 238–243. IEEE, 2011.

Larimore W. Canonical variate analysis in identification, filtering and adaptive control. In *Proc. 29th IEEE Conference on Decision and Control*, pages 596–604, 1990.

Li Q. and Lin N. The bayesian elastic net. *Bayesian Analysis*, 5(1):151–170, 2010.

Lii Y. Etching. In *USLI Technology*. New York: McGraw-Hill, 1996.

Lilliefors H. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 62:399–402, 1967.

Lin T.-H., Cheng F.-T., Wu W.-M., Kao C.-A., Ye A.-J., and Chang F.-C. Nn-based key-variable selection method for enhancing virtual metrology accuracy. *IEEE Transactions on Semiconductor Manufacturing*, 22:204–211, 2009.

Lin T.-H., Cheng F.-T., Ye A.-J., and Hung M.-H. A novel key-variable shifting algorithm for virtual metrology. In *IEEE International Conference on Robotics and Automation*, pages 3636–3641, 2008.

Lin T., Hung M., Lin R., and Cheng F. A virtual metrology scheme for predicting cvd thickness in semiconductor manufacturing. In *IEEE Conference on Robotics and Automation*, pages 1054–1059. IEEE, 2006. ISBN 0780395050.

Liu J. and Chen R. Sequential monte carlo methods for dynamical systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

Ljung L. *System Identification - Theory For the User.* Prentice Hall, 1999.

Lu Y., Sundararajan N., and Saratchandran P. Performance evaluation of a sequential minimal radial basis function (rbf) neural network learning algorithm. *IEEE Transactions on Neural Networks*, 9:308–318, 1998.

Lynn S., Ringwood J., Ragnoli E., McLoone S., and MacGearailt N. Virtual metrology for plasma etch using tool variables. In *Advanced Semiconductor Manufacturing Conference*, pages 143–148, 2009.

McKenna C. A personal historical perspective of ion implantation equipment for semiconductor applications. In *Conference on Ion Implantation Technology*, pages 1 – 19, 2000.

Miller K. *Multidimensional Gaussian distributions.* Wiley, 1964.

Mizutani T. Correct substrate temperature monitoring with infrared optical pyrometer for molecular-beam epitaxy of iiiŰv semiconductors. *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures*, 6:1671–1677, 1988.

Mobley R. *An Introduction to Predictive Maintenance.* Butterworth-Heinemann, 2002.

Monahan K. Enabling dfm and apc strategies at the 32nm technology node. In *IEEE International Symposium on Semiconductor Manufacturing*, pages 398–401, 2005.

Monostori L. Ai and machine learning techniques for managing complexity, changes and uncertainties in manufacturing. *Engineering Applications of Artificial Intelligence*, 16:277–291, 2003.

Montgomery D. *Introduction to Statistical Quality Control.* Wiley-India, 2007.

Moore T., Harner B., Kestner G., Baab C., and Stanchfield J. Intel's fdc proliferation in 30mm hvm: Progress and lessons learned. In *AEC/APC Symposium XVIII*, 2006.

Moreno P., Ho P., and Vasconcelos N. A kullback-leibler divergence based kernel for svm classification in multimedia applications. *Advances in Neural Information Processing Systems*, 16:1385–1393, 2003.

Muller K.-R., Mika S., Ratsch G., Tsuda K., and Scholkopf B. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12: 181–201, 2001.

Nakamura K. Lithography in ulsi technology. In *USLI Technology*. New York: McGraw-Hill, 1996.

Pampuri S., Schirru A., DeLuca C., and DeNicolao G. Proportional hazard model with l1 penalization applied to predictive maintenance in semiconductor manufacturing. In *IEEE Conf. on Automation Science and Engineering*, 2011a.

Pampuri S., Schirru A., Fazio G., and DeNicolao G. Multilevel lasso applied to virtual metrology in semiconductor manufacturing. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 244–249. IEEE, 2011b.

Pampuri S., Schirru A., Susto G., DeNicolao G., Beghi A., and DeLuca C. Multistep virtual metrology approaches for semiconductor manufacturing processes. In *8th IEEE International Conference on Automation Science and Engineering*, 2012.

Pasadyn A. and Toprac A. Method and apparatus for dynamic sampling of a production line, 2002.

Patel N. and Jenkins S. Adaptive optimization of run-to-run controllers: the ewma example. *IEEE Transactions on Semiconductor Manufacturing*, 13:97 – 107, 2000.

Picard R. R. and Cook R. D. Cross-validation of regression models. *Journal of the American Statistical Association*, 79:575–583, 1984.

Pierson H. *Handbook of Chemical Vapor Deposition*. Noyes Publications, 1992.

Pillonetto G., Dinuzzo F., and DeNicolao G. Bayesian online multitask learning of gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:193–205, 2010.

Platt J. C. Fast training of support vector machines using sequential minimal optimization. In *Advancec in Kernel Methods - Support Vector Learning*. MIT Press, 1999.

Pollard D. *A User's Guide to Measure Theoretic Probability*. Cambridge University Press, 2002.

Prakash P., Johnston A., Honari B., and McLoone S. Optimal wafer site selection using forward selection component analysis. In *23rd IEEE/SEMI Conference on Advanced Semiconductor Manufacturing Conference*, pages 91–96, 2012.

Principe J., Xu D., and Fisher J. Information theoretic learning. *Unsupervised adaptive filtering*, 1:265–319, 2000.

Quirk M. and Serda J. *Semiconductor Manufacturing Technology*. Prentice Hall, 2001.

Ragnoli E., McLoone S., Lynn S., Ringwood J., and MacGearailt N. Identifying key process characteristics and predicting etch rate from high-dimension datasets. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 2009.

Ramirez I., Lecumberry F., and Sapiro G. Universal priors for sparse modeling. In *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 197–200. IEEE, 2010.

Ramjee R. and N.Crato and B. R. A note on moving average forecasts of long memory processes with an application to quality control. *International Journal of Forecasting*, 18:291–297, 2002.

Rao C. The use and interpretation of principal component analysis in applied research. *Sankhya: The Indian Journal of Statistics*, 26:329–358, 1964.

Ringwood J., Lynn S., Bacelli G., Ma B., Ragnoli E., and McLoone S. Estimation and control in semiconductor etch: Practice and possibilities. *IEEE Transactions on Semiconductor Manufacturing*, 23:87–98, 2010.

Ristic B., Arulampalam S., and Gordon N. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

Rudin W. *Real and Complex Analysis*. McGraw-Hill, 1966.

Rying E. *A Novel Focused Local Learning Wavelet Network with Application to In Situ Monitoring during Selective Silicon Epitaxy.* PhD thesis, North Carolina State University, 2001.

Sachs E., Hu A., and Ingolfsson A. Run by run process control: Combining spc and feedback control. *IEEE Transactions on Semiconductor Manufacturing*, 8:26–43, 1995.

Sarmiento T., Hong S., and May G. Fault detection in reactive ion etching systems using one-class support vector machines. In *IEEE/SEMI Advanced Semiconductor Manufacturing Conference and Workshop*, 2005.

Schaller R. Moore's law: Past, present and future. *IEEE Spectrum*, 34:52–59, 1997.

Schirru A. *Statistical and Machine Learning Methods for Predictive Maintainance and Virtual Metrology in Semiconductor Manufacturing.* PhD thesis, (in Italian) Universitá degli Studi di Pavia, 2011.

Schirru A., Pampuri S., DeLuca C., and DeNicolao G. Multilevel kernel methods for virtual metrology in semiconductor manufacturing. In *IFAC World Congress*, volume 18, pages 11614–11621, 2011.

Schirru A., Pampuri S., DeLuca C., and DeNicolao G. Virtual sensors for semiconductor manufacturing: A nonparametric approach-exploiting information theoretic learning and kernel machines. *Informatics in Control, Automation and Robotics*, 173: 175–193, 2012a.

Schirru A., Susto G., Pampuri S., and McLoone S. Learning from time series: Supervised aggregative feature extraction. In *51st IEEE Conference on Decision and Control*, 2012b.

Schirru A., Susto G., Pampuri S., and McLoone S. Supervised aggregated feature extraction for functional regression in time series spaces. *IEEE Transactions on Neural Networks and Learning Systems*, Submitted:xx, 2012c.

Schirru A., Pampuri S., and DeNicolao G. Multilevel statistical process control of asynchronous multi-stream processes in semiconductor manufacturing. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 57–62, 2010a.

Schirru A., Pampuri S., and DeNicolao G. Particle filtering of hidden gamma processes for robust predictive maintenance in semiconductor manufacturing. In *IEEE Conference on Automation Science and Engineering (CASE)*, 2010b.

Scholkopf B. and Smola A. *Learning with Kernels*. The MIT Press, 2001.

Scott D. *Multivariate Density Estimation. Theory, Practice and Visualization*. John Wiley & Sons, 1992.

Shao J. Linear model selection by cross-validation. *Journal of American Statistical Association*, 88:486–494, 1993.

Sheater S. and Jones M. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53:683–690, 1991.

Su A.-J., Yu C.-C., and Ogunnaike B. A. On the interaction between measurement strategy and control performance in semiconductor manufacturing. *Journal of Process Control*, 18:266–276, 2008.

Susto G. and Beghi A. An information theory-based approach to data clustering for virtual metrology and soft sensors. In *3rd International conference on Circuits, Systems, Control, Signals*, pages 198–203, 2012a.

Susto G. and Beghi A. Least angle regression for semiconductor manufacturing modeling. In *IEEE Multi-Conference on Systems and Control*, 2012b.

Susto G. and Beghi A. A virtual metrology system based on least angle regression and statistical clustering. *Applied Stochastic Models in Business and Industry*, x:x, 2012c.

Susto G., Beghi A., and DeLuca C. A predictive maintenance system for silicon epitaxial deposition. In *IEEE Conference on Automation Science and Engineering (CASE)*, pages 262–267, 2011a.

Susto G., Beghi A., and DeLuca C. A virtual metrology system for predicting cvd thickness with equipment variables and qualitative clustering. In *IEEE Conference on Emerging Technologies & Factory Automation*, 2011b.

Susto G., Beghi A., and DeLuca C. A predictive maintenance system for epitaxy processes based on filtering and prediction techniques. *IEEE Transactions on Semiconductor Manufacturing*, 25(4):638–649, November 2012a.

Susto G., Pampuri S., Schirru A., and Beghi A. Optimal tuning of epitaxy pyrometers. In *Proceeding of 23rd IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 2012b.

Susto G., Pampuri S., Schirru A., DeNicolao G., McLoone S., and Beghi A. Automatic control and machine learning for semiconductor manufacturing: Review and challenges. In *10th European Workshop on Advanced Control and Diagnosis*, 2012c.

Susto G., Schirru A., Pampuri S., and Beghi A. A predictive maintenance system based on regularization methods for ion-implantation. In *Proceeding of 23rd IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 2012d.

Susto G., Schirru A., Pampuri S., DeNicolao G., and Beghi A. An information-theory and virtual metrology-based approach to run-to-run semiconductor manufacturing control. In *8th IEEE International Conference on Automation Science and Engineering*, 2012e.

Tachikawa T. Assembly and packaging. In *USLI Technology*. New York: McGraw-Hill, 1996.

Tibshirani R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288, 1996.

Toprac A. J., Downey D. J., and Gupta S. Run-to-run control process for controlling critical dimensions, 1999.

Touchette H. and Lloyd S. Information-theoretic limits of control. *Physical review letters*, 84(6):1156–1159, 2000.

VanOverschee P. and Moor B. D. N4sid: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30:75–93, 1994.

VanOverschee P. and Moor B. D. *Subspace Identification for Linear Systems: Theory - Implementation - Applications*. Kluwer Academic Publishers, 1996.

Verhaegen M. Identification of the deterministic part of mimo state space models given in innovations form from input-output data. *Automatica*, 31:1853–1864, 1995.

Vincent T., Stirton J., and Poolla K. Metrology sampling strategies for process monitoring applications. *IEEE Transactions on Semiconductor Manufacturing*, 24: 489–498, 2011.

Wand M. and Jones M. *Kernel Smoothing*. Chapman & Hall, 1995.

Wu M.-F., Lin C.-H., Wong D. S.-H., Jang S.-S., and Tseng S.-T. Performance analysis of ewma controllers subject to metrology delay. *IEEE Transactions on Semiconductor Manufacturing*, 21:413–425, 2008.

Wu S., Gebraeel N., Lawley M., and Yih Y. A neural network integrated decision support system for condition-based optimal predictive maintenance policy. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37: 226–236, 2007.

Zeng D. and Spanos C. Virtual metrology modeling for plasma etch operations. *IEEE Transactions on Semiconductor Manufacturing*, 22:419–431, 2009.

Zhang C., Deng H., and Baras J. Run-to-run control methods based on the dhobe algorithm. *Automatica*, 39:35–45, 2003.

Zou H. and Hastie T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Methodological)*, 67:301–320, 2005.

Acknowledgments

First of all, I would like to thank my advisor, Alessandro Beghi, for his guidance during the PhD. Alessandro has been a great teacher, a pragmatic and smart boss and a good friend. My PhD wouldn't have been such a wonderful experience without his help and support.

During the months spent abroad, I had the privilege of having a 'second advisor', Seán McLoone. I cannot fully express my gratitude for the opportunity and support he has given me, working with such a passionate and organized person was both a pleasure and inspiring.

Andrea Schirru and Simone Pampuri are great companions and friends, I considered myself blessed to work with such skilled, creative and non-ordinary people.

I wish to extend my gratitude to all the people involved in the European Project IMPROVE, especially Giuseppe De Nicolao and Cristina De Luca, and all the people I have collaborated with in industries.

My PhD years wouldn't have been so pleasant without all the officemates I had, in Padova and in Maynooth. When I'll look back to my PhD years, the memories of the Bang!/Worms games and of the passionate discussions during the breaks will always put a smile on my face.

I wish to thank some people outside the academic world: in the first place, my parents Filomena and Stefano. They have faced a lot in the past years, but they have been brave, strong and wise. Their support and love is the root of everything that I have achieved and that I am.

Finally I want to thank my future wife Angela. Her love and unconditionally support, always being by my side, are the main reasons for my strength, my constancy, my tenacity. I cannot adequately express how much she means to me.

To the aforementioned and to the other friends that I have not cited: thanks for being part of my journey.

Layout by Saverio Bolognani
Creative Commons Attribution-NonCommercial 3.0 Italy License.