# An Integrated Physics-Informed Process Control Framework and Its Applications to Semiconductor Manufacturing

Wei-Ting Yang

▶ **To cite this version:**

Wei-Ting Yang. An Integrated Physics-Informed Process Control Framework and Its Applications to Semiconductor Manufacturing. Other. Université de Lyon, 2020. English. NNT : 2020LYSEM004 . tel-03461289

**HAL Id: tel-03461289**
**https://theses.hal.science/tel-03461289**

Submitted on 1 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N°d'ordre NNT : 2020LYSEM004

# THÈSE de DOCTORAT DE L'UNIVERSITÉ DE LYON

opérée au sein de
**l'École des Mines de Saint-Étienne**

**École Doctorale** N° 488
**Sciences, Ingénierie, Santé**

**Spécialité de doctorat** : Microélectronique

Soutenue publiquement le 20/01/2020, par :

Wei-Ting Yang

---

# An Integrated Physics-Informed Process Control Framework and Its Applications to Semiconductor Manufacturing

---

Devant le jury composé de :

| | |
|---|---|
| Berti-Equille Laure, Professeur, Université Aix-Marseille | Présidente |
| Castanier Bruno, Professeur, Université d'Angers | Rapporteur |
| Zamaï Éric, Professeur, Institut National des Sciences Appliquées de Lyon | Rapporteur |
| Verdier Ghislain, Maître de conférences, Université de Pau et des Pays de l'Adour | Examinateur |
| Roussy Agnès, Maître de conférences, Mines St Etienne | Directrice de thèse |
| Blue Jakey, Maître-assistant, National Taiwan University - Taiwan | Co-encadrant |
| Reis Marco, Professeur, Université de Coimbra - Portugal | Co-encadrant |
| Juge Michel, Ingénieur, STMicroelectronics | Invité |
| Pinaton Jacques, Ingénieur, STMicroelectronics | Invité |

| Spécialités doctorales | Responsables : | Spécialités doctorales | Responsables |
|---|---|---|---|
| SCIENCES ET GENIE DES MATERIAUX | K. Wolski Directeur de recherche | MATHEMATIQUES APPLIQUEES | O. Roustant, Maître-assistant |
| MECANIQUE ET INGENIERIE | S. Drapier, professeur | INFORMATIQUE | O. Boissier, Professeur |
| GENIE DES PROCEDES | F. Gruy, Maître de recherche | SCIENCES DES IMAGES ET DES FORMES | JC. Pinoli, Professeur |
| SCIENCES DE LA TERRE | B. Guy, Directeur de recherche | GENIE INDUSTRIEL | N. Absi, Maitre de recherche |
| SCIENCES ET GENIE DE L'ENVIRONNEMENT | D. Graillot, Directeur de recherche | MICROELECTRONIQUE | Ph. Lalevée, Professeur |

## EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'Etat ou d'une HDR)

| | | | | |
|---|---|---|---|---|
| ABSI | Nabil | MR | Génie industriel | CMP |
| AUGUSTO | Vincent | CR | Image, Vision, Signal | CIS |
| AVRIL | Stéphane | PR2 | Mécanique et ingénierie | CIS |
| BADEL | Pierre | MA(MDC) | Mécanique et ingénierie | CIS |
| BALBO | Flavien | PR2 | Informatique | FAYOL |
| BASSEREAU | Jean-François | PR | Sciences et génie des matériaux | SMS |
| BATTON-HUBERT | Mireille | PR2 | Sciences et génie de l'environnement | FAYOL |
| BEIGBEDER | Michel | MA(MDC) | Informatique | FAYOL |
| BLAYAC | Sylvain | MA(MDC) | Microélectronique | CMP |
| BOISSIER | Olivier | PR1 | Informatique | FAYOL |
| BONNEFOY | Olivier | PR | Génie des Procédés | SPIN |
| BORBELY | Andras | MR(DR2) | Sciences et génie des matériaux | SMS |
| BOUCHER | Xavier | PR2 | Génie Industriel | FAYOL |
| BRODHAG | Christian | DR | Sciences et génie de l'environnement | FAYOL |
| BRUCHON | Julien | MA(MDC) | Mécanique et ingénierie | SMS |
| CAMEIRAO | Ana | MA(MDC) | Génie des Procédés | SPIN |
| CHRISTIEN | Frédéric | PR | Science et génie des matériaux | SMS |
| DAUZERE-PERES | Stéphane | PR1 | Génie Industriel | CMP |
| DEBAYLE | Johan | MR | Sciences des Images et des Formes | SPIN |
| DEGEORGE | Jean-Michel | MA(MDC) | Génie industriel | Fayol |
| DELAFOSSE | David | PR0 | Sciences et génie des matériaux | SMS |
| DELORME | Xavier | MA(MDC) | Génie industriel | FAYOL |
| DESRAYAUD | Christophe | PR1 | Mécanique et ingénierie | SMS |
| DJENIZIAN | Thierry | PR | Science et génie des matériaux | CMP |
| BERGER-DOUCE | Sandrine | PR1 | Sciences de gestion | FAYOL |
| DRAPIER | Sylvain | PR1 | Mécanique et ingénierie | SMS |
| DUTERTRE | Jean-Max | MA(MDC) | | CMP |
| EL MRABET | Nadia | MA(MDC) | | CMP |
| FAUCHEU | Jenny | MA(MDC) | Sciences et génie des matériaux | SMS |
| FAVERGEON | Loïc | CR | Génie des Procédés | SPIN |
| FEILLET | Dominique | PR1 | Génie Industriel | CMP |
| FOREST | Valérie | MA(MDC) | Génie des Procédés | CIS |
| FRACZKIEWICZ | Anna | DR | Sciences et génie des matériaux | SMS |
| GARCIA | Daniel | MR(DR2) | Sciences de la Terre | SPIN |
| GAVET | Yann | MA(MDC) | Sciences des Images et des Formes | SPIN |
| GERINGER | Jean | MA(MDC) | Sciences et génie des matériaux | CIS |
| GOEURIOT | Dominique | DR | Sciences et génie des matériaux | SMS |
| GONDRAN | Natacha | MA(MDC) | Sciences et génie de l'environnement | FAYOL |
| GONZALEZ FELIU | Jesus | MA(MDC) | Sciences économiques | FAYOL |
| GRAILLOT | Didier | DR | Sciences et génie de l'environnement | SPIN |
| GROSSEAU | Philippe | DR | Génie des Procédés | SPIN |
| GRUY | Frédéric | PR1 | Génie des Procédés | SPIN |
| HAN | Woo-Suck | MR | Mécanique et ingénierie | SMS |
| HERRI | Jean Michel | PR1 | Génie des Procédés | SPIN |
| KERMOUCHE | Guillaume | PR2 | Mécanique et Ingénierie | SMS |
| KLOCKER | Helmut | DR | Sciences et génie des matériaux | SMS |
| LAFOREST | Valérie | MR(DR2) | Sciences et génie de l'environnement | FAYOL |
| LERICHE | Rodolphe | CR | Mécanique et ingénierie | FAYOL |
| MALLIARAS | Georges | PR1 | Microélectronique | CMP |
| MOLIMARD | Jérôme | PR2 | Mécanique et ingénierie | CIS |
| MOUTTE | Jacques | CR | Génie des Procédés | SPIN |
| NAVARRO | Laurent | CR | | CIS |
| NEUBERT | Gilles | | | FAYOL |
| NIKOLOVSKI | Jean-Pierre | Ingénieur de recherche | Mécanique et ingénierie | CMP |
| NORTIER | Patrice | PR1 | Génie des Procédés | SPIN |
| O CONNOR | Rodney Philip | MA(MDC) | Microélectronique | CMP |
| PICARD | Gauthier | MA(MDC) | Informatique | FAYOL |
| PINOLI | Jean Charles | PR0 | Sciences des Images et des Formes | SPIN |
| POURCHEZ | Jérémy | MR | Génie des Procédés | CIS |
| ROUSSY | Agnès | MA(MDC) | Microélectronique | CMP |
| ROUSTANT | Olivier | MA(MDC) | Mathématiques appliquées | FAYOL |
| SANAUR | Sébastien | MA(MDC) | Microélectronique | CMP |
| SERRIS | Eric | IRD | | FAYOL |
| STOLARZ | Jacques | CR | Sciences et génie des matériaux | SMS |
| TRIA | Assia | Ingénieur de recherche | Microélectronique | CMP |
| VALDIVIESO | François | PR2 | Sciences et génie des matériaux | SMS |
| VIRICELLE | Jean Paul | DR | Génie des Procédés | SPIN |
| WOLSKI | Krzystof | DR | Sciences et génie des matériaux | SMS |
| XIE | Xiaolan | PR0 | Génie industriel | CIS |
| YUGMA | Gallian | CR | Génie industriel | CMP |

Mise à jour : 30/08/2018

# Acknowledgements

I wish to express my sincere gratitude to many people who helped me complete my Ph.D. thesis. First of all, I would like to show my deep appreciation to my supervisors: Agnès Roussy, who provided this opportunity for me to pursue my Ph.D. and gives me a great support in many ways; Jakey Blue, who patiently supervises me through these years and leads me to explore new ideas; Marco Reis, who guides me with many valuable experiences and knowledge, and always gives me kindly encouragements. Without their guidance and persistent help, this thesis would not have been possible.

I would like to acknowledge Bruno Castanier and Éric Zamaï for accepting the invitation to be a member of my committee and reviewing the manuscript. I am grateful to Laure Berti and Ghislain Verdier for their constructive comments and suggestions. Warm words of thanks go to our industry partners, Jacques Pinaton, and Michel Juge, for accepting the invitation and providing valuable inputs from the industry point of view.

My heartfelt appreciation goes to the members of the SFL department for all the advice and kindness I received during these years. It's my pleasure to be a part of this team. I want to extend my special thanks to Karim, Sophia, and Hamideh, who gave me great help on both research and life in general. I am also grateful to my friends, Steffi, Jyshyan, and Peihsin, for the warmest encouragement through this research.

Last but not least, I would like to thank my family: my mother, who is always by my side and keeps me going on; my sisters, Una and Vernis, they are my role model and they give me the courage to take new challenges; my father, for giving me the greatest love and unconditional support.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

In this chapter, a general overview of the contents of this thesis is given. In Section 1.1, the most common process control problems and solutions in semiconductor manufacturing are presented. In Section 1.2, the motivation of this thesis is described, and its scope defined. The main contributions of this thesis are referred in Section 1.3. Finally, an overview of the thesis structure is provided in Section 1.4.

## 1.1 Process Control in Semiconductor Manufacturing

In the last decades, with the continuously increasing demand for integrated circuits (IC) and high-tech products, semiconductor manufacturing has become one of the fast-growing industries. , The existence of advancing manufacturing capabilities is a fundamental requirement for the success in this industry, where the primary task is to produce smaller size chips with high quality while keeping short production cycles and high yields. A wafer is fabricated by a sequence of hundreds of complex processes, such as deposition, lithography, etching, etc. Many factors can affect the quality of final products, including the equipment condition, hidden interactions across processes, random events, human interventions, etc. To reduce variation, a wide variety of process monitoring and control solutions have been proposed and implemented over the years. Statistical Process Control (SPC) systems are implemented for monitoring the quality of processes based on the measurements of sample wafers, which are also called metrology readings. Through statistical control charts, out-of-control wafers can be detected, and corrective actions taken to eliminate the faults (Spanos, 1992; Kourti & MacGregor, 1996; Qin, 2003). Fault Detection and Classification (FDC) is another approach used for either monitoring the state of equipment and the stability of the process. FDC aims to monitor process equipment using large amounts of sensor data, which can be used to establish SPC charts for fault detection. When a fault is detected, the next step is to identify the root cause through classification, finally correcting it before the next run starts.

Although SPC is useful for fault detection and diagnosis, it does not include an automatic mechanism to correct or compensate for the existence of process disturbances (Moyne et al., 2000). Advanced Process Control (APC) systems can further adjust the process deviations in time instead of posterior actions. Run-to-Run (R2R) control is a type of discrete process control system which manipulates process control parameters between "runs". The objective of R2R control is to minimize process drift, shift, or other variability patterns. Depending on the process specifications or equipment types, a "run" can be either a wafer or a batch (Castillo & Hurwitz, 1997). Generally, R2R systems cover two kinds of control schemes: feed-forward control (Stoddard et al., 1994) and feedback control (Butler & Stefani, 1994). Feed-forward control refers to adjust the subsequent process based on the output of preceding operations. For example, the measurement of the Critical Dimension (CD) of the lithography process can be used as an input of the etching process to compensate the estimated post-etch CD deviation. The goal of feedback control is to ensure that the distribution of the process outputs stays centered on the target. Thus, the output of the previous run will be used to update the control parameters

for the incoming run. A classic example is Chemical-Mechanical Polishing (CMP) process, where the output of the CMP process can be vary due to inconsistent removal rate. Through feedback the measurements of the previous run, the polishing time for the next run can be adjusted to fit current equipment state. In this way, process variability can be reduced.

In recent decades, the steady progress of information technologies enable the capability of collecting and storing massive amounts of data, that can be processed using advanced data-driven approaches. These information technologies open new perspectives and opportunities to develop new process control systems. Several solutions based on machine learning algorithms have been proposed, and some of them have been implemented in fabrication. The analytics solutions may have a different goal, but fundamentally they can either directly or indirectly improve process control capabilities. Among the most well-known applications, are Virtual Metrology (VM) and Predictive Maintenance (PdM). VM models enhance the stability of a process from a quality inspection point of view. Given the current limitations in the capacity of metrology tools and the reduced production cycle of wafers, only a few of them can be sampled and sent for inspection. In order to circumvent the technological limitations of inspection tools, VM approaches were developed. The VM systems aim to predict metrology measurements for all wafers based on the information collected from process equipment. Various benefits of implemented VM models have been addressed and proved. Instead of a fixed sampling scheme, output of VM can be used to estimate the failure risk of each wafer and served as the input of the sampling decision system. By generating a dynamic sampling scheme, the high-risk wafers will be sent to inspection, and the corresponding action can be taken in real time if needed.

The conventional R2R control system can be improved by incorporating a VM model. Since metrology data still serve as the main input for controllers, combining actual measurements and predictive measurements offers more information to support control decision making. To be more specific, a VM model includes many real-time equipment signals which are able to reflect the current equipment state. Consequently, the control decision should match better the current status. Unlike VM focus on product inspection, PdM models can support process control by considering equipment health condition. Conventionally, the schedule of Preventive Maintenance (PM) is determined based on engineers' experiences or the instructions provided by equipment vendors. Instead of a rule-based decision making, the PdM is a predictive model based on equipment data. Since these data describe the real-time equipment conditions, a PdM model can provide prognosis of the future state and the associated potential risk. The output of PdM can be used to implement a better maintenance schedule. Several advantages can be associated to PdM, such as preventing unexpected equipment shut-down and reducing low yield products due to unstable process equipment.

To support advanced analytics solutions, a big data platform is crucial not only for high computing requirements but also for maintaining numerous analytics models (Moyne & Iskandar, 2017). Most online advanced analytics models are working based on extensive streams of data, such as sensor data, metrology data, etc. Therefore, handling multiple data sources with various formats are necessary, and high computing capability for complex models is essential as well. In this context, a big data platform for advance fabrication becomes inevitable. In practice, advanced analytics models all face the problem that the possibility of

decaying accuracy due to various reasons, such as process dynamics, equipment deterioration, random events, etc. The existence of self-adapting mechanisms is important to maintain the feasibility of models and their accuracy. Moreover, to development of dedicated models tailored for each equipment (possibly each product as well), results in many coexisting models and requiring proper management and maintenance. In order to reduce the maintenance burden, practical analytics models should be able to scale to all fabrication and to adapt quickly for rapid changes in the production lines.

Abundant analytics solutions promote and support the development of new intelligent process control system, which is an integral part of the roadmap of Smart Manufacturing (SM). Smart Manufacturing is a term generally applied to the improvement in manufacturing operations through integration of systems, linking physical and cyber capabilities, and taking advantage of all information sources and leveraging big data solutions (Moyne & Iskandar, 2017). To achieve this goal, multiple control systems should eventually coordinate together, not only sharing information but also making optimized and consistent decisions.

## 1.2 Motivation and Scope

As described above, several data-driven approaches have been employed in semiconductor manufacturing, namely in the scope of virtual metrology (VM), VM-based R2R control schemes, equipment healthy monitoring and PdM. These approaches are based on models that were usually built individually, and the tasks of maintaining these models are also conducted independently as well. Generally, these models are constructed for specific products and processes. In this context, there can be many models that are implemented in the production line, implying more work on both constructing and maintaining them. The following subsections discuss the potential opportunities from different aspects, including information consolidation, applications, and the integration of different modules.

### 1.2.1 A General-Purpose Model

From the data source point of view, data-driven models essentially are based on some common information. An example is illustrated in Fig. 1.1. The most common data source of these models is sensor data, which is also called FDC data. Depending on the purpose of these models, other information might be included as well, such as metrology data or Manufacturing Execution System (MES) data.

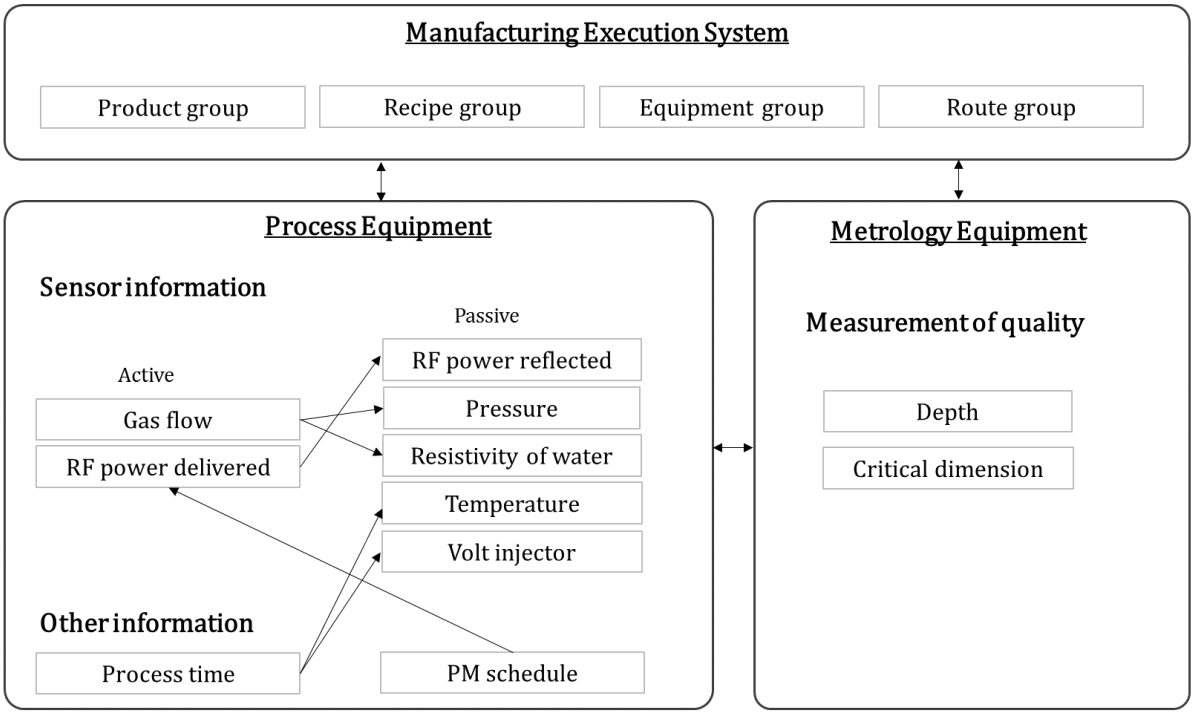


Figure 1.1 An example of collected information in semiconductor manufacturing

Given such a background, an integrated framework seems to be a favorable solution in terms of unifying various data-driven models. The foundation of this framework should be an integrated informatics network, which can deal with common data processing and connect the enormous information. Based on this informatics network, many data-driven models can be generated or constructed for different applications.

Although data-driven approaches have been applied for modeling natural phenomena in many applications, the structure of the inferred model is critically dependent upon the conditions under which data were collected. Data may be corrupted with noise, with missing records, mutual associations (collinearity), etc., limiting the quality of the models achieved. Furthermore, under passive data collection conditions, the relationships found in data are often non-causal and not always aligned with the system physics. In this context, models tend to be unreliable for control and optimization, as well as when used for extrapolation. To overcome these issues, several recent studies have shown that physics-informed approaches, i.e., machine learning algorithms incorporating the domain knowledge, can be substantially more robust and more efficient. Raissi et al. (2017) developed a physics-informed deep learning model which considers physics equations as the prior knowledge, enabling training complex model from the limited data. More studies using physic-informed deep learning for solving Partial Differential Equations (PDEs) can also be found in the literature (Raissi, 2018; Tartakovsky et al., 2018). Yang et al. (2018) proposed a physics-informed Kriging method, where a partial physical model was combined with the Gaussian process to improve the accuracy of the estimates.

Since many studies have proved that the importance of incorporating existing knowledge into data-driven approaches, an integrated framework should be able to include such physical

information as well. In this way, the derived models can leverage the best of two information sources: data and the physics domain knowledge.

Therefore, in this thesis, an *Integrated Physics-Informed Control Framework* is proposed. The framework is schematically described by the block diagram shown in Fig. 1.2. It consists of two main parts, offline stage, and online stage. The goal of offline stage is to build the general-purpose model, which will be supported in a Dynamic Bayesian Network. Three steps are required to obtain this network, namely historical data pre-processing, knowledge-based configuration, and structure learning. Based on a learned DBN, several function modules can be derived and employed for different online applications, such as likelihood monitoring, a VM model for predict metrology, equipment condition inference, and a structured R2R controller. These function modules can either work together or independently.

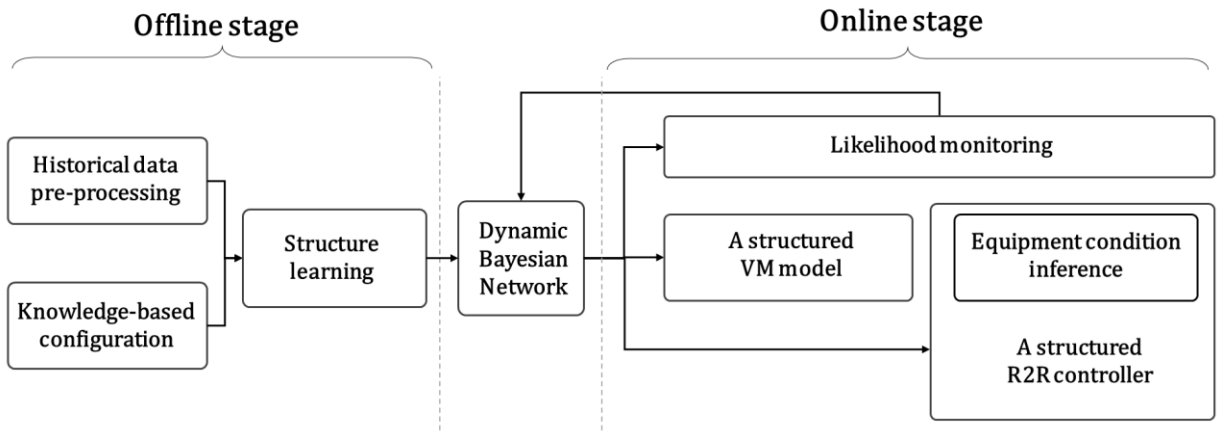


Figure 1.2 The Integrated Physics-Informed Control Framework.

As shown in Fig. 1.1, large amounts of data is available in the semiconductor manufacturing processes. To gather that information and clearly present them as an information network, a DBN is employed as the modeling foundation of the framework. The reasons for choosing a DBN are listed below.

*A. Interpretability*

The directed edges in a DBN indicate the causalities between variables. Missing edges between variables indicate that those variables are irrelevant or conditionally independent. The network is able to provide a global view of all information in the semiconductor manufacturing process. Without using numerous complicated methods, the meaning of these variables and their relationships are more straightforward. As a DBN is presented in a connected form, it is easy to analyze and verify these causalities with domain knowledge experts.

*B. Separability*

Although a DBN presents all information together, it still provides the possibility to extract part of the network given a particular task. Furthermore, local updating of a DBN is also possible, which is easier for maintenance and can be a benefit when the computation time is a concern.

*C. Extensibility*

The example shown in Fig. 1.1 is only a small part of the overall wafer process. Generally, there are hundreds of equipment and a large amount of MES data are involved in semiconductor manufacturing. A DBN can start with a specified product type of certain process equipment, which can provide several applications. If there is an application involving multiple process operation, such as a feed-forward R2R control, then more information collected from the next process operation can be included. Suppose there is a high-mix process consists of several product groups, where the product group can be considered as context information, i.e., a categorical variable. Since a DBN can deal with both continuous variables and categorical variables, it allows to include such context information as well. In this way, the interactions between these context information and other variables can be clearly illustrated.

Given the advantages referred above, a DBN fits well the purpose of the framework: to integrate information and support the development of solutions for different applications. An example of how a DBN can present this information is shown in Fig. 1.3. Each node of a DBN indicates a variable, such as FDC variables, metrology variables. The order of time-slice presents the sequence of wafers. The variables of time slice $t$ present the information is collected from the current wafer, and the variables of time slice $t-1$ mean the information is collected from the previous wafer. The edges demonstrate the causal relationships between variables. Two types of edge imply a different source of causalities; an intra-edge indicates the variable is affected by other variables at the same time slice, and an inter-edge suggests that this variable is influenced by variables at the previous time slice.

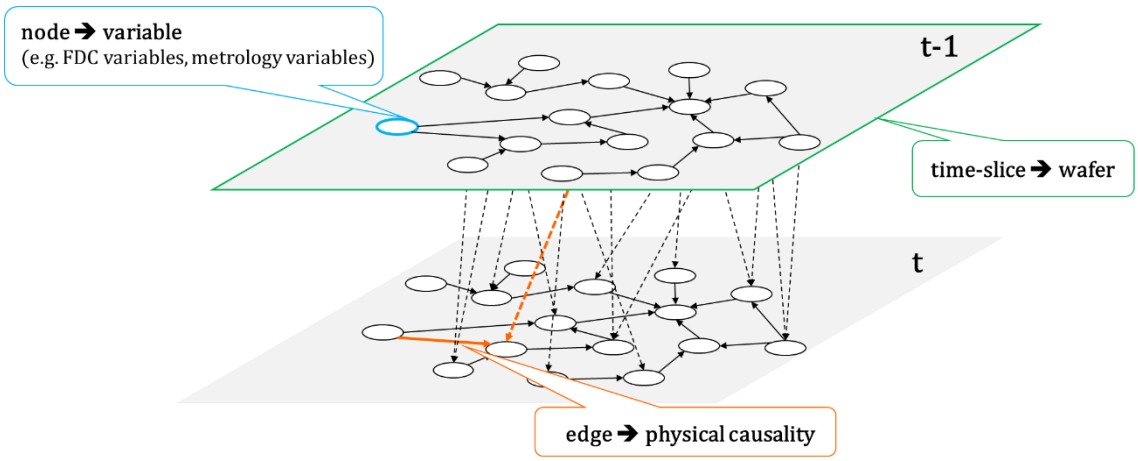


Figure 1.3 Representation of a DBN for semiconductor manufacturing.

A DBN not only can provide a basic overview of how these variables correlate to each other but also can be used for different applications. Suppose a basic DBN consists of three types of variable,

- FDC variables ($X$): the data collected by equipment sensors.
- Controllable variables ($U$): the value of these variables can be manipulated depending on the process requirement, the data can either collected by equipment sensor or other systems.
- Metrology variables ($Y$): the measurements obtained from metrology tool.

An example of a DBN with these variables is illustrated in Fig. 1.4a. Note that the DBN is presented in two-dimension so that the ideas can be clearly addressed, although those nodes should be located in different time slices. Some applications can be realized based on this network. By taking a sub-network of a DBN which includes a metrology variable and its parent nodes, a VM model can be formed (see Fig. 3.4b). Consider a sub-network which consists of the controllable variables, metrology variables, and some FDC variables; a structured R2R controller can be created (see Fig. 1.4c). As shown in Fig. 1.4d, by looking at the overall structure, we are able to monitor the stability of process and fitness of the model. Thus, with a DBN-based framework, only one data-driven model can support a variety of applications. In this way, the work of model maintenance can be simplified, and their connectivity enables more consistent decision making processes.

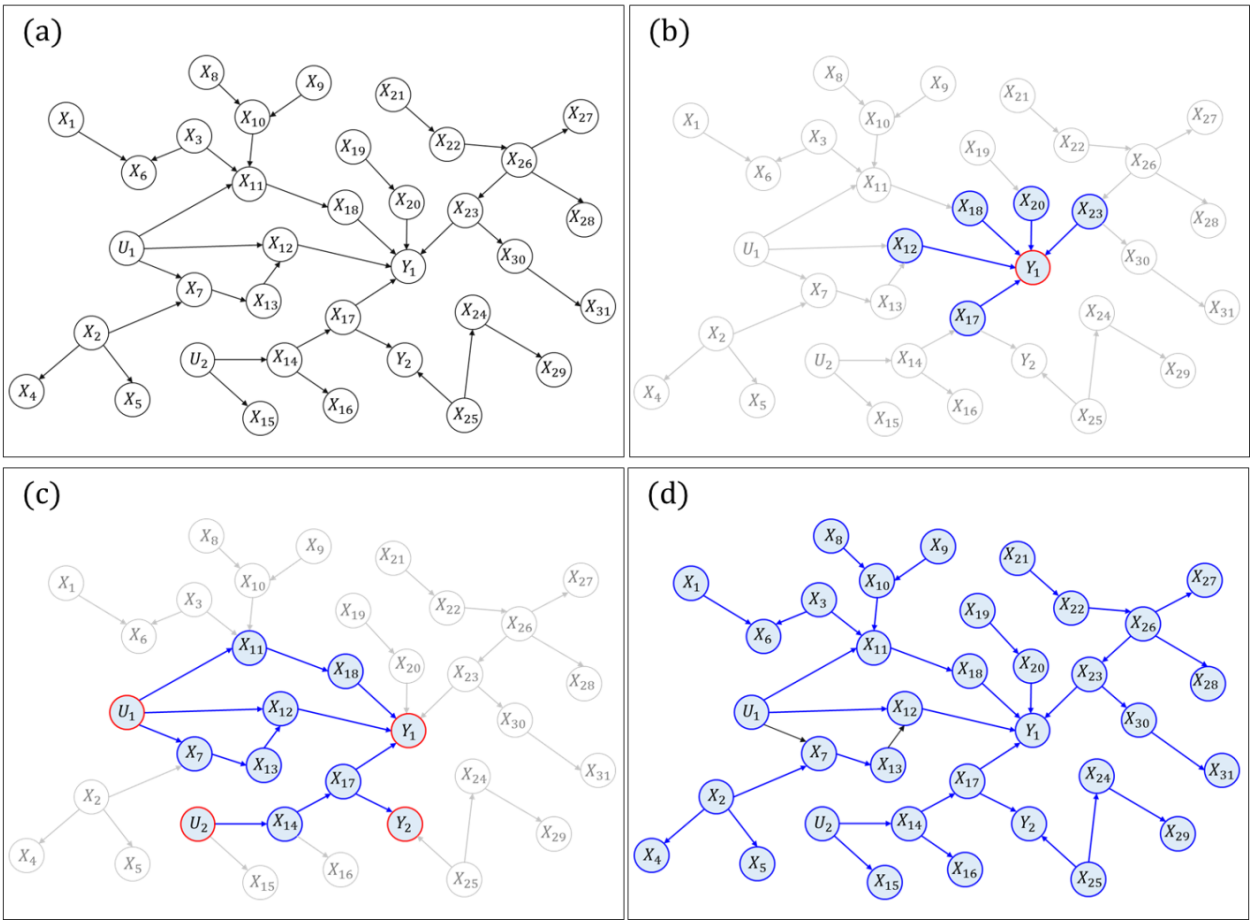


Figure 1.4 (a) An example of a DBN; (b) a sub-network for VM; (c) a sub-network for R2R control; (d) the whole network for monitoring.

## 1.2.2 Applications

In the previous subsection, the potential benefits of the integrated physics-informed control framework have been referred, including the integration ability to consolidate various sources, the flexibility of supporting multiple applications. In this section, three applications will be addressed more fully, with the following sequence: starting with a brief background and current challenges, and then presenting the advantages of a DBN-based module.

### *A. Monitoring*

In semiconductor manufacturing, various types of monitoring systems are employed to make sure the performance of each process meets its requirements. These systems can be categorized based on the target of monitoring, the quality of wafers, or the state of equipment which produce wafers (May, G.S, 2006). The quality of wafer is usually evaluated through the measurements obtained by metrology tools. For example, the depth measurements for the etching process, and several Statistical Process Control (SPC) can be used to monitor their stability. The state of process equipment in advanced manufacturing can be monitored as well. Plenty of sensors are embedded in the equipment, and these sensors can record the equipment conditions with high time resolution. Data collected by these sensors are usually called FDC data. Depending on the process, some statistics of FDC raw data, which is also called FDC indicators, can be used for monitoring with the corresponding SPC charts. Both types of monitoring systems can be used for evaluating the overall process stability. Classical SPC follows a univariate approach which can be inefficient when a large number of features need to be monitored, especially for FDC indicators. In recent decades, many multivariate methods based on FDC data have been investigated to enhance the efficiency of process monitoring, such as Hotelling's $T^2$ (H. Hotelling 1947), and Principal Component Analysis (PCA) (Yue & Tomayaus, 2004; Cherry & Qin, 2006; Good et al., 2010). PCA is able to reduce the dimensionality of data so that only a few dimensions need to be monitored. These methods have been proved their efficiency on fault detection. When a fault is detected, the next task is to identify the root cause. Since these multivariate approaches considered the transformed features which are the combination of original features, some methods have been proposed to trace back the root cause (Good et al., 2010; Blue et al., 2012; Hong et al., 2012).

Instead of using a set of methods for process monitoring and root cause identification, the monitoring mechanism proposed in this thesis is able to cover these two tasks simultaneously. The monitoring mechanism has been embedded in the DBN, which is learned from historical data. Then the learned DBN is used for online monitoring. For each wafer, an individual likelihood score is computed to show the likelihood of the data of this wafer given the current structure. A high likelihood score implies that this wafer is similar to the historical wafers and the underlying relationships among variables are described well by the DBN. When a low likelihood is observed, this indicates that an abnormality (or fault) is detected. Since the likelihood score is computed by the sum of the likelihood of each node, the root cause of abnormality can be easily identified as well. If the low likelihood state is sustained, the irregularities are not random cases and corrective actions should be triggered, such as process maintenance. The continuous low likelihood may imply the process has a drift, and the current DBN is no longer able to represent well the underlying structure. Therefore, a DBN-based monitoring mechanism in the proposed framework can achieve the following tasks: fault detection, fault identification, and auto-updating model if necessary. The auto-updating stage requires a few extra settings, but the computations can be done in the same fashion as during DBN learning, which reduces the implementation complexity. The idea of likelihood monitoring might be used for PdM as well, which is not in the scope of the thesis, but can be an interesting topic for future.

*B. Virtual Metrology*

Virtual metrology models aim at predicting metrology measurements of the wafers based on Fault Detection and Classification (FDC) data. Various modeling methods have been proposed and analyzed in the literature (Cheng & Cheng, 2005; Hung et al., 2007; Khan et al., 2007; Kang et al., 2009; Zeng & Spanos, 2009; Pampuri et al., 2011; Susto et al., 2013; Wan et al., 2014), such as Multiple Linear Regression (MLR) (Andersen & Bro, 2010), Partial Least Squares (PLS) (Wold et al.,1984), Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996), Gaussian Process Regression (GPR) (Rasmussen, 2003), Neural Networks (NN) (Roy et al.,1995), K-Nearest Neighbors (KNN) (Denoeux, 1995), Regression Trees (RT) (Strobl et al., 2009), Support Vector Regression (SVR) (Smola & Schölkopf, 2004), etc. Several comparison studies are reported in the literature as well. Analyzing the existing literature, it is possible to verify that the procedures to derive a VM model essentially consist of comparing different methods. The best method is chosen on a case-by-case basis, as the one providing the lowest overall prediction error.

This pure data-driven approach is highly case dependent and, most importantly, overlooks existing knowledge about the process. However, the practical use of a model should not only consider the accuracy of predictions but also its interpretability, physical meaning, soundness, and consistency with existing knowledge. This is the main goal of the proposed framework, where a new proposal is made for deriving a VM model that is able to integrate both domain knowledge from Subject Matter Experts (SME), such as process engineers, with information induced from the significant data resources available in the process databases. A VM model based on Bayesian Network (BN) is capable of combining the information from both SME and data. In the previous work of this thesis (Yang et al., 2019), a VM model based on Gaussian Bayesian Network (GBN) has been introduced. And its efficiency has been validated through an industrial case study. However, many process variables have temporal causal dependencies with their previous states, which have not been explicit through a GBN model. Therefore, in this thesis, a more general model, DBN, is employed to reveal more hidden causalities.

Based on a DBN, the underlying VM model, a structured VM model, can be extracted by selecting the variables which connect to the target process output (see Fig. 1.4b). Since the missing edges in a DBN indicate the irrelevant or conditional independent relationships between pairs of variables, the indirect and direct interactions can be clearly identified, which corresponds to a feature selection procedure. Therefore, a DBN already embedded a VM model without additional learning. As a structured VM model presents the relationship among the variables in a form of a connected graph, the interpretation of the associations between process variables is very easy and intuitive. We would like to point out that Regression Trees (RT) can also be considered interesting methodologies in terms of model visualization and interpretation. With a RT, it is possible to illustrate how a set of variables contributes to the target variable, a feature that has been practiced in VM applications in semiconductor manufacturing (Kang et al., 2009). However, RT does not provide the structural map of the inner variables as DBNs do. More importantly, during the learning procedure, DBNs provide users with the opportunity to integrate domain knowledge so that one can prevent inappropriate or unfeasible solutions from being contemplated during the estimation stage of the model. This capability is not seen in RT

or other related tree-based methods.

### *C. Run-to-Run Control*

For decades, Run-to-Run (R2R) controllers have been widely implemented in semiconductor manufacturing. They operate over key process parameters on the basis of the metrological measurements acquired from the process and their deviations from the target set-points. The basic and the most well-known controller is the Exponentially Weighted Moving Average (EWMA) controller, which uses a EWMA filter to adjust the control setting. To further correct the underlying drift trend of the process, the controller with double EWMA filters (dEWMA) was proposed (Butler & Stefani, 1994). Several extensions based on dEWMA controllers have been studied to overcome different scenarios, such as MIMO system, non-linearity correlations between variables, or various types of underlying disturbance (Sachs & Hu, 1995; Castillo & Yeh, 1998). Despite the diversity of controllers proposed in the past, all of them strongly rely on metrology data as the main input source of information. However, given the current limitations in the capacity of metrology tools and the reduced product life cycle of wafers, only a few units can be sampled and sent for inspection. In order to enrich the sparse metrology data, in the past decades, Virtual Metrology (VM) has been widely studied in the literature (Chen et al., 2005; Cheng & Cheng, 2005; Kang et al., 2009). Several studies have showed that the R2R control system can benefit from the introduction of a VM model (Khan et al., 2008; Kang et al., 2011, Susto et al., 2012).

Comparing to conventional controllers which only depend on metrology data, controllers with VM models are more successful in linking the relationships between equipment conditions and metrology. However, control decisions may change several equipment conditions and indirectly have an impact on the process output, which have not been explicitly studied before. Furthermore, the dynamic and non-stationary nature of equipment conditions may cause the variability of the process to increase as well. Therefore, before making the control decisions, we should understand more about these hidden dependencies.

A DBN in the proposed framework provides a more transparent picture of the actual underlying causal structure linking the variables. By extracting a sub-network with control relevant variables, a Structured R2R Control (SRC) can be generated. A simple example to address the difference between SRC and other controllers is illustrated in Fig. 1.5, where the dashed lines of each subgraph indicate the sources used for control decision making. The conventional controllers, such as a EWMA controller, determine the control setting only based on the process output, where the real-time equipment conditions are not taken into account (see Fig. 1.5a). Fig. 5.1b shows that if a controller incorporates a VM model, then the predicted measurements can reflect part of equipment states, but the effect of control actions on equipment conditions may be under-assessed. With an SRC, control decisions can be made by taking into account not only metrology but also the multiple interactions between process variables, as shown in Fig. 5.1c. Furthermore, with the graphical visualization, it is easier to discuss with Subject Matter Experts (SME) the physical meanings behind the issue and receives insightful remarks.

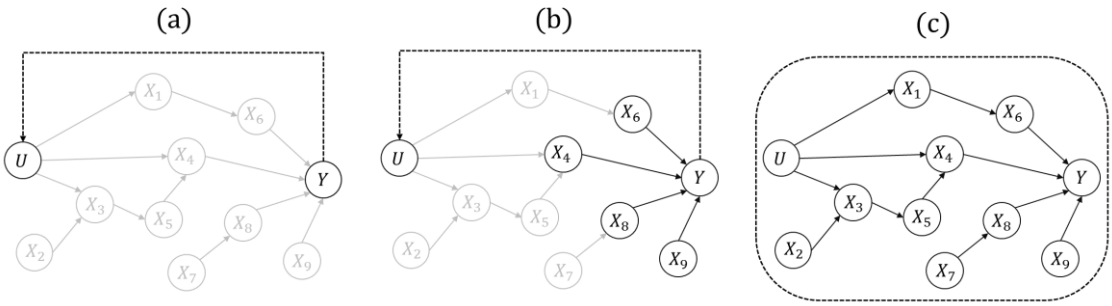


Figure 1.5 The information using for control decision making: (a) conventional controller; (b) controller incorporates a VM model; (c) SRC.

### 1.2.3 Integration of Online Function Modules

Generally, there are several function modules working simultaneously to maintain the quality of the process. Those function modules can be classified into three layers, as shown in Fig. 1.6. The monitoring layer is responsible for primary screening based on physical measurements and readings, including numerous SPC charts for process outputs and major equipment indexes. An Out-Of-Control (OOC) case will trigger the necessary actions, such as rework or equipment maintenance. The modules in the second prognosis aim to proactively predict the future states, including VM for predicting process outputs and Equipment Healthy Management (EHM) for predicting equipment conditions. Those modules are designed to supplement the limited capacity of inspection so that more hidden insights can be discovered. And the general approach is to employ the available data to generate virtual indexes. Having the monitoring tier and the prognosis tier as preliminary, the last one is the control tier. Based on both actual and virtual information, the optimized control actions can be made, such as R2R control or Predictive Maintenance (PdM).

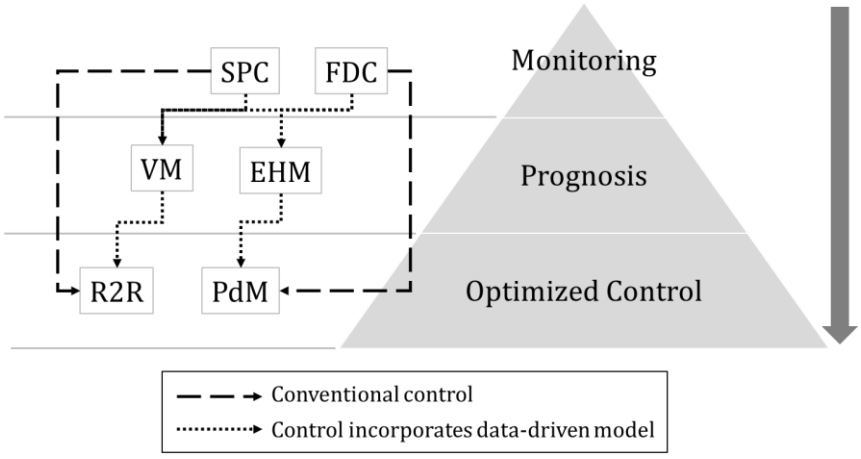


Figure 1.6 Three-layer control system.

Conventionally, the control layer only depends on the monitoring layer to make control decision (see dashed line in Fig. 1.6). In recent years, various data-driven models are proposed to strengthen the prognosis layer, and the output of prognosis models can be the input of control models (see dotted in Fig. 1.6). Although the vertical connection has been well established,

those models are designed and developed independently based on different purposes. In this context, the control procedure and model maintenance can be burdensome.

The proposed framework derives different function modules based on the same foundation – DBN (see Fig. 1.7). Only one data-driven model is employed, which mitigates the load of model maintenance. The likelihood monitoring can be used for both process fault detection and model fitness monitoring. The result can be used to decide if the following prognosis and control should proceed, or if the DBN need to be updated. The output of the VM model can be the input of SRC when the actual measurement is not available. The SRC includes the equipment condition inference in the determination of the control settings. With this integrated control system, the function modules can be operated consistently.

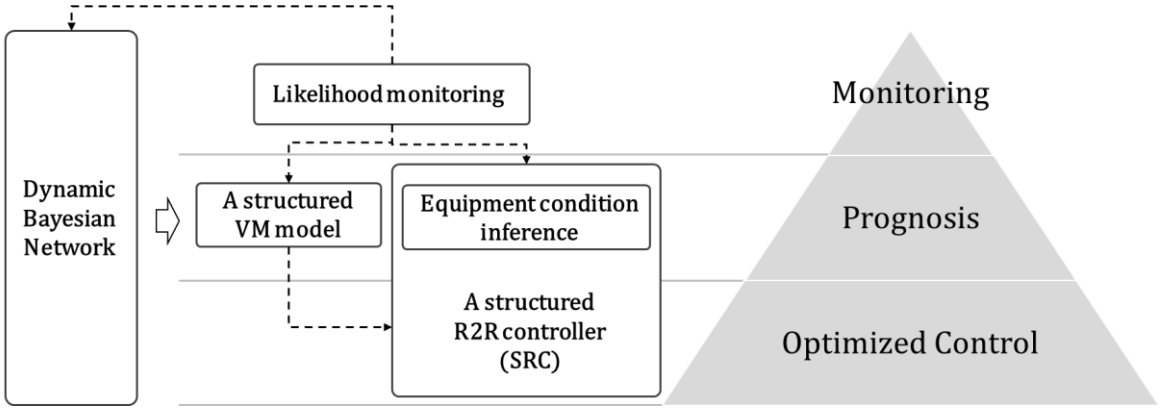


Figure 1.7 A DBN-based control system.

## 1.3 Contributions

The main contributions of this thesis can be summarized, from a top-down point of view, as follows:

*i. Integrated framework*

Instead of individual control systems, an integrated framework consolidates various function modules. With a universal model based on DBN, the maintenance tasks of different modules can be simplified. Furthermore, as all modules are essentially correlated to each other, the proposed framework enables the possibility of combining these modules. In this way, more consistent control decisions can be made.

*ii. General-purpose model*

A general-purpose model, DBN, consolidates available data related to process control. It also presents the flexibility of including existing domain knowledge instead of just data-driven information (Chen et al., 2012). With the connected graph, the relationships among variables can be easily interpreted and validated. As the most modules rely on the same data sources, data preprocessing and modeling of this general-purpose model will be just a one-time work, and it can be used to support different applications.

*iii. Applications*

A DBN-based monitoring approach can be quickly employed without complex computations. A structured VM model can be simply obtained without additional effort. As the DBN has been incorporated existing domain knowledge, the VM model can fit better the physical laws. The structured R2R controller take into account equipment information so that the control actions are optimized.

## 1.4 Thesis Overview

The remainder of the thesis is organized according to the following structure. In Chapter 2, a brief introduction about the process of semiconductor manufacturing is given, and then a literature review of three important applications (process monitoring, VM, R2R control) is presented. In Chapter 3, the theoretical background of Bayesian Networks (BNs) is introduced, including the concept of BNs and other networks, their inference, and the structure learning algorithm. The details of the proposed framework are illustrated in Chapter 4, including offline and online stages. The evaluation of the proposed method using both a real-world case study and simulated data, are presented in Chapter 5. Finally, conclusions and future work are summarized in Chapter 6.

# 2 Review of Process Control Systems in Semiconductor Manufacturing

The objective of semiconductor manufacturing is the fabrication of Integrated Circuits (IC) on wafers, and products that can be widely used in electrical and electronic devices. Semiconductor manufacturing involves various process operations, such as deposition, photolithographic, etching, implantation (see Fig. 2.1). Through these complex operations, the circuits are gradually formed in the wafers. Since the characteristics of each process are very different, the control systems are process-specific. In the rest of this manuscript, a *process* refers to a single process operation.

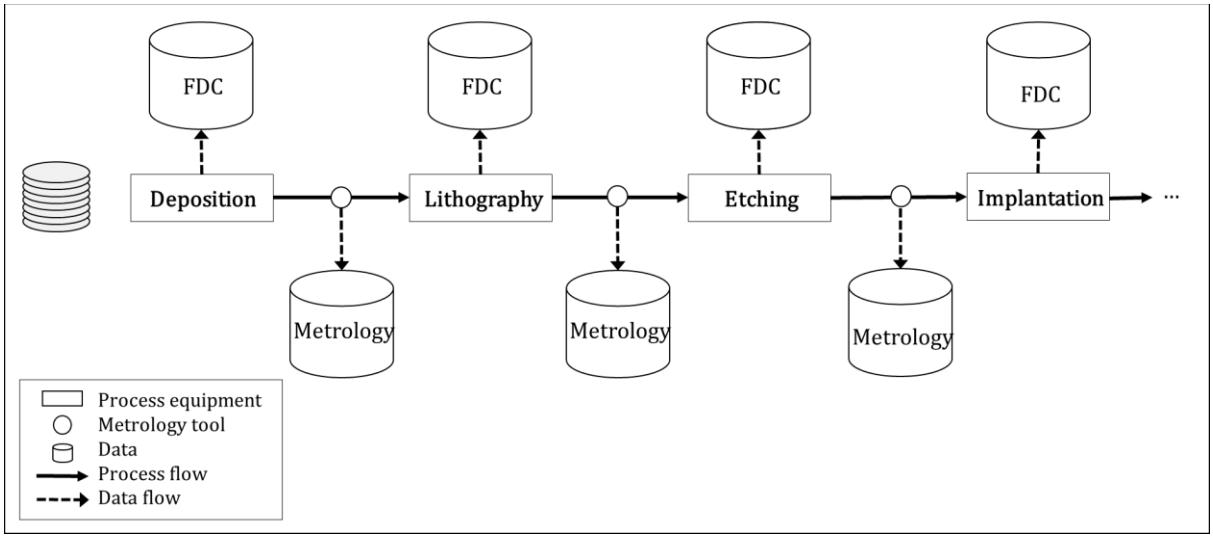


Figure 2.1 An example of process flow and data flow.

Process stability and the corresponding quality of the wafers produced are always of primary concern. Depending on the process, relevant measurement data are used for conducting quality assessment. For example, the Critical Dimension (CD) is used to evaluate the performance of the lithography process, and the depth is used for the etching process (see Fig. 2.2). Some sampled wafers are sent to metrology tools to obtain these measurements, and the set of such measurements is called *metrology data* (see Fig.2.1).

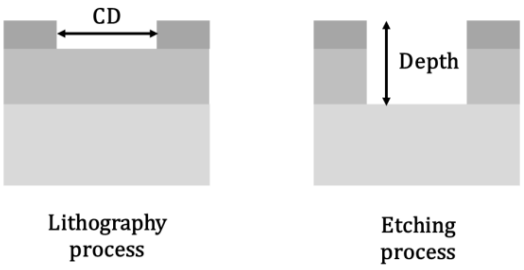


Figure 2.2 Examples of the measurements for different process.

Generally, several sensors are embedded in the process equipment, and they collect signals in real-time during wafers processing. These sensors represent several equipment parameters and

are usually called SVIDs (Status Variable IDentification). SVIDs comprise information from both physical and chemical parameters which reflect the actual state during the process, for all wafers. The data collected through these sensors are used for Fault Detection and Classification (FDC). These sensor data are simply called *FDC data*. FDC data are composed of many SVIDs collected as time series at high sampling rates (e.g., every second). Each wafer can be described by various sensors in the chamber of process equipment. Ideally, the profile of temporal data should follow the target settings of the recipe; however, the values vary due to process fluctuations, drifts, and machine aging. Data collected have great potential to be used for assessing the status of the process and inferring the consequence in the quality of wafer.

In this chapter, the state-of-the-art concerning three main applications domains is reviewed. In Section 2.1, monitoring methodologies for process fault detection and fault diagnosis are covered. The virtual metrology modeling approaches and modeling methods are addressed in Section 2.2. A detailed review of several R2R controllers is presented in Section 2.3. Finally, some current trends and research prospects are referred in Section 2.4.

## 2.1 Process Monitoring

Generally, process monitoring systems can be categorized as targeting the quality of the wafers or the state of process equipment (see Fig. 2.3). The first one usually uses metrology data as the index, while the second one considers the data collected by sensors in processing equipment, i.e., FDC data. However, metrology data also can reflect the state of equipment, and FDC data can be used to predict the quality of the wafer. Therefore, process monitoring should consider all these systems together.

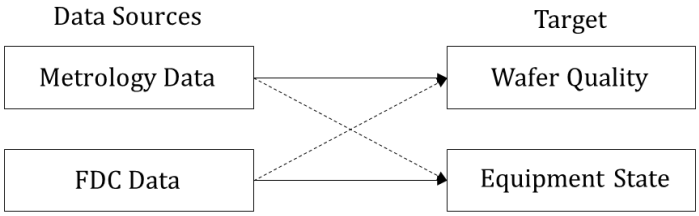


Figure 2.3 Proocess monitoring in terms of data sources and targets.

Due to the numerous sensors of advanced process equipment, the dimensionality of FDC data is usually very high. The collected FDC temporal data of $n$ wafers from a process equipment can be presented in a $n \times t \times p$ cube as shown in Fig. 2.4a, where there are $p$ sensors and those sensors collected $t$ readings during the process. The number of reading depends on the sensor resolutions. For example, if the processing time is $t$ seconds, and the sensor reading is recorded every seconds, then, $t$ readings are obtained. Fig. 2.4b shows the temporal profile of a sensor of a wafer. By computing summary statistics of the steps, those readings can be aggregated into several *indicators*. Let us assume that the data collected by each sensor can be transformed into $s$ indicators. Then, the process information of a wafer can be presented by $p \times s$ indicators as shown in Fig. 2.4c. Traditionally, SPC focus on monitoring FDC indicators which are established based on domain knowledge. More indicators can capture the state of equipment more comprehensively, but this can significantly increase the difficulty

of monitoring. In this section, a review of multivariate monitoring methods based on FDC data is presented.

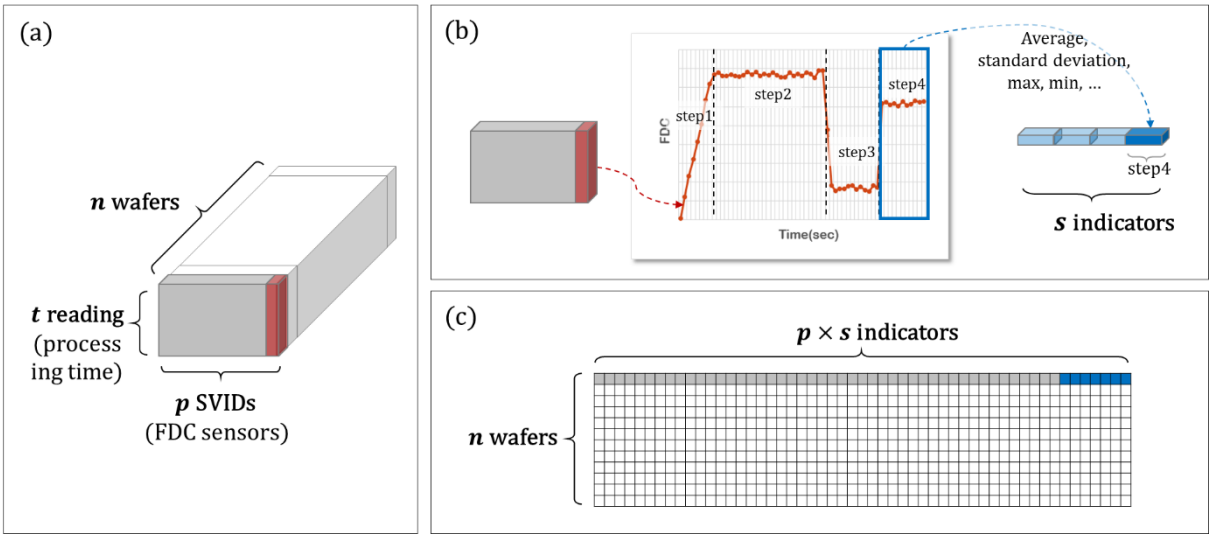


Figure 2.4 (a) FDC data cube; (b) the temporal profile of a sensor of a wafer; (c) the summarized indicators of all wafers.

The task of process monitoring can be separated into fault detection and fault diagnosis. A fault is defined as abnormal behavior during the process; the reason can be equipment aging, equipment failure, or other disturbances. The task of fault detection is to determine if a fault occurs, whereas fault diagnosis aims to identify the underlying root cause. The main challenge of fault detection is the sensitivity of monitoring multivariate observations. One of the conventional approaches is to employ Hotelling's $T^2$ (Hotelling, 1947; Bunkofske et al., 2003; Biton & Ratner, 2005). Assume that $k$ indicators follow multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, where $\mu$ is a $q \times 1$ mean vector, $\Sigma$ is a $q \times q$ covariance matrix based on historical data. The Hotelling's $T^2$ statistics is defined as $t_w^2 = (X_w - \mu)'\Sigma^{-1}(X_w - \mu)$, where $X_w$ is $q \times 1$ vector of wafer $w$. In this way, the monitoring task can be simplified by monitoring the $t^2$ statistics of each wafer. Principal Component Analysis (PCA) is another common multivariate method which has been widely studied for process monitoring, especially in the recent decades (Mastrangelo et al.,1996; Raich & Cinar, 1997). The objective of PCA is to reduce the number of variables by projecting data into lower dimension space, while these new variables keep the most original information. Assume that there are $q$ variable $X = [X_1 \quad \cdots \quad X_q]$, and the covariance $\Sigma$ is a $q \times q$ matrix. PCA aims to transform $q$ variables into $q$ new variables, through a particular orthogonal matrix $P$ such that $P'\Sigma P = L$, where $P = [P_1 \quad \cdots \quad P_q]$. The diagonal elements $l_j$ of $L$ are called eigenvalues of $\Sigma$ and $P_j$ are the eigenvectors of $\Sigma$, where $j = 1, \ldots, q$. The new variables $Z_j$ are called principal components $X$ and can be obtained using the eigenvectors $P_j$, as $Z_j = P_j'X$. The mean of $Z_j$ is zero and its variance is given by the corresponding eigenvalue $l_j$. The first principle component $Z_1$ is a linear combination of the original variables with the largest variance. The second component can be obtained by subtracting the first component from $X$ and then searching the linear combination which can maximize the remaining variance. It turns out to be the second eigenvector $\Sigma$. Finally, all $q$ components can be found through the same way. Since the first few principal components are

able to describe most of the original variance in the data, numerous monitoring chart can be reduced to monitoring only a few variables, which can be event reduced to two complementary statistics, namely the Squared Prediction Error (SPE) and the Hotelling's $T^2$ statistic of the PCA scores (Dunia & Qin, 1998; Wise & Gallagher 1996). Several approaches based on PCA have been proved their efficiency on fault detection (Wise & Gallagher, 1996; Cherry,& Qin, 2006).

Generally, the methods for fault detection aim to compress the multivariate data into a small number of features or indexes. Once a fault is detected, the next task is to diagnosis the root cause, so that the corresponding calibration can proceed. The approaches for fault diagnosis can be categorized into two types, contribution-based approaches and the reconstruction-based approaches (Qin, 2003). The former one does not need prior knowledge, while the latter one requires historical data with faults. One of the popular contribution-based approaches is the contribution plots, which provides a link between out-of-control signal on a control chart and the original variables that caused this signal (MacGregor et al., 1994; Miller et al. 1998; Goodlin et al., 2003). Another approach, for the Hotelling's $T^2$ statistic of $X$, is MYT-decomposition (Mason et al., 1995). The idea of this method is to decompose the $T^2$ statistics into different terms, including conditional terms and unconditional terms. An advantage of MYT-decomposition is to present not only the contribution of independent variables but also the relationship between two or more variables.

If the historical data of particular faults are available, the reconstruction-based approaches can be used to provide a more conclusive result. Raich and Cinar (1996) proposed a PCA-based discriminant framework. Several PCA models are built based on historical processing data with different disturbances. If a fault is detected, these PCA models will be used to compute the similarity index and so that the type of fault can be identified. Instead of only considering similarity, Chiang et al. (2000) employed Fisher Discriminant Analysis (FDA) which further takes into account the differences among the various types of fault class. The idea is to find a set of projection vectors and order them in terms of maximizing the differences between fault classes while minimizing the differences within each fault class.

More advanced methods for fault detection and diagnosis that incorporate machine learning algorithms have been proposed as well (Qin, 2012). To handle nonlinearity data, He and Wang (2007) consider an approach based on K-Nearest Neighbor (KNN), and Botre et al. (2016) proposed kernel PLS. To improve the efficiency of monitoring, Hong et al. (2012) employed a Modular Neural Network (MNN) for fault diagnosis and used Dempster-Shafer (D-S) theory for fault diagnosis. Yang and Lee (2012) considered Bayesian Networks for fault diagnosis based on several discrete indexes. The continuous data collected by some sensors were transformed into discrete variables with some state, such as *normal*, *warning*, or *error*. The states were determined based on process specifications or engineering tolerance. The output of some binary sensors, such as *on* and *off*, can be used without preprocessing. The quality measurements of wafer, i.e., metrology data, are labeled as either *good* or *bad*, based on process specifications. Through a Chemical Vapor Deposition (CVD) case study, the BN approach has demonstrated its capability for identifying the problematic sensors for a bad wafer.

More researchers investigate BN for fault diagnosis can be found in the literature (Lerner et al., 2000; Cai et al., 2017).

## 2.2 Virtual Metrology

In semiconductor manufacturing, process stability and the corresponding quality of the wafers produced are always of primary concern. Given the current limitations in the capacity of metrology tools and the reduced product life cycle of wafers, only a few of them can be sampled and sent for inspection. To circumvent the technological limitations of metrology tools and enhance the efficiency of process monitoring, Virtual Metrology (VM) systems have been developed in recent decades. The VM systems can be considered as an effective solution to employ intelligent software to decrease the dependency in expensive and time-consuming hardware (Chen et al., 2005; Cheng & Cheng, 2005; Hung et al., 2007; Khan et al., 2007; Kang et al., 2009; Zeng & Spanos, 2009; Pampuri et al., 2011; Susto et al., 2013; Wan et al., 2014). Several VM models have been investigated in literature for various process, including Chemical Vapor Deposition (CVD) (Hung et al., 2007; Olson & Moyne, 2010; Susto et al., 2011; Ferreira et al., 2011 ; Besnard et al., 2012; Susto & Beghi, 2013), etching (Kang et al., 2009; Zeng & Spanos, 2009; Lynn et al., 2010), lithography (Kang et al., 2011), Chemical-Mechanical Polishing (CMP) (Jebri et al., 2017).

The VM models aim at predicting metrology measurements of the wafers based on FDC data. An offline VM modeling procedure starts by aligning historical FDC data with the sampled wafers with metrology data. Then, identifying the relationship between FDC information and wafer quality, usually through a variety of machine learning models. Finally, a VM model can be used for online prediction as shown in Fig. 2.5.

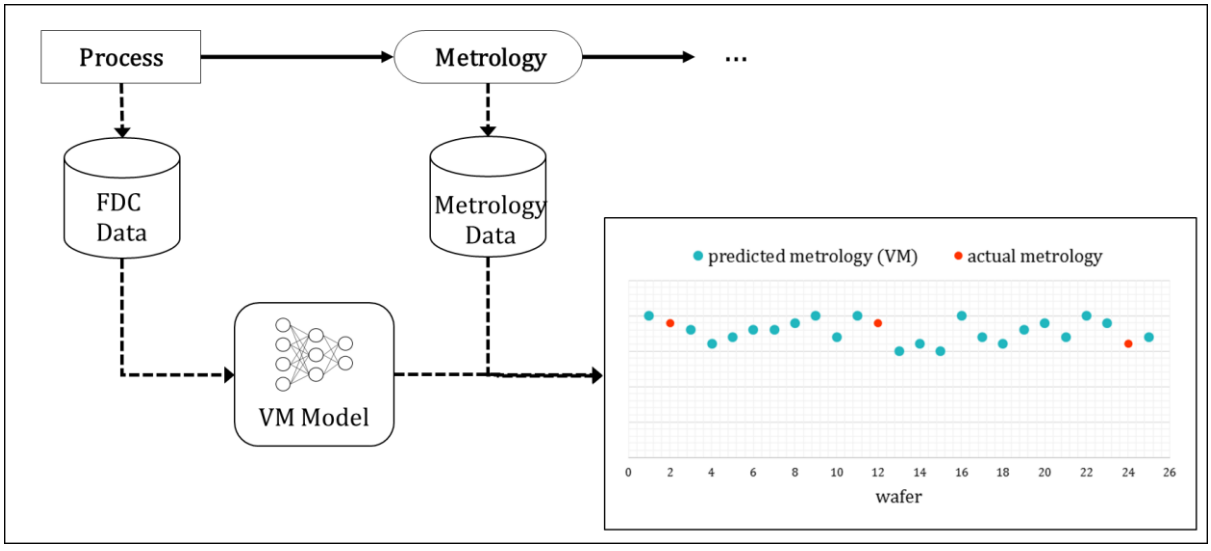


Figure 2.5 A VM model for online prediction.

As described in Section 1.2.2, many data mining or machine learning methods have been studied for VM modelling, such as MLR, PLS, LASSO, KNN, RT, etc. In the work presented in references (Rendall & Reis, 2017; Rendall et al., 2018), a categorization into four groups is proposed: variable selection methods (e.g. stepwise regression, genetic algorithms, etc.),

penalized regression (e.g. LASSO, Ridge Regression, Elastic Net), latent variable methods (e.g. PLS, PCR) and tree-based methods (e.g. random forests, gradient boosting). Several comparison studies are reported in the literature, where these and other methods are assessed and discussed. Ferreira et al. (2011) shows that PLS can provide better predictive capabilities when compared to tree-based ensemble methods for oxide thickness prediction in Plasma Enhanced Chemical Vapor Deposition (PECVD). A comparison between five methods (MLR, KNN, NN, RT, SVR), with prior variable selection and variable extraction procedures was carried out showing that MLR and SVR perform well for etching process (Kang et al., 2009). Wan et al. (2014) consider four methods, including MLR, LASSO, GPR, NN, and evaluated these methods with a CVD dataset, observing that GPR was the one that performed better.

The patterns captured by VM models may change over time due to various reasons, such as equipment aging and unexpected disturbances. The reliability of the VM models should be taken into account as well. Cheng et al. (2008) introduced Reliance Index (RI) for evaluating the quality of VM. They firstly standardized actual measurements and denoted as $Z_{y_i} \sim \mathcal{N}\left(\mu_{Z_{y_i}}, \sigma_{Z_{y_i}}\right)$. Then, the standardized data are used to establish a VM model based on Neural-Network (NN), called conjecture model. The same data are used to construct another Multiple Linear Regression (MLR) model called reference model. When $n \to \infty, Z_{y_i} = Z_{y_i(NN)} = Z_{y_i(MLR)}$, where $Z_{y_i(NN)}$ is the standardized predicted measurement by NN and $Z_{y_i(MLR)}$ is the standardized predicted measurement by MLR. The RI is defined as the overlap area of two distributions. For online reliance evaluation, if the actual measurement is available, the RI will be calculated by $Z_{y_i}$ and $Z_{y_i(NN)}$. When the actual measurement is not available, $Z_{y_i}$ will be replaced by $Z_{y_i(MLR)}$. The acceptance threshold $RI_T$ can be computed based on a defined error limit. To access the similarity between historical data and new data, Cheng et al. (2008) also proposed a Global Similarity Index (GSI) to help the RI gauge the reliance level. The GSI is calculated based on Mahalanobis distance, i.e., small GSI indicates that the virtual measurement of new data is relatively accurate. If GSI is large, Individual Similarity Indexes (ISIs) can be used to identify the root cause of this dissimilarity. Empirically, the threshold $GSI_T$ is set to be three times its standard deviation. Other approaches for reliance evaluation or modeling updating mechanisms have been investigated as well (Khan et al., 2008; Kang & Kang, 2017).

## 2.3 Run-to-Run Control

Nowadays, one of the toughest challenges for semiconductor manufacturing is to maintain high product yield with decreasing process-variation tolerance. However, many processes exhibit a steady drift in equipment performance. Those drifts can be different depending on the characteristics of the process. For some process, the drift can be caused by the build-up of material on the chamber wall of the equipment or gradual wear of components. For example, the deposition rate in a metal sputtering process is highly correlated to the life of components of the equipment. If the component has been used for a long time, the measurements of thickness may show a steady decreasing. Conventional Statistical Process Control (SPC) system can detect such problems, but it is not capable of compensating these aberrations. These

problems can only be fixed by engineers. But these kinds of approaches can be time-consuming and reduce Overall Equipment Effectiveness (OEE), and finally lead to high production cost. However, some process output can shift back to the target by a slight adjustment of process parameters. For example, the decreasing of thickness measurement is caused by a declining sputter deposition rate. This problem can be fixed by simply adjusting the deposition time; in this condition, the gap between measurements and the desire target can be reduced while maintaining high throughput.

### 2.3.1 R2R Controller Basics

Run-to-Run (R2R) control systems are designed to conduct a process tuning approach. A variety of controllers have been introduced and applied in different processes (Butler & Stefani, 1994; Boning et al., 1996; Smith et al., 1998; Castillo & Hurwitz, 1997). A R2R controller can manipulate some recipe settings, i.e., process parameters, for each run, with the aim to minimize the impacts of process drift, shift, and variability. Note that each run can indicate different granularity, such as wafer-to-wafer, lot-to-lot, that depends on target performance or machine capability. In general, a R2R controller consists of a mechanism that encompasses three fundamental modules: the R2R model; the filtering model; and the controller. These fundamental modules and their relationship are depicted in Figure 2.6 (Castillo & Hurwitz, 1997). The R2R model, which defines the mapping between controllable variables and wafer metrology (the target response), is usually built after conducting a series of trials planned and executed according to the principles of statistical Design of Experiments (DOE), and it is also called Response Surface Model (RSM). The filter model generates the adjusted coefficients based on historical settings and results, while the controller computes the control actions using the updated coefficients obtained by the filter model for the controllable variables. The filter most often adopted in practice is the Exponentially Weighted Moving Average (EWMA) smoothing filter. It can be proved that R2R equipped with an EWMA filter can provide the minimum variance control under the assumption that the noise follows a first-order Integrated Moving Average process, IMA(1,1) (Box, 1974).

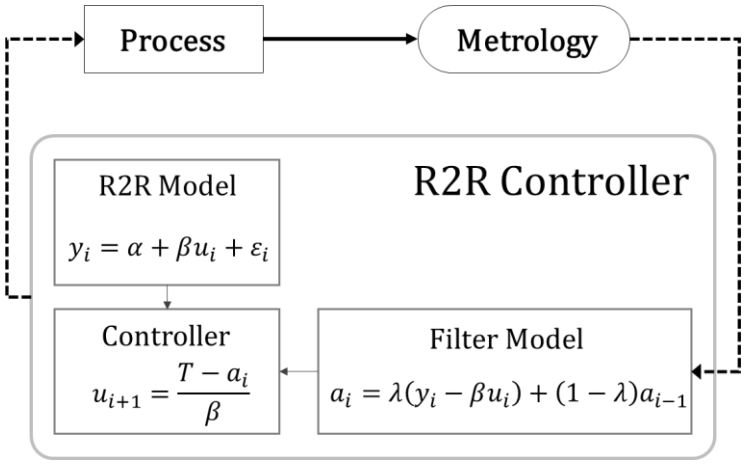


Figure 2.6 The framework of a Run-to-Run controller.

Considering a Single-Input-Single-Output (SISO) system, i.e, only one controllable variable and one process output. Let the R2R model be defined as

$$y_i = \alpha + \beta u_i + \varepsilon_i, \tag{2.1}$$

where $y_i$ denotes the output at $i^{\text{th}}$ run, $u_i$ is the corresponding controllable (or manipulated) variable, $\alpha$ and $\beta$ are model parameters and $\varepsilon_i \sim i.i.d.N(0, \sigma^2)$ is a natural white noise process disturbance. Let us also assume that intercept $\alpha$ is time-varying parameter and denote $a_i$ as the estimate of $\alpha$ at run $i$ which can be obtained through the EWMA filter as follows:

$$a_i = \lambda(y_i - \beta u_i) + (1 - \lambda)a_{i-1}, \tag{2.2}$$

where the weight $\lambda$ is between 0 and 1. Thus, the estimated process output at the $(i+1)^{\text{th}}$ run is computed as

$$\hat{y}_{i+1} = a_i + \beta u_{i+1}. \tag{2.3}$$

Therefore, in order to keep the process output at the target level ($y_{target}$), the value of the controllable variable for the $(i+1)^{\text{th}}$ run should be given by,

$$u_{i+1} = \frac{y_{target} - a_i}{\beta}. \tag{2.4}$$

### 2.3.2 Controller for Complex Processes

The basic R2R controller has been shown its capability of reducing process variation. However, several studies have shown that such an approach is ineffective and does not meet the industrial needs (Castillo & Hurwitz, 1997; Chen & Guo, 2001; Wang et al., 2015), and propose including the effect of additional stochastic terms. Butter and Stefani (1994) proposed a Predictor-Corrector Control (PCC) scheme, with the aim of estimating the process trend through a double exponential filter,

$$a_i = \lambda_1(y_i - \beta u_i) + (1 - \lambda_1)a_{i-1},$$

$$p_i = \lambda_2(y_i - \beta u_i - a_{i-1}) + (1 - \lambda_2)p_{i-1}, \tag{2.5}$$

where $\lambda_1$ and $\lambda_2$ are the weights for the first and second EWMA filters, respectively, and $p_i$ is used to compensate for the error incurred by $a_i$, and the controllable value is then given by,

$$u_{i+1} = \frac{T - (a_i + p_i)}{\beta}. \tag{2.6}$$

In this case, both the intercept and trend terms will be updated and fed back to the controller. Chen and Guo (2001) further extended this approach to an age-based double EWMA scheme that takes into account the process age.

Generally, the stability of processes is affected by various unknown disturbances. To determine the appropriate control decision based on different disturbances, Sachs and Hu (1995) proposed an approach including Gradual Mode (GM) and Rapid Mode (RM) for dealing with

gradual drift and rapid step change, respectively. They firstly adopt Bayesian statistics in Run-to-Run control to estimate the magnitude and location of step change. The proposed controller has three major components, a diagnosis mode that decides whether the process is behaving in accordance with the current process model. A gradual model that gradually modifies the process model when a process undergoes a slow drift, such as that decreasing deposition rate due to build-up material on the chamber wall. And a rapid mode that quickly updates the process model when a process undergoes a rapid shift, such as maintenance. Another controller incorporated with Bayesian approach is B-EWMA (Wang & He, 2007). By matching the pattern of the pre-change window and the pattern of the post-change window, various disturbances can be detected and classified.

In most cases, the process output can be controlled by more than one controllable variable. Therefore, Multiple-Input–Multiple-Output (MIMO) controllers are more efficient than the SISO controllers (Castillo & Yeh., 1998; Tseng et al., 2002; Chen & Wang, 2007). Castillo proposed the Optimizing Adaptive Quality Controller (OAQC) which can handle MIMO system, by solving the following multidimensional optimization problem,

$$\min \quad (\hat{y}_i - y_{target})'\Gamma_1(\hat{y}_i - y_{target}) + (u_i - u_{i-1})'\Gamma_2(u_i - u_{i-1})$$

$$\text{s.t.} \quad y_{min} \le \hat{y}_i \le y_{max}$$

$$u_{min} \le u_i \le u_{max}$$

where $\Gamma_1$ is a diagonal matrix representing the relative importance of each process output, and $\Gamma_2$ is a diagonal matrix indicating the costs associated with changes in the control elements. This objective function includes two common control laws: minimize the deviation of metrology from the target and penalize the change of regulating variable. Once the process is at optimal operation, the control law of the OAQC attempts to maintain the process at this condition. Most controllers need to specify the R2R model which is usually obtained by DOE, while OAQC is capable of estimating parameters by the recursive method. To model complex underlying process variation, OAQC is also capable of dealing with both linear and non-linear processes through the incorporation of a quadratic term in the controllable variables (Castillo & Yeh, 1998).

Process variability may be caused not only by offset drift, but also by parameters drift. Palmer et al. (1996) considered a state-space model – Kalman Filter Model (KFM) to reduce the variability of resist coating in the lithography process. The idea of KFM control scheme is to recursively update the parameters so that the desire control setting can be determined based on the most updated state. The corresponding filtering model and R2R model can be written in the following standard form.

$$\theta_{i+1} = \theta_i + w_i$$

$$y_{i+1} = u_i\theta_i + v_i$$

Where $\theta_i$ is the parameter vector, $y_{i+1}$ is the process output vector, $v_i$ is measurement noise, $w_i$ is the parameter noise, and $u_i$ is the controllable input vector. In this study, assuming a

simple static input-output relationship has been defined. In order to regulate $i+1$ wafer, the estimation of $\hat{\theta}_{i+1}$ can be derived by

$$\hat{\theta}_{i+1} = \hat{\theta}_i + \mathrm{K}_i(y_i - U_i\hat{\theta}_i),$$

where $\mathrm{K}_t$ is Kalman filter gain which can be computed by the new observation and the updated covariance matrix. With the estimated parameter vector $\hat{\theta}_{i+1}$, the control setting can be determined.

Most R2R control algorithms were based on the assumption that only a single product is manufactured in a process equipment. However, most production lines are high-mix assembly line where the products with same specification can be manufactured by different parallel equipment. And an equipment may produce several product types. Therefore, many researchers also investigated more sophisticated controllers to handle such high-mix production lines (Firth et al., 2006; Wang et al., 2009; Tan et al., 2015), the efficient of these controllers have been validated through simulated data or industrial data.

### 2.3.3 Controllers based on Machine Learning

Some controllers also incorporate with machine learning approaches, such as Neural Network (NN), decision trees. Smith and Boning (1997) proposed a self-tuning EWMA controller which using NN to train a state-weight mapping so that the optimal weight can be obtained. Hankinson (1997) introduced a Knowledge-based Interactive Run-to-run Control (KIRC) that uses leaves from a classification decision tree to suggest control actions for process improvement. Each branch indicates the control action and each leaf suggests the expected output based on the sequential control setting. The starting operating point is chosen from the largest leaf in the decision tree where all outputs are inside the target range. As gradual drifts or sudden changes occur in the process, the output values may leave the region of the target leaf. The decision tree then searches for a neighboring leaf that matches the current output classification. By comparing the control variable ranges of the leaf matching the new process state with the original target leaf, a control action is computed to move the process back in the target output region.

Comparisons for a variety of controllers have been made. Their effectiveness is examined either using industrial data or simulated data. These comparisons primarily evaluate different controllers based on their capabilities of handling different types of process disturbances. Ning et al. (1996) compared the Gradual Mode (GM), Time-based Gradual Mode (GMt), KIRC, OAQC with a simulated CMP dataset. Considering a linear process with linear drift, and a non-linear process with linear drift. The result shows that GMt and KIRC perform well only in the linear process while OAQC perform well in both the linear and the non-linear process.

### 2.3.4 Controllers based on Virtual Metrology Modeling

Although many sophisticated controllers have been proposed, metrology data still serve as the main input source for controllers in the semiconductor industry. However, given the current limitations in the capacity of metrology tools and the reduced product life cycle of wafers, only a few of them can be sampled and sent for inspection. The accuracy of controllers might not be

enough due to the disconnected physical measurements and metrology delay (Wu et al., 2008). In order to enrich the sparse metrology data, in the past decades Virtual Metrology (VM) has been widely studied in the literature (Chen et al., 2005; Cheng & Cheng, 2005; Kang et al., 2009; Susto et al., 2013; Wan et al., 2014).

Several studies have shown that R2R control systems can benefit from the introduction of a VM model (Khan et al., 2007; Susto et al., 2012; Fan & Chang, 2013), the concept of R2R control system incorporating a VM model is illustrated in Fig. 2.7. A FDC data analysis system collects the signal data and calculates some statistics (such as mean, standard deviation, etc.), which are considered as features of each wafer. An initial VM model should be built in advance based on historical data, and then used together with the R2R control, which will work based on both the actual output $y_i$ and the predicted output $\hat{y}_i$. In parallel, the VM model will be recursively updated when new actual metrology data are available.

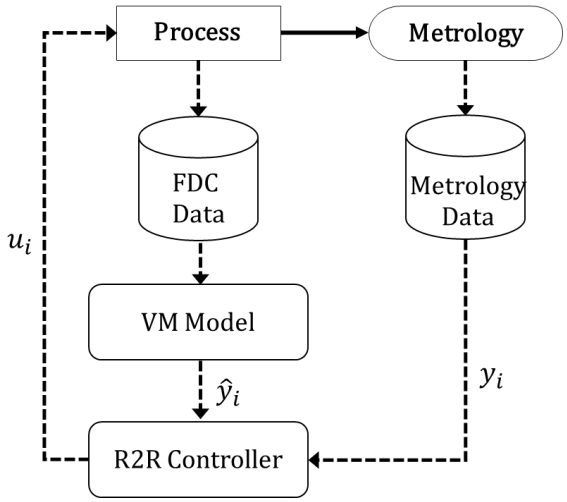


Figure 2.7 A VM model integrated Run-to-Run controller.

As the VM model is based on the equipment data, the real-time equipment conditions can be taken into account in the form of the predicted measurements. Therefore, by leveraging the extra information brought by VM, the controller can be improved through both the physical measurements and predicted measurements.

A factory-wide VM and R2R control framework has been proposed in the literature (Khan et al., 2007). Considering several consecutive processes, the predicted metrology of the former process can be used for both feedback control and feed-forward control (Fig. 2.8 green line). The physical metrology of the former process not only can be used to update the VM model (Fig. 2.8 blue line) but also can enter as an input variable of another VM model of the next process (Fig. 2.8 orange line).

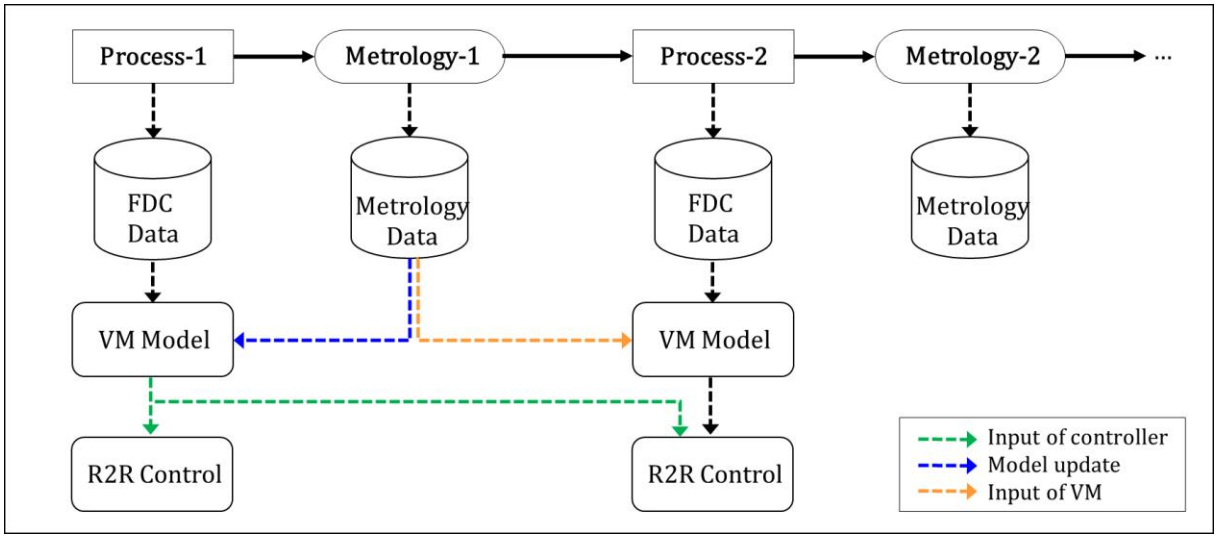


Figure 2.8 An example of factory-wide VM and R2R controller framework.

A complete VM and control framework should also incorporate several associated mechanisms. For instance, the control action should be decided based on the quality of VM output. Various approaches have been proposed to monitor and adjust the control action (Khan et al., 2007; Susto et al., 2012; Kao et al., 2013). Khan et al. (2007) consider VM in R2R control and revise equation 2.2. The filter model would be expressed as either $a_i = \lambda_{\mathrm{p}}(y_i - \beta u_i) + (1 - \lambda_{\mathrm{p}})a_{i-1}$ when the physical measurement is available or $a_i = \lambda_{\mathrm{v}}(\hat{y}_i - \beta u_i) + (1 - \lambda_{\mathrm{v}})a_{i-1}$ when only virtual measurement is available. Generally, $\lambda_{\mathrm{p}}$ is larger than $\lambda_{\mathrm{v}}$, depends on the quality of VM. Kao et al. (2013) further utilized both GSI and RI in VM (Cheng et al., 2008) and the R2R control framework. Considering the EWMA control scheme, the weight of the filter model should be set dynamically based on different scenarios, as shown in Table 2.1.

Table 2.1 The setting of $\lambda_v$.

| Scenario | Conditions | $\lambda_{\mathrm{v}}$ |
|---|---|---|
| 1 | $RI < RI_T$ or $GSI > GSI_T$ | 0 |
| 2 | $RI \geq RI_T$ , $GSI \leq GSI_T$ and the wafer belongs to the *first lot*. | $RI \times \lambda_{\mathrm{p}}$ |
| 3 | $RI \geq RI_T$ , $GSI \leq GSI_T$ and the wafer doesn't belong to the *first lot*. | $(1 - RI) \times \lambda_{\mathrm{p}}$ |

The rules are that $\lambda_{\mathrm{v}}$ should depend on the quality of VM, and $\lambda_{\mathrm{v}} < \lambda_{\mathrm{p}}$. The first scenario indicates that the VM result cannot be adopted if either RI is low or GSI is large, and $\lambda_{\mathrm{v}}$ should be zero. The VM result can be employed when RI is higher than threshold and GSI is small, which leads to the second and the third scenario. These two scenarios can be distinguished by the maintenance schedule. Kao et al. defined a lot after a maintenance event as the *first lot*, and they suggested to give a higher $\lambda_{\mathrm{v}}$ after maintenance, i.e., $\lambda_{\mathrm{v}} = RI \times \lambda_{\mathrm{p}}$. Because the *first lot* is relatively unstable and the higher regulation weight can accelerate the adjustment process. The third scenario describes the normal cases, where $\lambda_{\mathrm{v}}$ is set to be $(1 - RI) \times \lambda_{\mathrm{p}}$ when the process is relatively stable. With such a guarding mechanism, the R2R controller incorporates a VM model that should be able to work better under different circumstances.

## 2.4 Research Prospects

In this chapter, the state-of-the-art of some process control applications were presented, including process monitoring, virtual metrology, and R2R control. Although these applications have been investigated deeply for complex processes, they were constructed and operated individually, which require a substantial o resources for modeling and maintenance. However, these applications are essentially under the same scope of process control. Therefore, the objective of this thesis is to develop a framework based on a DBN which can consolidate multiple modules for these applications. Under this framework, these function modules can either work independently or together. As all modules are constructed based on the same model, the integration will be simpler. Furthermore, the framework should not only provide accurate predictions and suitable control actions but also present an interpretable and rational structure to accommodate fundamental restrictions and relationships. These aspects have been missing in the solutions proposed hitherto. Thus, a physics-informed framework is proposed in this thesis, which can employ the best of both information sources: data and the Subject Matter Expert (SME) knowledge.

Some challenges of existing solutions for each module should be discussed as well. Reviewing current monitoring approaches, one can verify that they usually involve a set of methods for fault detection, and require extensive computing procedures to identify the location of a fault (fault diagnosis). The proposed monitoring mechanism based on DBN is able to cover both fault detection and fault diagnosis. Since this monitoring mechanism has been embedded in the DBN, it does not require additional learning procedure. With an interpretable structure, the analysis can be more straightforward.

Analyzing the existing literature, it is possible to verify that the procedures to derive a VM model essentially consist of comparing different methods and choosing the method with lowest prediction error. This pure data-driven approach is highly case dependent and, most importantly, overlooks existing knowledge about the process. Therefore, the proposed VM model based on DBN not only can provide accurate predictions but also to disclose relevant causal-effect associations among process variables.

Many R2R controllers have been proposed in the literature, which all consider metrology data as input, either actual measurements or predicted measurements. And the relationships have been established are either between controllable variables and metrology variables, or between equipment conditions and metrology variables. However, control decisions may change several equipment conditions and indirectly have an impact on the process output, which has not been explicitly studied before. Therefore, the main goal of the proposed R2R controller is to consider more hidden dependencies for control decision making.

# 3 Bayesian Networks

In this thesis, the Dynamic Bayesian Network (DBN) is employed as a foundation of the proposed framework. The objective of this chapter is to provide the theoretical background of DBN. The DBN is an extension of Bayesian Networks (BNs) (Pearl, 1988), where BNs are probabilistic models expressing the conditional dependencies of a set of variables through a Directed Acyclic Graph (DAG). This chapter starts with the general terminologies and properties of DAGs. BNs and the extended types of networks are presented in Section 3.2. In Section 3.3, different methods for network inference are introduced. The details of structure learning are explained in Section 3.4. The applications of BNs and DBNs are discussed in Section 3.5. Finally, a chapter summary and the core concept of this thesis is explained in Section 3.6.

## 3.1 Directed Acyclic Graph

Since Bayesian Network is a type of DAG, the general terms of DAG will be explained first. In graph theory, a directed graph is denoted as $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, where

- $\mathbb{V} = \{V_1, V_2, \dots, V_q\}$ is a set of vertices, also known as a set of nodes.
- $\mathbb{E}$ is a set of directed edges or arcs, where each edge links two nodes in $\mathbb{V}$.

A directed edge $e \in \mathbb{E}$ can be presented in the form of a pair of nodes, $e = (V_i, V_j)$, denoted as $parent$ and $child$, respectively, where $V_i$ is the head of the edge and $V_j$ is the tail of the edge. A *path* in a graph is defined as a sequence of edges which joins a set of nodes. Assuming there is a path from $V_i$ to $V_j$. $V_i$ is called the *ancestor* of $V_j$. $V_j$ is called the *descendant* of $V_i$. A DAG guarantees that no node can be its own ancestor and descendent at the same time.

Reachability in graph theory refers to the ability to go from one node to another node through a path. Assuming a pair of nodes $(V_i, V_j)$, $V_i$ can reach $V_j$ if there exists a path that starts with $V_j$ and ends with $V_2$, denoted by $V_i \prec V_j$. Let $\mathcal{R}$ be the set of relations on $\mathbb{V}$, defined as $\mathcal{R} = \{V_i \prec V_j, \text{ where } (V_i, V_j) \in \mathbb{V} \times \mathbb{V}\}$, $\mathcal{R}$ indicates the reachability relation of $\mathcal{G}$.

Topological sorting of a graph is a process to find a permutation $L_{ts}$ of $\mathbb{V}$ according to the precedence relation, $\mathcal{R}$. For each pair of nodes $(V_i, V_j) \in \mathbb{V} \times \mathbb{V}$, if $V_i \prec V_j \in \mathcal{R}$, then $V_i$ must precede $V_j$ in $L_{ts}$ (Knuth, 1997), i.e., $V_i$ is the *ancestor* of $V_j$. Any DAG has at least one permutation $L_{ts}$ by topological sorting. The topological sorting problem is essentially equivalent to arranging the nodes of a directed graph into a straight line, so that for any node $V \in \mathbb{V}$, the ancestors of $V$ must be in front of $V$ in $L_{ts}$.

In this thesis, one of the primary tasks is to model the consequences of control actions, including the impact on process variables and eventually the indirect impact on the metrology. With the definition of the reachability relation $\mathcal{R}$, these connected effects can be expressed in a simpler form. More details of using these properties in the proposed approach will be introduced in Chapter 4.

## 3.2 Types of Network

Bayesian Networks (BNs) are a type of probabilistic graphical model that represents a set of variables and their relationships. To understand the concept of BNs, the introduction usually starts with discrete variables. Those networks with continuous variables can be considered as a type of BNs. In this section, the idea of BNs with discrete variables is firstly presented, followed by the extensions of the basic BN are introduced as well.

### 3.2.1 Bayesian Network (BN)

Considering a set of random variables $\mathrm{X} = \{X_1, X_2, \dots, X_q\}$, a Bayesian Network is a representation of the dependencies existing between these variables in a graph form (Pearl, 1988). The graph, denoted as $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, is composed by the set $\mathbb{V} = \{V_1, V_2, \dots, V_q\}$ of nodes (vertices). Each $V_k \in \mathbb{V}$, $k = 1, \dots, q$, indicates one of the random variables in $\mathcal{D}$. A random variable $X_k$ is also referred to as the node $V_k$ in graph $\mathcal{G}$. $\mathbb{E}$ is the set of directed edges and each edge $e \in \mathbb{E}$ indicates the link between two nodes. The absence of an edge implies the existence of conditional independence between the corresponding variables.

Each node indicates a random variable and the edges describe the cause-effect relationships existing between variables as an asymmetric dependency. An example is shown in Fig. 3.1, where $X_1$ is a parent node of $X_3$ and $X_3$ is a child node of $X_1$. This dependency can be described as: $X_1$ causes $X_3$ while $X_3$ cannot cause $X_1$.

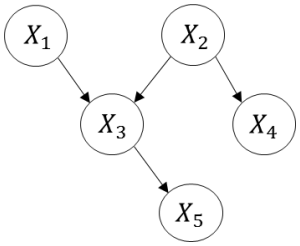


Figure 3.1 A simple illustration of a Bayesian Network.

A Bayesian network satisfies the Markov condition: each node is conditionally independent of its non-descendants given its parents. With the Markov condition, the joint probability can be expressed as a product form: $P(X_1, X_2, \dots, X_q) = \prod_{k=1}^{q} P(X_k|\mathbb{V}_{pa(X_k)})$, where $\mathbb{V}_{pa(X_k)}$ is the set of parent nodes of $X_k$ and $P(X_k|\mathbb{V}_{pa(k)})$ is the conditional probability of $X_k$ given $\mathbb{V}_{pa(X_k)}$. The probability distribution of a variable $X_k$ is a local distribution. If $X_k$ is a root node, i.e., without any parents, the local distribution of $X_k$ is unconditional. An example of a BN and its local distributions is shown in Fig. 3.2. Assuming all the variables are binary (*true* or *false*), the local distribution of each node can be expressed by the Conditional Probability Distribution (CPD) and the join probability is $P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_2)P(X_5|X_3)$.

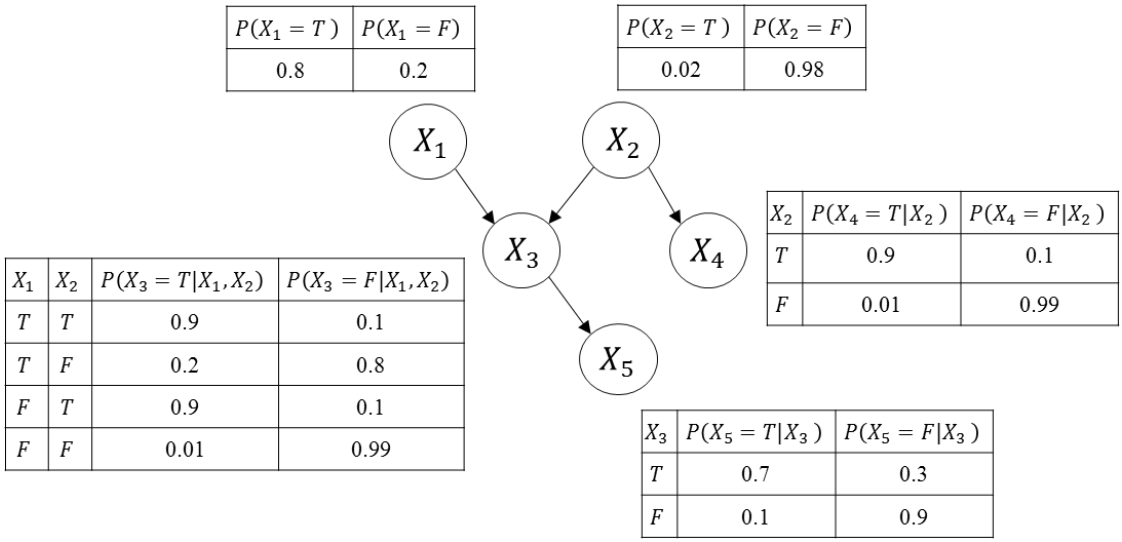


Figure 3.2 An example of a Bayesian Network and the probability table of the local distribution of each node.

### 3.2.2 Gaussian Bayesian Network (GBN)

Gaussian Bayesian Network (GBN) is a special case of BNs (Heckerman & Geiger, 1995). Each variable of a GBN follows the Gaussian distribution and is linearly related to its parents. Assuming that there are $q$ random variables in GBN $\mathcal{G} = (\mathbb{V}, \mathbb{E})$. The joint probability can be expressed as a multivariate Gaussian distribution, $\mathcal{N}(\mu, \Sigma)$, with mean vector $\mu \in \mathbb{R}^q$ and covariance $\Sigma \in \mathbb{R}^{q \times q}$. Let $\mathbb{V}_{q_1}$ be one subset of $\mathbb{V}$ and $\mathbb{V}_{q_2}$ be another subset of $\mathbb{V}$, $q = q_1 + q_2$, $\mu$ and $\Sigma$ can be rewritten as

$$\mu = \begin{pmatrix} \mu_{q_1} \\ \mu_{q_2} \end{pmatrix} \text{ and } \Sigma = \begin{pmatrix} \Sigma_{q_1,q_1} & \Sigma_{q_1,q_2} \\ \Sigma_{q_2,q_1} & \Sigma_{q_2,q_2} \end{pmatrix}.$$

The conditional distribution of $\mathbb{V}_{q_1}$ given $\mathbb{V}_{q_2}$, is expressed as $\mathbb{V}_{q_1}|\mathbb{V}_{q_2} = V_{q_2} \sim \mathcal{N}(\mu_{q_1|q_2}, \Sigma_{q_1|q_2})$, where $\mu_{q_1|q_2} = \mu_{q_1} + \Sigma_{q_1,q_2}{\Sigma_{q_2,q_2}}^{-1}(V_{q_2} - \mu_{q_2})$, and $\Sigma_{q_1|q_2} = \Sigma_{q_1,q_1} - \Sigma_{q_1,q_2}{\Sigma_{q_2,q_2}}^{-1}\Sigma_{q_2,q_1}$. Note that $\Sigma_{q_1|q_2}$ can be obtained by referring to the Schur complement. Therefore, for each variable in $\mathcal{G}$, the local distribution given its parents can be expressed as the form of conditional probability (see Fig. 3.3).

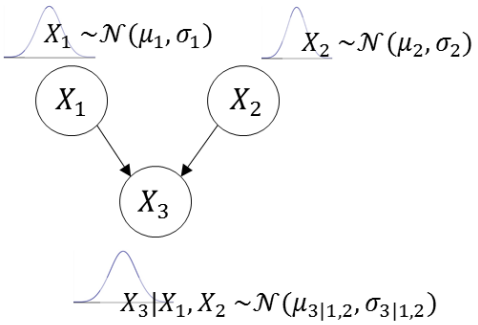


Figure 3.3 An example of a Gaussian Bayesian Network and the local distribution of each node.

### 3.2.3 Conditional Linear Gaussian Bayesian Network (CLGBN)

Conditional Linear Gaussian Bayesian Network (CLGBN) is a hybrid BN which consists of discrete and continuous variables, where the continuous ones cannot be the parents of the discrete ones (Lauritzen & Wermuth, 1989). The local distribution of the discrete variables will be CTPs and the local distribution of the continuous variables $X_k$ given its parents $\mathbb{V}_{pa(X_k)} = \mathbb{V}_{pa(X_k),D} \cup \mathbb{V}_{pa(X_k),C}$ is defined as a conditional Gaussian distribution,

$$f(X_{\mathrm{k}}|\mathbb{V}_{pa(X_k)}) = \mathcal{N}\left(\alpha(\mathbb{V}_{pa(X_k),D}) + \beta(\mathbb{V}_{pa(X_k),D})^T \mathbb{V}_{pa(X_k),C}, \sigma^2(\mathbb{V}_{pa(X_k),D})\right),$$

where $\mathbb{V}_{pa(X_k),C}$ is the set of continuous parents, $\mathbb{V}_{pa(X_k),D}$ is the set of discrete parents, and $\alpha$ and $\beta$ are the coefficients of a linear regression model of $X_{\mathrm{k}}$ given $\mathbb{V}_{pa(X_k),C}$. This model can be different depending on $\mathbb{V}_{pa(X_k),D}$.

### 3.2.4 Dynamic Bayesian Network (DBN)

Dynamic Bayesian Network (DBN) is an extension of the static Bayesian Networks which can further express the temporal dependencies (Friedman et al., 1998; Murphy, 2002). Assuming a set of variables $\mathrm{X} = \{X_1, X_2, \dots, X_q\}$ from a stochastic process, DBNs provide a factored representation of the joint probability distribution of $\{X_1^{(1)}, \dots, X_1^{(\tau)}, \dots, X_q^{(1)}, \dots, X_q^{(\tau)}\}$ on a finite time interval $[1, \tau]$, defined as $p(X_1^{(1)}, \dots, X_1^{(\tau)}, \dots, X_q^{(1)}, \dots, X_q^{(\tau)}) = \prod_{i=1}^{q} \prod_{t=1}^{\tau} p\left(X_i^{(t)}|\mathbb{V}_{pa\left(X_k^{(t)}\right)}\right)$, where $X_k^{(t)}$ indicates the value of random variable $X_k$ at time $t$. (Friedman et al.).

A DBN includes multiple sub-networks at different time slices with the order being consistent with the time flow. Each node corresponds to a variable state at a given discrete time slice. Intra-slice edges indicate the relationship between variables within the same time slice, while inter-slice edges represent the temporal relationship between variables across different time slices. In summary, any variable in a DBN at time $t$ may depend on other variables at the same time slice and/or on the variables from the previous time slice.

To simplify the DBN structure, two assumptions are commonly used: the Markov assumption and the time invariance assumption. The Markov assumption postulates that all nodes satisfy the Markovian condition, i.e., $P\left(X^{(t+1)}\middle|X^{(0)}, \dots, X^{(t)}\right) = P\left(X^{(t+1)}\middle|X^{(t)}\right)$, which means that the states of variables at time $t$ only depend on the states at time $t+1$. The second assumption implies that the underlying structure does not change over time, i.e., the transition probability $P\left(X^{(t+1)}\middle|X^{(t)}\right)$ is independent of $t$. Note that the term "dynamic" only means that DBNs are able to model dynamic systems. A DBN is defined by the union of two parts: a prior network and a transition network, denoted as $\mathcal{G} = (\mathcal{G}_0, \mathcal{G}_{\rightarrow})$, where $\mathcal{G}_0$ is the prior which refers to the initial state (see Fig. 4.a), and $\mathcal{G}_{\rightarrow}$ is the transition network that represents the probability $P\left(X^{(t+1)}\middle|X^{(t)}\right)$ for all $t$ (see Fig4.b). Considering a finite interval $\{0, \dots, T\}$, a DBN can be unrolled into a BN over the time, as shown in Fig. 3.4.

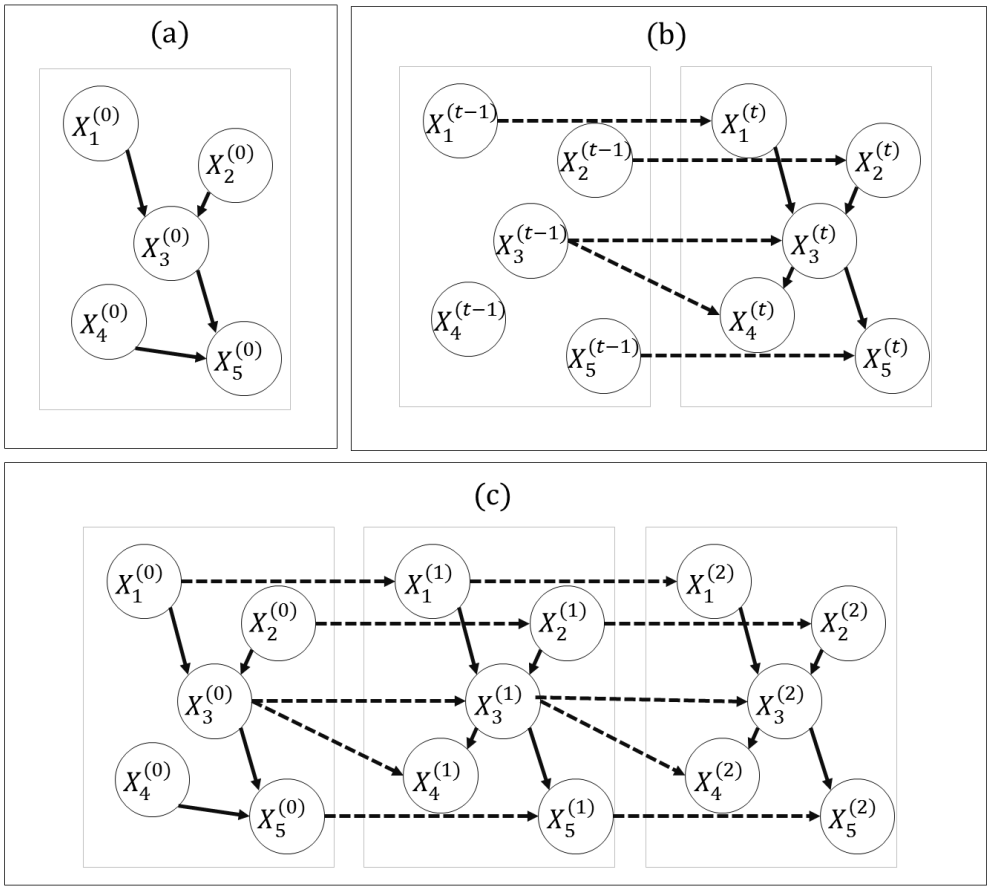


Figure 3.4 (a) A prior network $\mathcal{G}_0$; (b) a transition network $\mathcal{G}_{\rightarrow}$; (c) the corresponding unrolled network for three time slices.

## 3.3 Inference

In a BN, the observable nodes are denoted as *evidence* variables and unobservable nodes are denoted as *hidden* variables. This process of computing the posterior distribution of variables given evidence is called *inference*. Several inference algorithms have been developed for different types of BN. These algorithms can be categorized into two types: exact and approximate inference (Li & Mahadevan, 2018). Examples are shown in Table 3.1.

Table 3.1 Inference methods for different types of variables.

| Network | Variables | Exact inference | Approximate inference |
|---|---|---|---|
| BN | Discrete | • Variable elimination<br>• Junction tree<br>• Arc reversal<br>• Recursive decomposition | IS<br>• Logic sampling<br>• Adaptive sampling<br>MCMC<br>• Gibb sampling<br>• Metropolis sampling |
| GBN | Continuous | • Multivariate Gaussian | |
| CLGBN | Discrete, Continuous | • Multivariate Gaussian | |

| | | | |
|---|---|---|---|
| DBN | Discrete | • Forward-Backward<br>• Frontier | • Boyern-Koller |
| | Continuous | • Kalman filter | • Extended Kalman fitler<br>• Unscented Kalman filter<br>• Particle filter |

### 3.3.1 Exact Inference

Exact inference algorithms for discrete variables have been well-established, such as Variable Elimination (VE) (Zhang, 1994), junction tree algorithm (also known as clique tree) (Lauritzen & Spiegelhalter, 1988), the arc reversal method (Shachter, 1986), and recursive decomposition approach (Cooper, 1990).

The most common exact inference method is Variable Elimination, which eliminates the variables that are irrelevant to the query one-by-one. The VE algorithm repeatedly performs two factor operations: product and marginalization, where a *factor* is a multi-dimensional table assigning values of a set of variables. The factor product operation is defined as the product of two factors (see Fig. 3.5a). The factor marginalization is to sum over the possible values of one variable to obtain the marginal contribution of another as shown in Fig. 3.5b.

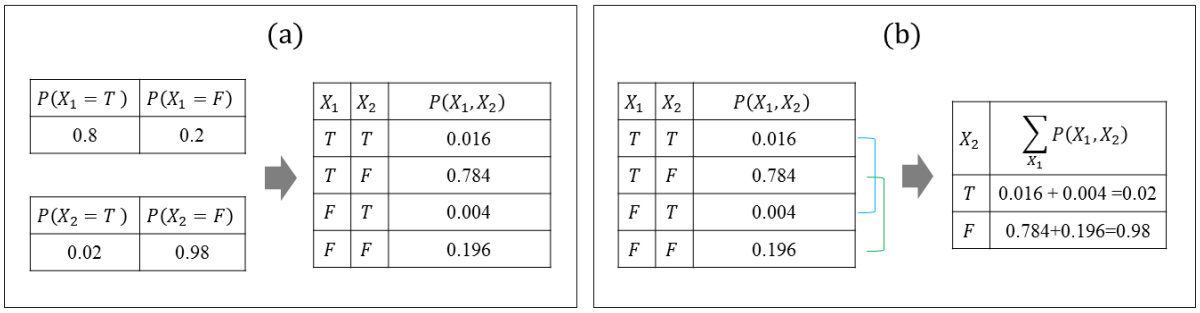


Figure 3.5 Examples of the factor operations for VE algorithm: (a) factor product; (b) factor marginalization.

For example, a query $P(X_5)$ in Fig 3.2 can be computed as below. The approach is to marginalize other variables from the join probability.

$$P(X_5) = \sum_{X_1}\sum_{X_2}\sum_{X_3}\sum_{X_4} P(X_1)P(X_2)P(X_3|X_1,X_2)P(X_4|X_2)P(X_5|X_3)$$

$$=\sum_{X_2}\sum_{X_3}\sum_{X_4} P(X_2)P(X_4|X_2)P(X_5|X_3)\psi_1(X_2,X_3)$$

$$=\sum_{X_3}\sum_{X_4} P(X_5|X_3)\,\psi_2(X_3,X_4)$$

$$=\sum_{X_4} \psi_3(X_4,X_5)$$

$$=\psi_4(X_5)$$

Starting with $X_1$, the factors involving $X_1$ are collected and used to create a new factor $\psi_1(X_2,X_3) = \sum_{X_1} P(X_1)P(X_3|X_1,X_2)$. Thus, $X_1$ can be eliminated by this new factor. By repeating the procedure for $X_2$, the factors involving $X_2$ are used to obtained another new factor

$\psi_2(X_3, X_4) = \sum_{X_1} P(X_2)P(X_4|X_2)\psi_1(X_2, X_3)$. Finally, after applying the same procedure to $X_3$ and $X_4$, one has $\psi_3(X_4, X_5) = \sum_{X_3} P(X_5|X_3)\psi_2(X_3, X_4)$ and $\psi_4(X_5) = \sum_{X_3} \psi_4(X_4, X_5)$.

The complexity of VE depends on the structure and the order of elimination, and finding the optimal order is NP-hard (Arnborg & Proskurowski, 1989). Approaches are developed to improve the performance by selecting a better order (Darwiche, 2009). In general, the complexity of the exact inference algorithms is exponentially increased with the network size (Lauritzen & Spiegelhalter, 1988), and the exact inference in Bayesian Network has been proved to be NP-hard (Cooper, 1990). Therefore, the approximate inference algorithms have been investigated and will be discussed later.

In many applications, variables can be continuous or mixed continuous-discrete. The network consists of both continuous and discrete variables is called the hybrid network. One way to deal with such cases is to discretize those continuous variables so that the exact inference can be employed. The quality of this approach depends on the number of bins, while a large number of bins increases the complexity of computation. To improve the quality of the exact inference for the hybrid networks, Lauritzen (1992) proposed an approach for CLGBNs. This study shows that the junction tree algorithm can be employed as long as the local distribution of the continuous variable is a conditional linear Gaussian distribution. More algorithms aiming to improve the inference efficiency can be found in the literature (Madsen & Jensen, 1999; Cowel, 2005; Salmerón et al.,2018).

As the joint probability of a GBN follows the multivariate Gaussian distribution, the inference task can be done through the same solution for the multivariate Gaussian model. Since the linear regression model also focuses on the conditional probability distribution of the response given the values of the independent variables, the relationship between a variable of a GBN and its parents can be formed as a linear regression model. The inference which queries the probability of a child node given its parents can be simply done through a linear regression model.

The general inference query for DBNs is to compute the probability of unknown variable $X_h^{(t)}$ at time $t$, given all the evidence during a certain time frame, denoted as $P\left(X_h^{(t)}|X_e^{(1)}, \dots, X_e^{(t)}\right)$, where $X_h$ is the unknown variable and $X_e$ is the evidence variable which can be observed. For a DBN with discrete variables, the exact inference is possible by unrolling the DBN over time-slices and applying the same inference approach as proposed in BN (Murphy 2002). However, it can be intractable in a large sized DBN. For a DBN with linear Gaussian variables, the inference task can be done through the Kalman filter (Minka, 1998).

### 3.3.2 Approximate Inference

Since the exact inference of BNs is computationally intractable, approximate inference algorithms have been studied. These algorithms can be classified into two categories: Importance Sampling (IS) and Markov Chain Monte Carlo (MCMC) (Kass et al.,1998). The IS approach generates samples independently, such as the logic sampling (Henrion, 1988) and adaptive important sampling (Cheng & Druzdzel, 2000). The MCMC method generates

samples sequentially and each sample also depends on the previous sample, such as the Gibb sampling (Geman & Geman, 1984) and the Metropolis sampling (Metropolis et al., 1953). These sampling-based approaches can apply to discrete and continuous variables without any distribution assumptions. Approximate inference algorithms for DBNs have been developed as well, such as the Boyern-Koller algorithm (Boyern & Koller, 2013), the factor frontier (Murphy, 2002), the extended Kalman filter (Jazwinski, 1970), and the particle filter (Kitagawa, 1996). These algorithms aim to deal with non-linear cases or non-Gaussian distribution, or to improve the inference efficiency. The details of these algorithms will not be discussed in this thesis but can be found in the literature (Guo & Hsu, 2002; Heskes & Zoeter, 2002).

## 3.4 Structure Learning

When the relationships among variables are explicit, the network structure is easily defined. However, the casualties are sometimes implicit and the structure should be learned from data. In this section, the details of structure learning are explained.

### 3.4.1 Bayesian Network Learning

Learning the structure of Bayesian Networks can be difficult and computationally intensive because the cardinality of the set of possible networks is enormous. There are two main categories of approaches for learning the graphical structure from data: constraint-based and score-based (Scutari, 2014). Constraint-based algorithms identify the conditional independencies of all variables through statistical tests that determine if each edge exists or not. The procedure starts with a fully connected undirected graph, and then determines the conditional independencies of each pair of variables given a subset of other variables. Many algorithms have been proposed, such as Inductive Causation (IC) algorithm (Verma & Pearl, 1992), PC algorithm which is named after its inventors (Spirtes et al., 2000), and Glow-Shrink (GS) algorithm (Margaritis, 2003). The outcomes of constraint-based algorithms are affected by the testing order, and some algorithms can be inefficient when dealing with a large number of variables (Margaritis, 2003).

Score-based algorithms firstly score each possible graphical structure based on how well it fits the observed data, and the structure with the highest score is selected. Considering a set of models $\mathcal{G}_s$, each model $\mathcal{G} \in \mathcal{G}_s$ indicates a possible network structure. The probability of model $\mathcal{G}$ given data $\mathcal{D}$ can be written as,

$$p(\mathcal{G}|\mathcal{D}) = \frac{p(\mathcal{D}|\mathcal{G})p(\mathcal{G})}{p(\mathcal{D})}. \tag{3.1}$$

Since $p(\mathcal{D})$ is independent of model $\mathcal{G}$, we can say, $p(\mathcal{G}|\mathcal{D}) \propto p(\mathcal{D}|\mathcal{G})p(\mathcal{G})$. A Bayesian score is defined as $Score_B(\mathcal{G},\mathcal{D}) = \log p(\mathcal{D}|\mathcal{G}) + \log p(\mathcal{G})$. Generally, it is assumed that $p(\mathcal{G})$ is a uniform distribution that can be ignored in the expression. From a Bayesian learning point of view, the parameters are random variables, denoted as $\theta_{\mathcal{G}}$, and thus, the marginal probability in Bayesian scores can be rewritten as $p(\mathcal{D}|\mathcal{G}) = \int p(\mathcal{D}|\mathcal{G},\theta_{\mathcal{G}})p_{\mathcal{G}}(\theta_{\mathcal{G}})d\theta_{\mathcal{G}}$, where $p_{\mathcal{G}}(\theta_{\mathcal{G}})$ is the prior distribution for the parameter $\theta_{\mathcal{G}}$. Suggestions of prior distributions are provided in the

literature (Cowell et al., 1999). However, the computation of the marginal likelihood is not simple because there is no closed form expression. The Bayesian Information Criterion (BIC) can be considered as an alternative that approximates the logarithm of $p(\mathcal{D}|\mathcal{G})$ because $Score_{BIC}(\mathcal{G},\mathcal{D}) = \log\left(\hat{L}\right) - \frac{1}{2} d_{\mathcal{G}} \log n$, where $\hat{L} = p(\mathcal{D}|\mathcal{G},\hat{\theta}_{\mathcal{G}})$ is the maximum value of the likelihood function, $\hat{\theta}_{\mathcal{G}}$ is the maximum likelihood estimate, $d_{\mathcal{G}}$ is the model complexity and $n$ is the sample size (Cowell et al., 1999). The BIC score does not require prior information $p_{\mathcal{G}}(\theta_{\mathcal{G}})$ so that it can be practical in real cases, because the prior information is usually hard to obtained. Other score functions can be used for structure learning (De Campos, 2006).

The objective of the score-based algorithms is to find an optimal structure that maximizes the score. However, this task is known to be NP-hard (Chickering et al.,1994). The standard approach to solve this problem is to perform a heuristic search. Many heuristic search algorithms have been proposed for leaning the BN structure (Chickering 2002; Elidan et al., 2002) but some are complicated and hard to implement. The simplest search algorithm – Hill Climbing (HC), can be a practical choice in terms of the trade-off between complexity and efficiency (Teyssier & Koller, 2005). Hill-Climbing is employed to conduct a greedy search on the feasible space of the directed graphs with BIC being the adopted scoring function. Generally, Hill-Climbing starts from either an empty, full or random network structure, and considers every possible movement of the current network, including adding an edge, removing an edge, or reversing the direction of an edge. The movement with the highest score is selected and the procedure is repeated. An example is shown in Fig. 3.6. The learning procedure stops when no improvement can be achieved by modifying any single edge.

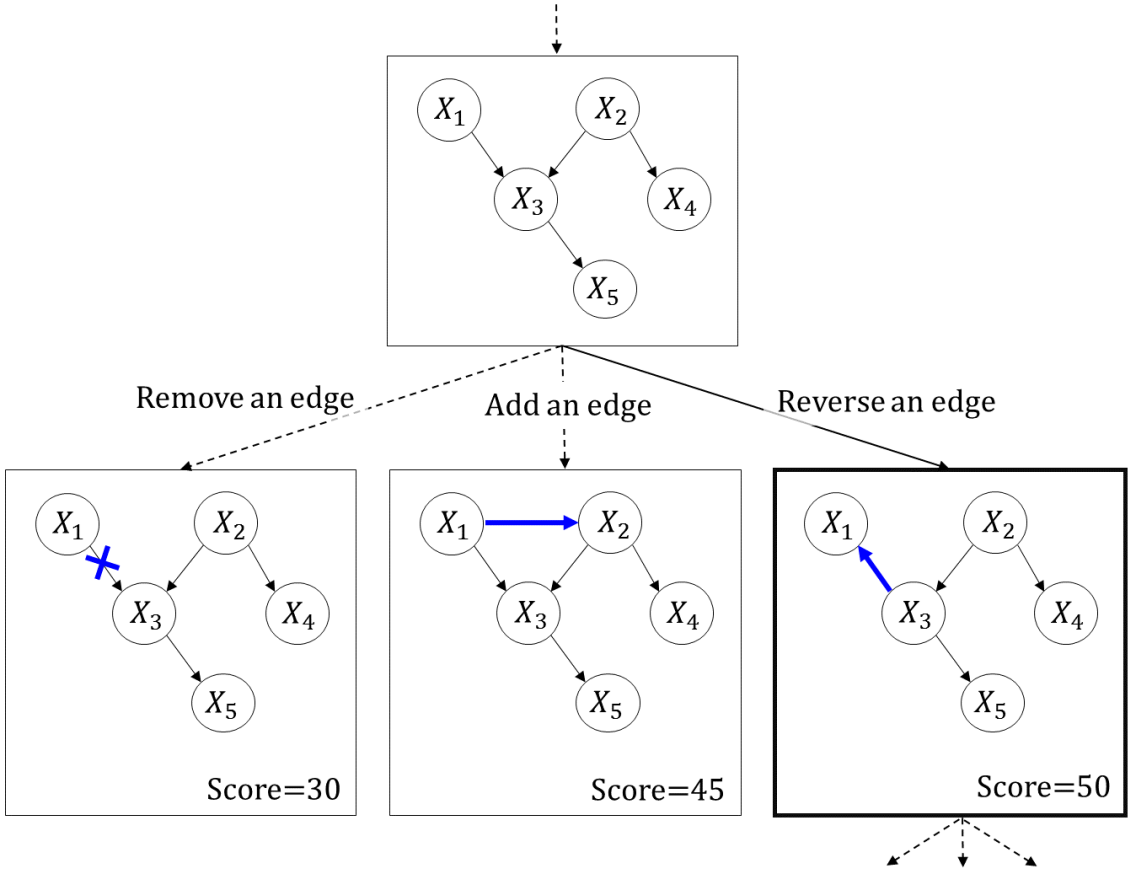


Figure 3.6 The illustration of the Hill-Climbing procedure (Margaritis, 2003).

Since Hill-Climbing is a greedy search algorithm, the global optimum will not be easily reached. To overcome this limitation, a random restart step is introduced to uniformly cover the search space. The optimal number of random-restarts and their corresponding perturbed edges are empirically determined at a preliminary analysis stage. The number of perturbed edges indicates the number of edges that will be randomly altered given the best network at the current moment.

The procedure of Hill-Climbing with a random restart step has been developed in R by Scutari (Scutari, 2009). The pseudocode is presented in Algorithm 3.1. The procedure starts with an empty network denoted as $\mathcal{G} = \mathcal{G}_{ini}$, and there are two stages performed in each iteration. The first stage is to find the best movement out of the three options: adding, removing an edge, or reversing the direction of an edge. The currently best network as $\mathcal{G}_{best}$ is updated accordingly. If the score of the currently best network $S_{best}$ is higher than the current score $S$, $\mathcal{G}_{best}$ will be used for the next iteration. If there is no movement that can improve the score, i.e., $S_{best} = S$, the second stage – random restart with the perturbing process will be activated. The restart network $\mathcal{G}_{restart}$ is defined for the perturbing process. The initial restart network is defined as $\mathcal{G}_{restart} = \mathcal{G}_{ini}$. If the score of $\mathcal{G}_{best}$ is higher than the score of $\mathcal{G}_{restart}$, $\mathcal{G}_{restart}$ is set to $\mathcal{G}_{best}$. Based on the restart network $\mathcal{G}_{restart}$, some edges will be randomly altered, and a new network $\mathcal{G}_{perturb}$ can be obtained. This network $\mathcal{G}_{perturb}$ will be used for the next iteration. If there is no operation that can increase the score and the number of random restarts has reached the specified limit, the Hill-Climbing procedure terminates.

**Algorithm 3.1** Hill-Climbing Structure Learning

$\boldsymbol{StructureLearning}(\mathcal{G}_{ini}, \mathcal{D}, restart, perturb)$

1. $\mathcal{G} \leftarrow \mathcal{G}_{ini}$; $\mathcal{G}_{best} \leftarrow \mathcal{G}$ ; $\mathcal{G}_{restart} \leftarrow \mathcal{G}$
2. $S \leftarrow Score(\mathcal{G}, \mathcal{D})$; $S_{best} \leftarrow S$; $S_{temp} \leftarrow 0$
3. $restart\_cnt \leftarrow 0$
4. **repeat**
5. #first stage : find best movement
6. **for** each pair of nodes $(V_i, V_j)$:
7. **if** $e_{ij} \notin \mathbb{E}$ **and** $e_{ji} \notin \mathbb{E}$ **then** $\mathbb{E}_{add} \leftarrow \mathbb{E} \cup \{e_{ij}\}$; $S_{temp} \leftarrow Score(\mathcal{G}_{add} = (\mathbb{V}, \mathbb{E}_{add}), \mathcal{D})$
8. **if** $S_{temp} > S_{best}$ **then** $S_{best} \leftarrow S_{temp}$ ; $\mathcal{G}_{best} \leftarrow \mathcal{G}_{add}$ **end if**
9. **end if**
10. **end for**
11. **for** each pair of nodes $(V_i, V_j)$:
12. **if** $e_{ij} \in \mathbb{E}$ **then** $\mathbb{E}_{drop} \leftarrow \mathbb{E} \setminus \{e_{ij}\}$ ; $S_{temp} \leftarrow Score(\mathcal{G}_{drop} = (\mathbb{V}, \mathbb{E}_{drop}), \mathcal{D})$
13. **if** $S_{temp} > S_{best}$ **then** $S_{best} \leftarrow S_{temp}$; $\mathcal{G}_{best} \leftarrow \mathcal{G}_{drop}$ **end if**
14. **end if**
15. **end for**
16. **for** each pair of nodes $(V_i, V_j)$:

17. **if** $e_{ij} \in \mathbb{E}$ **then** $\mathbb{E}_{reverse} \leftarrow \left(\mathbb{E} \cup \{e_{ji}\}\right) \setminus \{e_{ij}\}; S_{temp} \leftarrow Score(\mathcal{G}_{reverse} = (\mathbb{V}, \mathbb{E}_{reverse}), \mathcal{D})$
18. **if** $S_{temp} > S_{best}$ **then** $S_{best} \leftarrow S_{temp}; \mathcal{G}_{best} \leftarrow \mathcal{G}_{reverse}$ **end if**
19. **end if**
20. **end for**
21. #Second stage: perform random restart if no improvement
22. **if** $S_{best} = S$ **then**
23. **If** $restart\_cnt < restart$ **then**
24. $restart\_cnt \leftarrow restart\_cnt + 1$
25. **if** (Score($\mathcal{G}_{best}, \mathcal{D}$) > Score ($\mathcal{G}_{restart}, \mathcal{D}$))
26. $\mathcal{G}_{restart} = \mathcal{G}_{best}$
27. **end if**
28. $\mathcal{G}_{perturb} = \mathcal{G}_{restart}$
29. **for p in 1:** $perturb$
30. $\mathcal{G}_{add} = random\_add\_edge(\mathcal{G}_{perturb})$
31. $\mathcal{G}_{drop} = random\_drop\_edge(\mathcal{G}_{perturb})$
32. $\mathcal{G}_{reverse} = random\_reverse\_edge(\mathcal{G}_{perturb})$
33. $\mathcal{G}_{perturb} = sample(\mathcal{G}_{add}, \mathcal{G}_{drop}, \mathcal{G}_{reverse})$
34. **end for**
35. $\mathcal{G} = \mathcal{G}_{perturb}; S \leftarrow Score(\mathcal{G}, \mathcal{D})$
36. **next**
37. **else**
38. **break**
39. **end if**
40. **else**
41. $\mathcal{G} = \mathcal{G}_{best}$ ; $S \leftarrow S_{best}$
42. **end if**
43. **end repeat**

As a BN is a DAG, any edge causes a loop will be considered as the violation. During the learning procedure, the movement of each iteration will be examined for its legality. The illegal edge will not be used for updating the network. In addition to identifying the edges from data, the algorithm also provides the flexibility to integrate pre-defined directions on specific edges. Based on the domain knowledge, the edges that present known causalities can be defined as a whitelist, and the edges that present infeasible causalities will be defined as a blacklist (Scutari, 2009). During the structure learning procedure, any movement against either whitelist or blacklist will be considered as a violation.

### 3.4.2 Dynamic Bayesian Network Learning

The algorithms used for learning the structure of BNs can be used to learn that of DBNs as well. As described in Section 3.2.4, a DBN $\mathcal{G}$ consists of two parts: a prior network $\mathcal{G}_0$ and a transition network $\mathcal{G}_{\rightarrow}$. Friedman (Friedman et al., 1998) has proved that the task of learning a

DBN can be done through learning $\mathcal{G}_0$ and $\mathcal{G}_{\rightarrow}$ separately because of the decomposability of the score function. Assuming there are $q$ variables $\left\{X_1^{(1)}, \ldots, X_1^{(\tau)}, \ldots, X_q^{(1)}, \ldots, X_q^{(\tau)}\right\}$ on a finite time interval $[0, \tau]$. The dataset is defined as $\mathcal{D}^{(t)} = \begin{bmatrix} X_1^{(t)} & \cdots & X_q^{(t)} \end{bmatrix}$, where $t = 0, 1, \ldots, \tau$. This dataset, denoted as $\mathcal{D}_0 = \mathcal{D}^{(t)}$, will be used to learn $\mathcal{G}_0$ (see Fig. 3.7a). For learning $\mathcal{G}_{\rightarrow}$, the dataset is defined as $\mathcal{D}_{\rightarrow} = \{\mathcal{D}^{(t)}, \mathcal{D}^{(t-1)}\}$ as shown in Fig. 3.7b, where $t = 1, \ldots, \tau$.

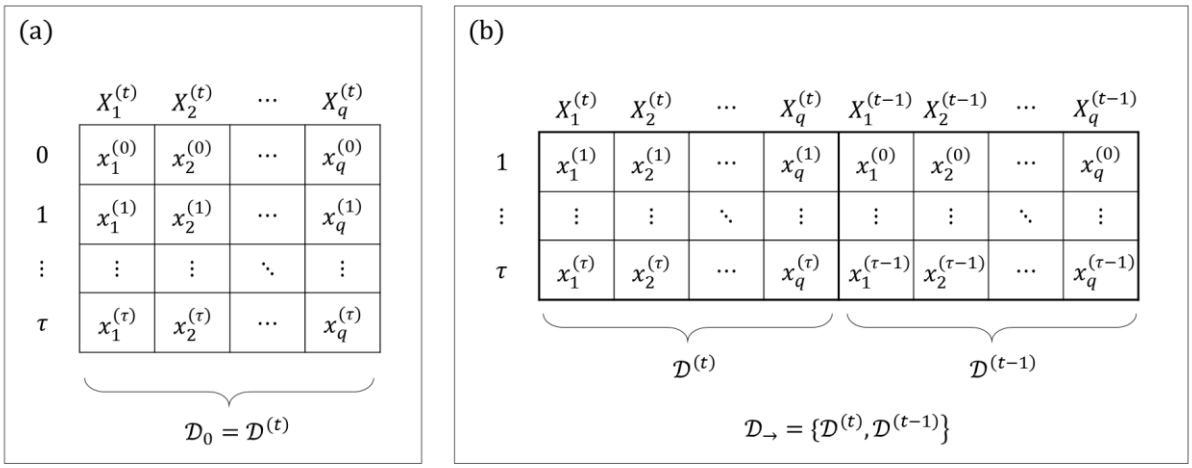


Figure 3.7 The datasets for learning a DBN: (a) $\mathcal{D}_0$ for learning $\mathcal{G}_0$, (a) $\mathcal{D}_{\rightarrow}$ for learning $\mathcal{G}_{\rightarrow}$.

The steps of the learning procedure are listed as:

1. Learning $\mathcal{G}_0$ with dataset $\mathcal{D}_0$ by standard BN structure learning algorithm.
2. Learning $\mathcal{G}_{\rightarrow}$ with dataset $\mathcal{D}_{\rightarrow}$ subject to the following constraints. Any edge from $X_i^{(t-1)}$ to $X_j^{(t)}$ is treated as a violation, where $i, j = 1, \ldots, q$. Every variable in $\mathcal{D}^{(t)}$ is considered as a root node, i.e., the node without parents.
3. By combining $\mathcal{G}_0$ and $\mathcal{G}_{\rightarrow}$, the final DBN can be obtained.

According to the definition proposed by Friedman et al. (1998), the structure of the initial network may be quite different from the structure of the other time slices. Therefore, the transition network might exclude the initial time slice. Other algorithms for learning DBNs have been studied in the literature, such as the genetics algorithm (Gao et al., 2007), dynamic MMHC (Max-Min Hill Climbing algorithm) (Trabelsi et al., 2013).

## 3.5 Applications

BNs can be used in various domains, such as the robot localization (Kortenkamp & Weymouth, 1994), gene expression analysis (Friedman et al., 2000), and weather prediction (Cofiño et al., 2002). For manufacturing industry, researchers have practiced applying BNs to fault diagnosis (Yang & Lee, 2012; Cai et al., 2017). Furthermore, Deventer (2004) proposed a Bayesian controller that uses Bayesian Network to model and control the dynamic system. Considering a SISO system, its state-space description can be written as

$$x_{i+1}^s = A_{BN} x_i^s + b_{BN} u_i \text{ and}$$

$$y_i^m = C_{BN} x_i^s + d_{BN} u_i,$$

where $x_i^s$ represents the state of the system, $u_i$ is the controllable variables, and $y_i^m$ is the process output. Assuming that $x_i^s$ and $u_i$ follow the Gaussian distributions, a Bayesian network can be used to express the system by the conditional distributions,

$$p(x_{i+1}^s | x_i^s, u_i) = \mathcal{N}\left(\beta_1 \begin{bmatrix} x_i^s \\ u_i \end{bmatrix}, \Gamma_1\right) \text{ and}$$

$$p(y_i^m | x_i^s, u_i) = \mathcal{N}\left(\beta_2 \begin{bmatrix} x_i^s \\ u_i \end{bmatrix}, \Gamma_2\right),$$

where $\beta_1 = [A_{BN} \quad b_{BN}]$ and $\beta_2 = [C_{BN} \quad d_{BN}]$. Let the desired output value be the evidence, the control setting can be derived by the marginal distributions. Deventer's approach requires that the structure is pre-defined and the parameters are learned from experiments.

DBNs have been applied in other domains as well, such as in genome informatics (Perrin et al., 2003; Murphy & Mian, 1999), speech recognition (Zweig & Russell, 1998), brain connectivity study (Rajapakse & Zhou, 2007), and disease diagnosis (Charitos et al., 2009). Some studies apply DBNs in the manufacturing industry as well. Lerner et al. (2000) employed a hybrid Dynamic Bayesian Network to track and diagnose faults in the chemical industry. They consider continuous variables as the system states and discrete variables indicating certain events. Tobon-Mejia et al. (2012) demonstrate how to access the health condition of the milling machine by DBNs based on the sensor data.

## 3.6 Conclusion

This chapter briefs the background of BNs, including the properties of DAGs, several networks depending on the variable types, and their corresponding inference approaches. The algorithms for learning the structure of BNs and DBNs are introduced as well. In practice, choosing the appropriate methods for inference and structure learning depends on the problems domain and the data characteristics.

In this research, since most process parameters are highly correlated to their previous states, the DBN is chosen as the fundamental model such that the temporal effect can be captured. To simplify the computing complexity, we assume that all variables are continuous and following Gaussian distributions, and the relationships between variables are linear. The inference method and learning approach for GBNs can be used in DBN. Additionally, the evidence variables are always parents in this study. Therefore, the inference problem in this thesis is relatively simple and can be easily solved by the prediction procedure of a linear regression model. Furthermore, the parameters learning can be done during the structure learning as the network essentially presents a multivariate Gaussian model. However, these assumptions usually do not fit well the real world situation such that the accuracy of a learned DBN may be reduced. On the other hand, considering the implementation cost, staring with a simpler approach shall be a practical option. Since the objective of this thesis is to demonstrate the capability of using a DBN as a foundation of the proposed framework, we will focus more on its applications rather than the inference details. The structure learning procedure of DBNs has been described in Section 3.4.2. In this thesis, instead of learning prior network and transition network separately, a two-phase learning algorithm is proposed. This algorithm aims

to learn the intra-slice edges and inter-slice edges at the same time. The details of the proposed learning approach will be presented in Section 4.3.

# 4 Integrated Physics-Informed Control Framework

The proposed framework can be schematically described by the block diagram presented before, in Fig. 1.2, In this chapter, the framework assumptions are presented and discussed (Section 4.1), and the details of each building block are introduced in the following sections (see Fig. 4.1).

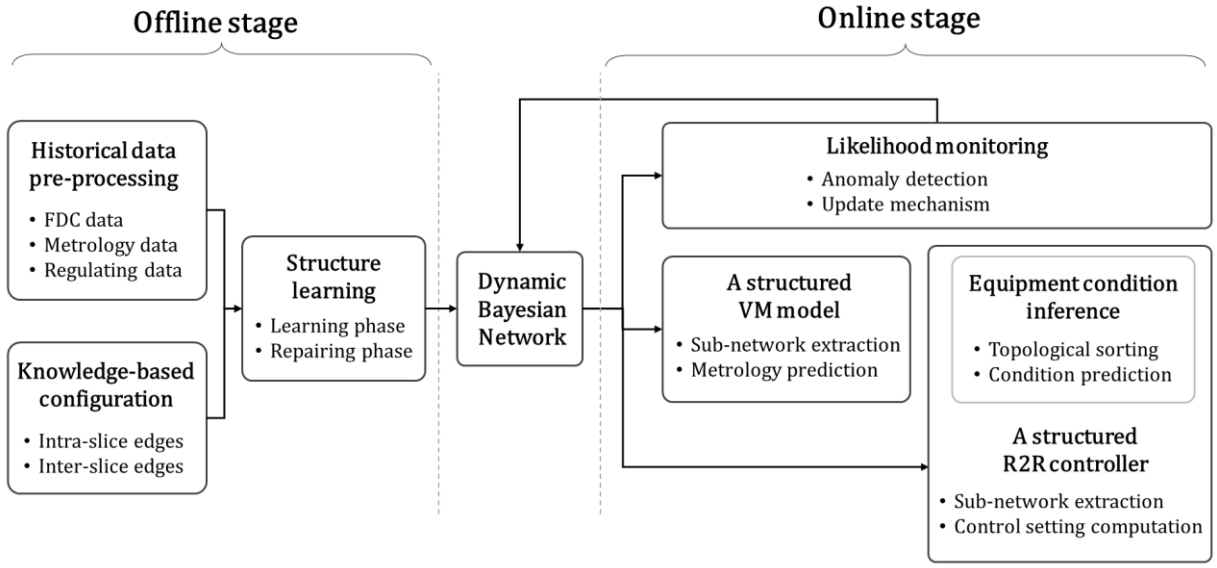


Figure 4.1 Details of the Integrated Physics-Informed Control Framework.

In general, a data-driven approach involves using accessible data to learn the main variation patterns. When done using a previously collected "static" dataset, it is defined as offline learning. The procedure for offline learning starts with historical data pre-processing. Since the input of learning a model should meet specific format, data transformation is a necessary step. As there will be several data sources, data consolidation is required as well. Both parts will be explained in Section 4.2. In the meantime, in order to learn a model which fits well the available physical information, several configurations should be set up before learning model. In this way, some meaningless patterns learned from data that violate the physical law can be prevented. The details of these configurations will be given in Section 4.3. In Section 4.4, by combining the above two inputs, it is possible to learn the DBN. Although a DBN can have multiple time slices, to simplify the structure, only a two-time-slice structure will be studied in this work.

A learned DBN is ready for various online applications, and will be presented following a top-down sequence, as shown in Fig. 1.7. A DBN-based monitoring approach will be introduced in Section 4.5, which is capable of detecting abnormalities and triggering updating mechanism if needed. In Section 4.6, we will show how a DBN can be used for prognosis, including metrology prediction and equipment condition prediction. The details of the proposed SRC will be illustrated in Section 4.7, including a sub-network extraction for identifying crucial variables, the equipment condition prediction procedure for assisting control decision making, and control value computation by levering all available information. In

Section 4.8, we will introduce an advanced SRC approach which incorporates a model update mechanism.

## 4.1 Assumptions

The proposed "*Integrated Physics-Informed Control Framework*" aims to gather all the available information and construct a general-purpose methodology, so that several derived function modules can be operated under the same structure. As described in Section 1.2.1, this framework based on a DBN can be extended in several aspects, such as across different process operations and different product types. In this thesis, the objective is to demonstrate the capability of the proposed framework and focus more on one of the function modules – R2R control. To clearly illustrate the details of the approach, we start with a simple framework with some assumptions. These assumptions will be explained in this section. More possible extensions which can release these assumptions will be discussed in Chapter 6.

*A. Operation*

In semiconductor manufacturing, producing a wafer requires hundreds of process operations. After some critical operations, the corresponding metrology operations will take place, to make sure the quality of wafer meets the target. Most control systems are tailored for the specific process operation. For example, the R2R controller for the deposition process is different from the controller for the etching process, because the controllers are designed based on their physical or chemical characteristics. In this thesis, we assume that the framework will be constructed for a specific process operation. Only the information regarding this operation will be considered, and a feedback R2R control system will be investigated.

*B. Equipment*

Generally, some equipment with the same capabilities can be used to operate the same process. Depending on the manufacturing schedule, different lots can be dispatched to different equipment so that the overall efficiency or cycle time can be optimized. Although those equipment are designed with the same capabilities, their performance and conditions can be different from each other due to either the usage or other random factors. The main idea of the proposed framework is to integrate the equipment information during the process, as different equipment conditions can lead to different control decisions. To simplify the presentation of the proposed approach, only single process equipment will be considered in this thesis, but the potential capability of comprising multiple equipment will be discussed in the last chapter.

*C. Product*

Equipment can usually be used to process various products. Depending on the design and specification, the process routes of each product type are different. In this context, the performance of the process reflected on each product may not be the same, because of the characteristics of product type or the impact caused by the former process operations. To filter this kind of variation, only one type of product will be considered in this thesis, but the design of high-mix framework surely should be a future task.

*D. Data Source*

Based on the first three assumptions, the required data are clear. The equipment data of a specified product type which collected from a specified equipment of a particular operation. The corresponding metrology data of these wafers will be gathered as well. We assume that the control setting is available and can either be obtained from equipment data or other information systems. In this thesis, we focus on dealing with these two primary data sources, equipment data and metrology data, which are the most common data sources in practice. More information, such as PM record or MES data, which might enhance the efficiency of process control, can be taken into account in the future framework.

### *E. Statistics assumptions*

According to the previous assumption, the data sources considered in this thesis will be of a numerical type. Although the core method – DBN does not require distributional assumptions, to mitigate the computation loading and reduce the complexity of the approach we assume that all variables follow Gaussian distributions, and the relationships among variables are linear.

### *F. Run-to-run control*

As explained in the first assumption, only feed-forward R2R control will be investigated in this thesis. Besides, we assume that the controllable variables have been determined by the process engineers, so choosing appropriate controllable variables will not be discussed here.

### *G. Wafer-level analytics*

The model is wafer-based, so different data sources will be transformed into wafer level. Generally, there are 25 wafers in a lot, and the order of the wafers can be one of the factors of process variation. For example, the idle time between lots can influence the equipment conditions, and the first wafer process after idle time can perform differently from others. This is also called the first wafer effect. In this thesis, we assume such systematic effects between lots to be negligible and will not be taken into account.

### *H. Sampling wafer for physical measurements*

Generally, only few sampling wafers will be sent to metrology tool to get physical measurement. In this thesis, those sampled wafers taken from processing equipment are called "*metrology runs*", following the nomenclature found in literature (Khan et al. 2008). To clearly explain the approach in the rest of thesis, we further define the remaining wafer as "*regular run*".

### *I. First order temporal dependency*

A DBN can present the temporal dependencies between variables on different time slices.These dependencies show how the equipment conditions or process outputs are affected by the former runs. To simplify the structure, we assume that the current run will be influenced only by the previous run, so that a two-time-slice DBN will be studied.

### *J. Identical Intra-slice structure*

Since the intra-slice edges present the interactions between variables, we assume that these interactions will not change in a short period, which is the time between two processing wafers. In this context, the learned DBN should have the same intra-slice structure.

Given the assumptions above, the breakdown of the proposed framework will be introduced in the following sections based on the modules shown in Fig. 4.1.

## 4.2 Offline Stage - Historical Data Pre-processing

In this research work, there are three types of datasets to be analyzed: FDC, metrology measurements, and regulating data. Since these datasets are collected at different granularity levels, data preprocessing is required and should be first applied.

### 4.2.1 FDC Data

In IC industry, multiple sensors are embedded in the process equipment, and they collect signals in real-time, such as the temperature readings. Typically, those sensors have a high resolution; for example, record the reading every second. These sensor readings are referred to as FDC data. We consider the general case where the process equipment is composed of multiple chambers, and each wafer passes through these chambers sequentially. Thus, each SVID in a different chamber will be treated as a distinct, unique quantity. Assuming the existence of $p_c$ SVIDs in chamber $c$, there are $p = \sum_c^{\mathcal{C}} p_c$ SVIDs, where $\mathcal{C}$ is the number of chambers in the process equipment. Note that the operation may require different steps in these chambers, depending on the settings of the recipe (see Figure 4.2).

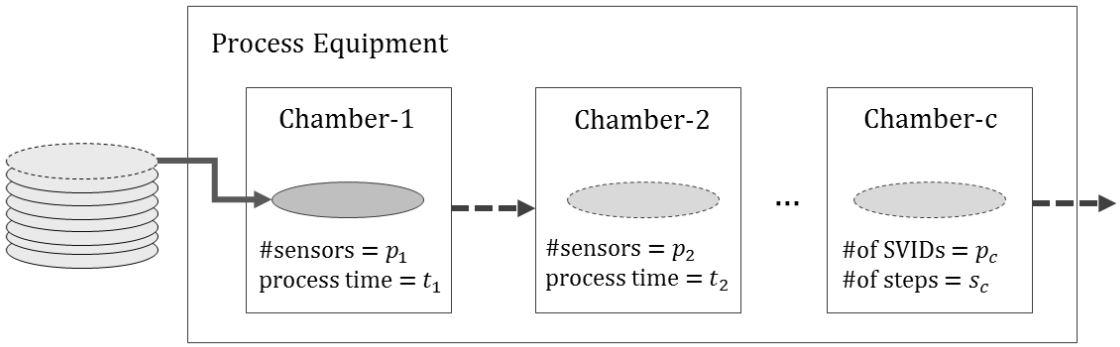


Figure 4.2 An example of a sequential process of multiple chambers within a process equipment.

The collected FDC temporal data for wafer $w$ in chamber $c$ can be represented by a table (see Figure 4.3). A cell includes multiple data points which indicate the observations per second of a SVID in a step. In order to transform the temporal FDC data into wafer-based data, the conventional approach is to summarize the observations in each step by descriptive statistics, such as the mean and standard deviation. The set of summary statistics for all SVIDs are taken as the features of the wafer that will be considered in the modeling stage.

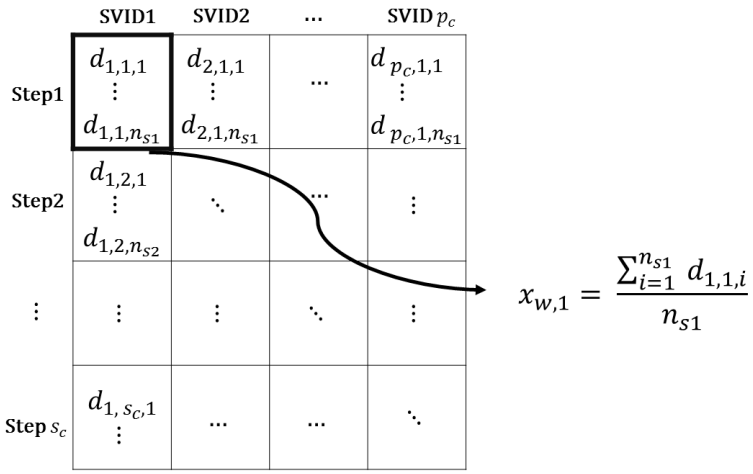


Figure 4.3 Figure 4.3 Schematic representation of the table of temporal data for wafer *w* in chamber *c*. Each cell can be summarized as a value that represents a process feature.

To extend the data processing from single chamber to all chambers, all features in a chamber must be collected and concatenated with those from the other chambers, as follows (see Figure 4.4). All the process characteristics of wafer $w$ in chamber $c$ can be summarized and expressed as an $s_c$ by $p_c$ matrix, where each element in the matrix indicates a feature of the wafer. This matrix is then reshaped to a 1 by $r_c$ row vector, where $r_c = s_c \times p_c$. Finally, the vectors for all the chambers are collected and concatenated into a 1 by $r$ vector. Consolidate the features of $n$ wafers and a $n$ by $r$ matrix can be obtained, denoted as $\mathbb{X} = [X_1 \quad \cdots \quad X_r]$. For each vector in $\mathbb{X}$, we defined it as an FDC variable.

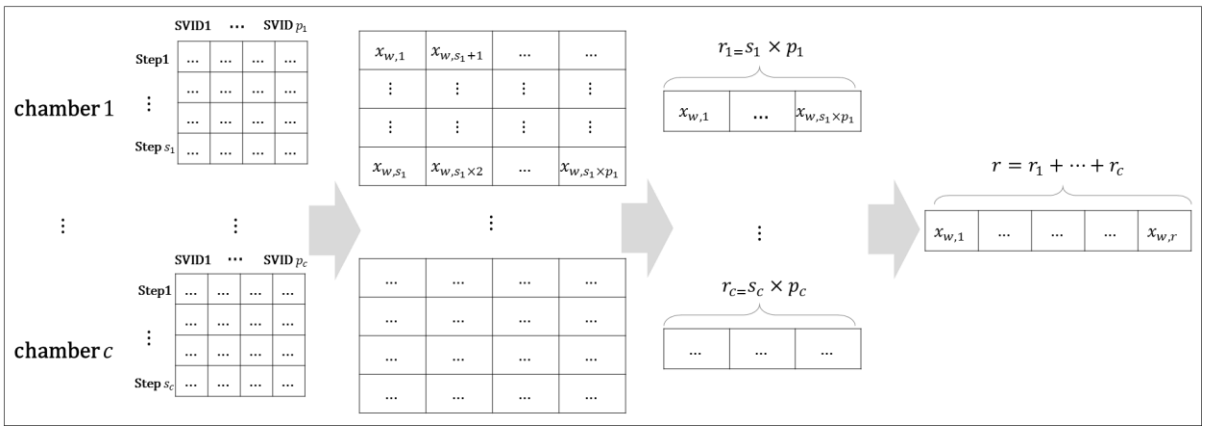


Figure 4.4 Figure 4.4 The feature extraction of the FDC temporal data for wafer *w* across all chamber.

The metadata associated to the features are important and need to be registered by tags attached to them. We propose three types of tags which are hierarchically related: $T_1$ is the tag indicating the operation, $T_2$ is the tag of chamber, and $T_3$ is the tag for the step. Considering a simple case that there are two operations: a process operation followed by a metrology measuring operation, we give $T_1 = 1$ for all FDC features collected during process operation and $T_1 = 2$ for the subsequent metrology variables. Furthermore, for each FDC feature, the tags, $T_2$ and $T_3$, are set according to the metadata of the features, e.g., a feature summarized from the second chamber in the first step would have tags: $T_2 = 2$ and $T_3 = 1$. This allows

tracing back any predictor to its original meaning. In addition, another tag $T_p$ that indicates the SVID of the features is also registered, such as if it is a temperature or pressure sensor.

### 4.2.2 Metrology Data

As explained in Section 2.1, depending on different process, there are relevant measurements data that can be used in the quality assessment, such as CD, depth. Usually only few wafers will be sent to metrology tools to obtain these measurements, which are also called *metrology data*. Generally, there are multiple measurements in a wafer as shown in Fig. 4.5. The sample dies indicate the location of the measurements. To evaluate the overall quality of a wafer, usually, the statistics of these measurements will be taken as the performance index, such as average or standard deviation. These indexes are presented in a wafer-level, and they are defined as metrology variables in the remaining of this manuscript. Suppose there are $\delta$ metrology variables in a process, the matrix of metrology variables is denoted as $\mathbb{Y} = [Y_1 \quad \cdots \quad Y_\delta]$.

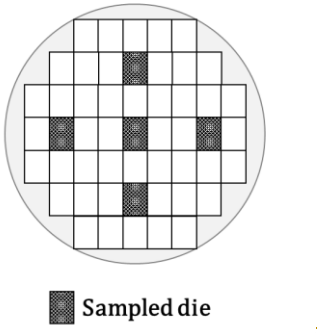


Figure 4.5 An example of the sampled dies of a wafer.

However, given the current limitations in the capacity of metrology tools and the reduced product life cycle of wafers, only a few of them can be sampled and sent for measuring. Depending on the critical level of product and process, there can be different sampling scheme. For example, a sampling scheme can be established as measuring slot 12 and slot 24 of every three lots, where the slot is the location (or order) of the wafers in a lot. In this thesis, one of the tasks is to learn the relationship between metrology variables and other variables. Therefore, the wafers without metrology data, i.e., regular run wafers as defined in Section 4.1, will be discarded during the learning phase. Several methods that are available to impute such missing data, but considering the proportion of regular run wafer is large, these methods may lead to bias. Thus, we choose the limited dataset which can present real phenomena.

### 4.2.3 Regulating Data

Considering two types of process, a process with R2R control and a process without R2R control. The *regulating data* of the first type process will include the control setting of each wafer. Note that if a lot-based R2R control is applied, the control setting of the wafer within a lot will be the same. For the second type of process, assume that the process engineers have determined the controllable variables for future R2R control, and the values of those

controllable variables can be found in FDC data. For example, if pressure is a future controllable variable, in order to analyze how pressure affects the process output, we can use the pressure readings of processed wafers which have been recorded in the historical FDC data. We define these readings of the controllable variable as *imitating regulating data*, which can be used to analyze how controllable variables affect the equipment states and process output. For learning a causality network, the imitating regulating data collected from the second type process should be superior to the real regulating data collected from the first type process. Because the manipulation may mask the actual physical behaviors, and it will be difficult to interpret different effects. Therefore, the imitating regulating data collected from a process without applying R2R control will be used in this thesis.

Since the control values of regulating data were collected from FDC data, which record the sensor reading of each wafer, the granularity of regulating data will be wafer-based as well. The controllable variables can be one or more, and the controllable variable matrix is defined as $\mathbb{U} = [U_1 \quad \cdots \quad U_k]$, where $k$ is the number of controllable variables.

### 4.2.4 Consolidation

After the data collection and transformation, these three sources of data, are consolidated at the wafer level, denoted as $\mathcal{D}^{(t)}$. For a two-time-slice DBN, all variables with one-period lag $\mathcal{D}^{(t-1)}$ will be included as well. Finally, all sources are put together in a $q$ by $n$ matrix $\mathcal{D}$,

$$\mathcal{D} = [\mathcal{D}^{(t-1)} \quad \mathcal{D}^{(t)}], \tag{4.1}$$

$$\mathcal{D}^{(t)} = \left[U_1^{(t)} \quad \cdots \quad U_k^{(t)} \quad X_1^{(t)} \quad \cdots \quad X_r^{(t)} \quad Y_1^{(t)} \quad \cdots \quad Y_\delta^{(t)}\right]$$

$$\mathcal{D}^{(t-1)} = \left[U_1^{(t-1)} \quad \cdots \quad U_k^{(t-1)} \quad X_1^{(t-1)} \quad \cdots \quad X_r^{(t-1)} \quad Y_1^{(t-1)} \quad Y_\delta^{(t-1)}\right],$$

where $q = (k + r + \delta) \times 2$, $n$ is the number of wafers with measurements.

Generally, a R2R controller may operate in a lot-based or wafer-based scheme, depending on the capability of equipment. In this work, a wafer-level dataset will be used for modeling and optimized controllable values calculation, thus, in the rest of this manuscript, a *run* indicates a wafer. To simplify the demonstration of the proposed approach, a Multiple-Input-Single-Output (MISO) system is used in the rest of this chapter, i.e., $k > 1$ and $\delta = 1$.

## 4.3 Offline Stage - Knowledge-based Configuration

The main objective of the proposed approach is to build a DBN which takes all sources of information into account: data and SMEs, so that this network can express the relationships among variables well. Therefore, instead of a pure data-driven technique for inferring such a structure, we integrate existing domain knowledge so that the final network reflects all sources of information available.

The directions of the edges in a DBN indicate the orientation of the causal effects between the pair variables. Each edge in DBN can be either an intra-slice edge or an inter-slice edge. Intra-slice edges imply casualties within the same wafer; while inter-slice edges suggest

casualties across different wafers. Two types of configurations need to be defined: intra-slice and inter-slice. Several blocking edges will be set according to these configurations and consolidated into a blacklist $\mathcal{L}$. Note that a blocking rule implies that the causal effect between two variables will not be assigned. A blacklist $\mathcal{L}$ is a two-column matrix, and each row presents a blocking edge $e_{ij}$. The first column indicates the source node $X_i$ of a edge, and the second column shows the sink node $X_j$ of this edge. Such that every $e_{ij} \in \mathcal{L}$ will be excluded during DBN structure learning.

To simplify the notation in this section, $X_i$ will be used to indicate any variable in the dataset, which can be a FDC variable, a metrology variable, or a controllable variable.

### 4.3.1 Intra-slice Configuration

For the intra-slice configuration, two types of characteristics of edges should be examined. The first characteristic is time-dependency, while the second one regards physical and chemical interactions. In order to define the blocking rules in a systematic and compact way, the general organization of process operations will be briefly reviewed.

The first type of intra-slice configuration considers the time dependency between the variables and presents a hierarchical structure. Since the manufacturing process involves multiple operations and an operation can refer to either a process operation or a metrology measuring operation, those operations can be considered as the first level of hierarchy (see Fig. 4.6). In the case of a process operation, it may consist of multiple sub-processes, which relate to different chambers and can be expressed at the second level of the hierarchy; finally, the third level indicates that each sub-process may include complex steps which are predefined by the recipe

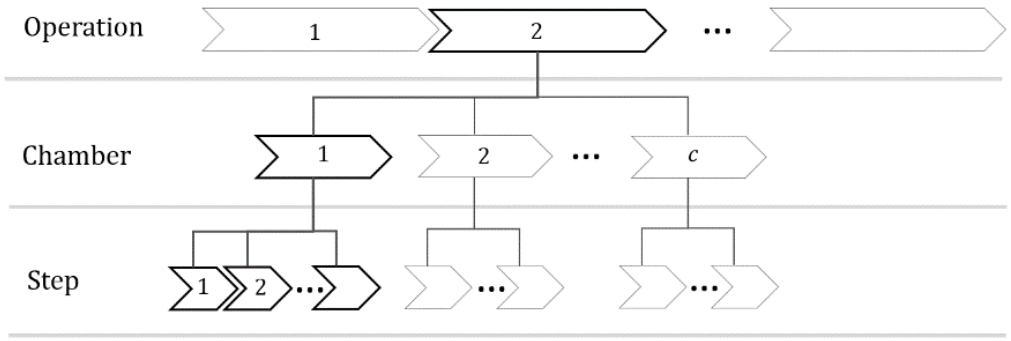


Figure 4.6 The hierarchical structure displays the time dependency between the three levels of activities of the production of wafers.

The sequential nature of the process leads to a causal structure among the various variables. The three types of tags introduced in the previous section enable the rigorous definition of this structure. The relevant blocking rules were generated and compiled into the blacklist $\mathcal{L}$, as follows. Starting with the operation level, for each pair of features $(X_i, X_j)$, if $T_1(X_i) > T_1(X_j)$, $e_{ij}$ will be added to $\mathcal{L}$. Considering the chamber level, each pair features $(X_i, X_j)$ where $T_1(X_i) = T_1(X_j)$, if $T_2(X_i) > T_2(X_j)$ then it will also be included to $\mathcal{L}$.

Similarly, for each pair $(X_i, X_j)$ where $T_1(X_i) = T_1(X_j)$ and $T_2(X_i) = T_2(X_j)$, if $T_3(X_i) > T_3(X_j)$, $e_{ij}$ will be added into $\mathcal{L}$ (see summary of blocking rules in Table 4.1).

Table 4.1 Time dependency blocking rules.

| Level | Blocking rule |
|---|---|
| operation | For $(X_i, X_j)$, if $T_1(X_i) > T_1(X_j)$ then $e_{ij} \in \mathcal{L}$. |
| chamber | For $(X_i, X_j)$, where $T_1(X_i) = T_1(X_j)$, if $T_2(X_i) > T_2(X_j)$ then $e_{ij} \in \mathcal{L}$. |
| step | For $(X_i, X_j)$, where $T_1(X_i) = T_1(X_j)$ and $T_2(X_i) = T_2(X_j)$, if $T_2(X_i) > T_2(X_j)$ then $e_{ij} \in \mathcal{L}$. |

Next, the physical and chemical interactions between pairs of FDC variables will be examined. Let us assume there are $p$ SVIDs, and based on domain knowledge it is possible to define a $p \times p$ association matrix $\mathcal{M}$ such that each binary element $m_{p_1,p_2}$ in $\mathcal{M}$ specifies if the corresponding association is impossible or not,

$$m_{p_1,p_2} = \begin{cases} 0, \text{ if } p_1 \text{cannot cause } p_2; \\ 1, \text{ if } p_1 \text{may cause } p_2. \end{cases} \quad (4.2)$$

where, $p_1 = 1, \dots p$ and $p_2 = 1, \dots p$. In the case $m_{p_1,p_2} = 0$ than the association between SVIDs $p_1$ and $p_2$ is impossible; e.g., "the chamber pressure cannot affect the gas flow". For $m_{p_1,p_2} = 1$, SMEs leave the possibility of an association open, which will be inferred later on, by the data-driven learning process.

Based on this association matrix $\mathcal{M}$, more blocking rules can be generated and added to the blacklist, $\mathcal{L}$. For each $m_{p_1,p_2} = 0$, a loop should cover all pairs of features $(X_{j1}, X_{j2})$, and for the cases where $X_{j1}$ is summarized from SVID $p_1$, and $X_{j2}$ is summarized from SVID $p_2$, the corresponding edge $e_{j1,j2}$ will be added to the blacklist $\mathcal{L}$.

### 4.3.2 Inter-slice Configuration

In the DBN, as the inter-slice edges are relative to the impact given by the previous run, any edges from the later run to the previous run are declared to be impossible. Therefore, $\forall e_{(j1[t],j2[t-1])} = (X_{j1}^{(t)}, X_{j2}^{(t-1)})$ is added to the blacklist $\mathcal{L}$. In the opposite direction, the edges from a previous run to the current run, $e_{(j1[t-1],j2[t])} = (X_{j1}^{(t-1)}, X_{j2}^{(t)})$, may be possible and will be inferred from data. Besides these relationships, there may be some additional edges to be handled carefully for specific process systems. For example, some FDC variables are deterministic variables and are not affected by other variables from a previous run, such as counter variables. Moreover, considering processes involving multiple chambers, any edge $e_{(j1[t-1],j2[t])} = (X_{j1}^{(t-1)}, X_{j2}^{(t)})$, where $X_{j1}^{(t-1)}$ and $X_{j2}^{(t)}$ are collected from different chambers, will be defined as illegal as well.

Given the above configuration, a blacklist $\mathcal{L}$ is ready to be used in DBN structure learning. Note that a whitelist is able to be included as well, if there are some causalities have

been confirmed. In this work, only blacklist will be considered and other possible edges will be determined by data.

## 4.4 Offline Stage - Structure Learning

The goal of the section is to learn a two-time-slices DBN based on the data $\mathcal{D}$ and the blacklist $\mathcal{L}$. As described in Section 4.4, the intra-slice edges present the interactions between variables. We assume that these interactions will not change in a short period, which is the time between two processing wafers. In this context, the learned DBN should have the same intra-slice structure. As explained in Chapter 2, the learning procedure starts by learning an intra-slice structure. After the intra-slice structure has learned, learning inter-slice edges can be considered as feature selection (Murphy, 2002) (see Fig. 4.7).

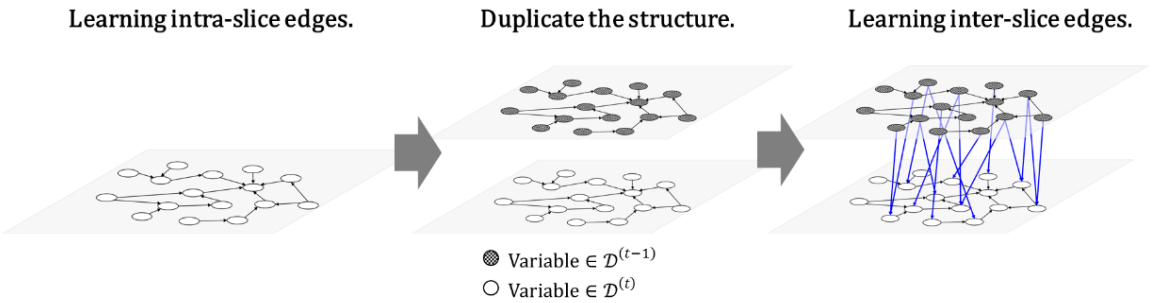


Figure 4.7 The learning procedure of a DBN.

However, the inter-slice edges may affect the learned intra-slice structure. The effect from the previous time slice and the effect from the same slice should be considered at the same time. Therefore, a new approach to learning a two-time-slice DBN is proposed in this framework. The structure learning procedure involves two phases, learning and repairing (see Fig. 4.8). Based on data D and blacklist L, The first learning phase aims to learn a BN regardless of the time factor. Then, by relocating these nodes into its time slice, the initial DBN is determined. The second phase will repair this structure to make sure the intra-slice edges are the same. The details of this procedure will be introduced in the following subsections.

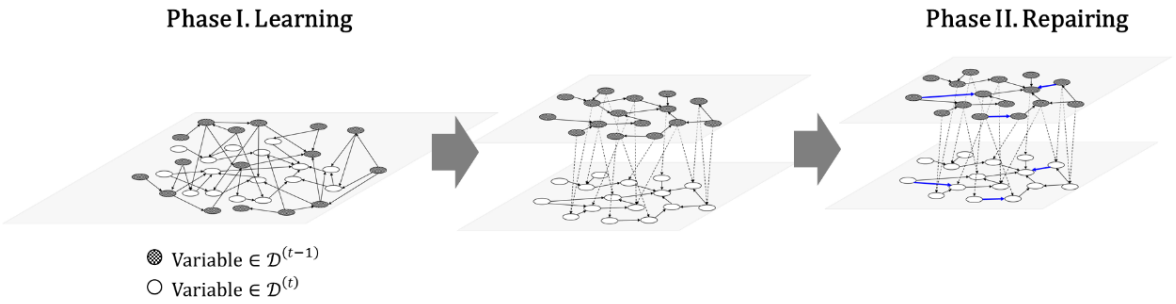


Figure 4.8 The proposed structure learning procedure.

### 4.4.1 Two-Phase DBN Structure Learning

As shown in Fig. 4.8, given the data $\mathcal{D}$ and the blacklist $\mathcal{L}$, the goal of the first learning phase is to learn an initial structure of BN $\mathcal{G}_{ini}(\mathcal{D})$ regardless of the time factor. The BN $\mathcal{G}_{ini}$ can be presented in a DBN form, where the nodes are relocated into different time slices. Essentially,

both inter-slice edges and intra-slice edges are learned simultaneously, but the intra-slice structures of the two-time slices may be different. Therefore, the repairing procedure is required later. As mentioned in Chapter 2, some search algorithms are available for learning structure. Since the objective of this framework is to demonstrate the capability of DBN for process control, we will not focus on addressing the efficiency of different search algorithms. In this work, Hill-Climbing is chosen as a search algorithm.

After the first learning phase, an initial DBN $\mathcal{G}_{ini}$ is obtained. Since the $\mathcal{G}_{ini}$ consists of two-time-slices structures, two partial structures can be extracted denoted as $\mathcal{G}_{ini}^{(t-1)} = (\mathbb{V}^{(t-1)}, \mathbb{E}_{ini}^{(t-1)})$ and $\mathcal{G}_{ini}^{(t)} = (\mathbb{V}^{(t)}, \mathbb{E}_{ini}^{(t)})$, where $\mathcal{G}_{ini}^{(t-1)}$ is the intra-slice structure of $\mathcal{G}_{ini}$ at time slice $t-1$ and $\mathcal{G}_{ini}^{(t)}$ is another intra-slice structure of $\mathcal{G}_{ini}$ at time slice $t$. As mentioned in the previous subsection, the structure of different time slice can be different. Therefore, comparing $\mathcal{G}_{ini}^{(t-1)}$ and $\mathcal{G}_{ini}^{(t)}$ it is possible to obtain the set of inconsistent edges, $\mathbb{E}_{diff} = \{\left(e_{i,j}^{(t-1)}, e_{i,j}^{(t)}\right), \forall e_{i,j}^{t-1} \neq e_{i,j}^{(t)}$, where $e_{i,j}^{(t-1)} \in \mathbb{E}_{ini}^{(t-1)}$ and $e_{i,j}^{(t)} \in \mathbb{E}_{ini}^{(t)}\}$ . The possible inconsistent pair edges are defined below:

- $e_{i,j}^{(t-1)} = \emptyset$ and $e_{i,j}^{(t)} \neq \emptyset$
- $e_{i,j}^{(t-1)} \neq \emptyset$ and $e_{i,j}^{(t)} = \emptyset$
- $e_{i,j}^{(t-1)} \neq \emptyset$, $e_{i,j}^{(t)} \neq \emptyset$ and $e_{i,j}^{(t-1)} = e_{j,i}^{(t)}$

In order to fix these inconsistencies, the second part of the learning procedure is the repairing phase, where the goal is to establish the same intra-slices structure for the two time slices. Assume $\mathcal{G}_{repair}$ is a repaired graph, which will be updated by synchronizing the inconsistent pair edges in $\mathbb{E}_{diff}$, the repairing procedure will stop when the two intra-slice structures are the same. Finally, the desired two-time-slice DBN $\mathcal{G} = \mathcal{G}_{repair}$ is obtained. The overall structure learning procedure is presented in Algorithm 4.1.

As Hill-Climbing is a greedy search algorithm, the global optimum will not be easily reached. To overcome this limitation, a random restart step was introduced in order to uniformly cover the search space. A random restart step is to explore other possibilities given the current best network. The number of perturbed edges should be specified as well, which indicates the number of edges that will be randomly changed given the best network at the current moment.

The optimal number of random-restarts and their corresponding perturbed edges are defined as hyperparameters and can be empirically determined at a preliminary experiment. This experiment is based on a grid search approach procedure. Let $h_1$ be the number of restarts and $h_2$ be the number of perturbed edges. Given a set of possible values of $h_1$ and $h_2$, $h_1 \in \{h_{1,1} \quad h_{1,2} \quad \cdots \quad h_{1,\phi_1}\}$ and $h_2 \in \{h_{2,1} \quad h_{2,2} \quad \cdots \quad h_{2,\phi_2}\}$. For each combination $(h_1, h_2)$, learn a DBN and denoted as $\mathcal{G}_{h_1,h_2}$, and obtain its score $Score(\mathcal{G}_{h_1,h_2}, \mathcal{D}_1)$. There are $\phi_1 \times \phi_2$ DBNs, let $\mathcal{G}_{best}$ be the DBN with the best score in this experiment.

**Algorithm 4.1** Two-phase DBN Structure Learning

**Phase I. Learning**

$\mathcal{G}_{ini}$ ← Initial BN regardless time slices using score-based learning algorithm.

$\mathcal{G}_{ini}^{(t-1)}$ ← Partial network of $\mathcal{G}_{ini}$, where nodes are at $t-1$.

$\mathcal{G}_{ini}^{(t)}$ ← Partial network of $\mathcal{G}_{ini}$, where nodes are at $t$.

$\mathbb{E}_{diff}$← A set of pair inconsistent edges.

**Phase II. Repairing**

$\mathcal{G}_{repair} \leftarrow \mathcal{G}_{ini}$

$\bar{e}_{i,j}$ ← edges in $\mathcal{G}_{repair}$

**for** each pair edges in $\mathbb{E}_{diff}$:

$\mathcal{G}_1$←set_edge($\mathcal{G}_{repair}, \bar{e}_{i,j}^{(t)} = e_{i,j}^{(t-1)}$)

$\mathcal{G}_2$←set_edge($\mathcal{G}_{repair}, \bar{e}_{i,j}^{(t-1)} = e_{i,j}^{(t)}$)

**if** score($\mathcal{G}_1$)>score($\mathcal{G}_2$)

$\mathcal{G}_{repair} \leftarrow \mathcal{G}_1$

**else**

$\mathcal{G}_{repair} \leftarrow \mathcal{G}_2$

**end if**

**end for**

$\mathcal{G} \leftarrow \mathcal{G}_{repair}$

A Relative Score (RS) of each combination is used to measure the distance from $\mathcal{G}_{best}$, which is defined as

$$RS_{h_1,h_2}(\%) = \frac{|Score(\mathcal{G}_{h_1,h_2}, \mathcal{D}_1) - Score(\mathcal{G}_{best}, \mathcal{D}_1)|}{Score(\mathcal{G}_{best}, \mathcal{D}_1)} \times 100.$$

The small value of $RS$ of a DBN indicates its score is close the best DBN. Based on this experiment, the optimal hyperparameters can be determined. The decision can be made either by choosing the settings with best score, or the settings with lower computation cost. After the experiment, the final network with specified hyperparameters has been determined, denoted as $\mathcal{G} = \mathcal{G}_{h_1,h_2}$.

### 4.4.2 Representation

A DBN illustrates the dependencies between variables in a connected graph form, and a random variable is presented as a node (or vertex) in the network. Figure 4.9 illustrates an example of resulting DBN, $\mathcal{G} = (\mathbb{V}, \mathbb{E})$, not only indicating how controllable variables are causally related to FDC features, but also identifying the direct and indirect impact in the metrology variable.

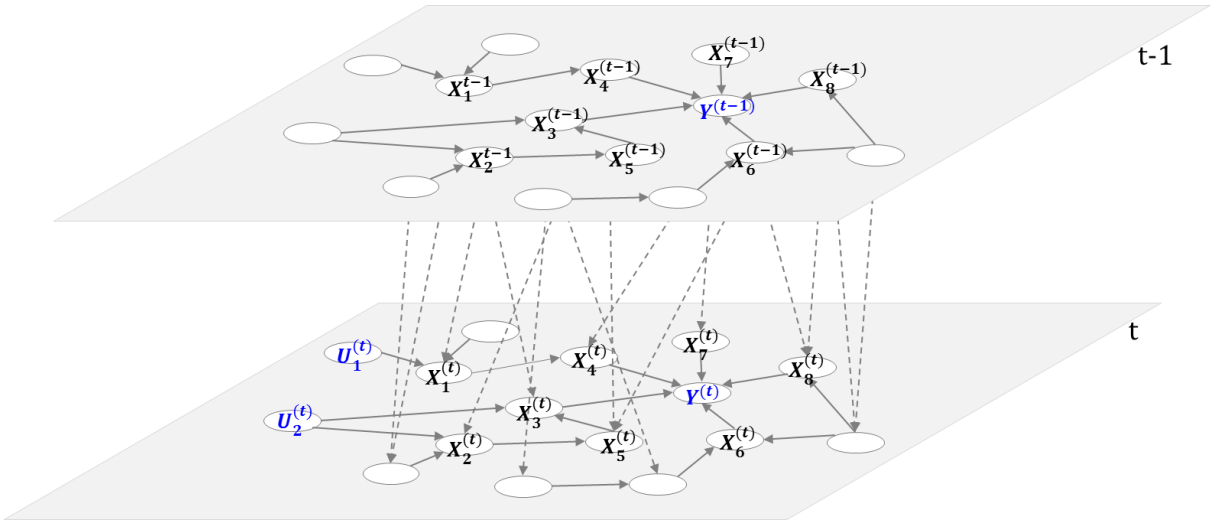


Figure 4.9 An example of a DBN.

In order to simplify the representation, in what follows a mapping notation will be used so that the properties of the graphic model can be more clearly demonstrated (see Table 4.2). Some of this notation will be referred in the following sections.

Table 4.2 A mapping notation table.

| Notation in $\mathcal{G}$ | Definition |
|---|---|
| $\mathbb{V}$ | All variables (nodes) in $\mathcal{D}$. |
| $\mathbb{V}_s^{(t)} = \{U_1^{(t)}, \dots, U_k^{(t)}\}$ | A set of source variables (nodes) which includes controllable variables at time $t$. |
| $\mathbb{V}_e^{(t)} = \{Y^{(t)}\}$ | A sink variable (node) which indicates the metrology variable at time $t$. |
| $\mathbb{V}_{br}^{(t)}$ | A set of bridge variables (nodes) which connect source nodes and sink node. |
| $\mathbb{V}_{sp}$ | A set of supporting variables (nodes). |
| $\mathbb{V}_{pa(V)}$ | A set of parent variables of variable $V$, $\forall\, V \in \mathbb{V}$. |
| $\mathbb{V}^{(t)}$ | Variables at time $t$. |
| $\mathbb{V}^{(t-1)}$ | Variables at time $t$-1. |

As referred in Section 2.1, a node $X_i$ and its parent nodes $\mathbb{V}_{pa(X_i)}$ can be expressed as a linear regression model. Thus, for each FDC variable in $\mathcal{G}$, there is a linear function $f_{X_i}: \mathbb{V}_{pa(X_i)} \rightarrow X_i$, where $X_i$ is depend variable and $X \in \mathbb{V}_{pa(X_i)}$ are independent variables;

similarly, there is a function $f_y$: $\mathbb{V}_{pa(y)} \rightarrow Y$. Thus, the relationships involved in a DBN, $\mathcal{G}$ ,can be presented by a set of functions, $\mathbb{f} = \{f_{X_1}, \dots, f_{X_r}, f_Y\}$.

$$X_i = f_{X_i}(\mathbb{V}_{pa(X_i)}),\ \forall\, i = 1, \dots, r \tag{4.3}$$

For each function in $\mathbb{f}$, the independent variables can be further classified by time-slices as shown in 4.4. In this way, the variation of any variable in DBN can be decomposed into the effect arising from the previous run and the effect of the current run. In the Section 4.7.2, we will explain how to use this property to classify unknown and known information.

$$\mathbb{V}_{pa(X_i)} = \mathbb{V}_{pa(X_i)}^{(t-1)} \cup \mathbb{V}_{pa(X_i)}^{(t)} \tag{4.4}$$

## 4.5 Online Stage - Monitoring

Generally, SPC charts of the metrology data are used to monitor the quality of wafer and the overall stability of process. The SPC charts of major FDC indicators are used to monitor the condition of process equipment. Monitoring those control charts separately can be inefficient, and it implies substantial costs. Thus, many studies proposed using the summarized index to simplify the monitoring task. For example, employing Principal Component Analysis (PCA) can reduce dimensions into a few important components (Wise & Gallagher, 1996; Cherry & Qin, 2006). However, for further diagnosis, the additional method is necessary for tracing back the original feature. Considering the primary goal of monitoring is to detect the fraud and irregularity during the process, a DBN can help to streamline the work. Since a DBN is learned from historical data based on likelihood, this allows us to evaluate the likelihood of future data by the same fashion.

### 4.5.1 Anomaly Detection

As referred in Section 3.4.1, the Bayesian information criterion (BIC) is used to evaluate the possible network structure, $Score_{BIC}(\mathcal{G}, \mathcal{D}) = \log\left(\hat{L}\right) - \frac{1}{2} d_{\mathcal{G}} \log n$ , where $\hat{L} = p(\mathcal{D}|\mathcal{G}, \hat{\theta}_{\mathcal{G}})$ is the maximum value of likelihood function, $\hat{\theta}_{\mathcal{G}}$ is the maximum likelihood estimate. The idea of likelihood function $L$ is how likely the parameters $\theta_{\mathcal{G}}$ are for a given dataset $\mathcal{D}$, which is also equal to the joint probability density function of the random variables. Instead of considering all possible parameters setting, the BIC simply uses maximum likelihood estimation to compute the likelihood.

Since the joint distribution of $\mathcal{G}$ can be decomposed into the local distribution of individual variables, the likelihood function can be expressed as the product of local likelihood, $L(\theta_{\mathcal{G}}: \mathcal{D}) = \prod_{k=1}^{q} p(X_k | \mathbb{X}_{pa(k)}: \hat{\theta}_{X_k|pa(k)}) = \prod_{k=1}^{q} L(\mathcal{D}: \hat{\theta}_{X_k|pa(k)})$. Without loss of generality, usually the natural logarithm of the likelihood function is applied:

$$l(\theta_{\mathcal{G}}: \mathcal{D}) = \ln\left(L(\theta_{\mathcal{G}}: \mathcal{D})\right) = \sum_{k=1}^{q} l(\mathcal{D}: \hat{\theta}_{X_k|pa(k)}), \tag{4.5}$$

where $l(\mathcal{D}: \hat{\theta}_{X_k|pa(k)})$ is the local log likelihood function. As we assume that all the variables in the DBN follow Gaussian distribution, the local likelihood of each variable is defined as $L(\hat{\mu}_k, \hat{\sigma}_k; x_{1,k}, \dots, x_{n,k}) = (2\pi\hat{\sigma}_k^2)^{-\frac{n}{2}} \exp\left(-\frac{1}{2\hat{\sigma}_k}\sum_{j=1}^{n}(x_{jk} - \hat{\mu}_k)^2\right)$, and its log likelihood is defined as below,

$$l(\hat{\mu}_k, \hat{\sigma}_k; x_{1,k}, \dots, x_{n,k}) = -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\hat{\sigma}_k^2) - \frac{1}{2\hat{\sigma}_k}\sum_{j=1}^{n}(x_{jk} - \hat{\mu}_k)^2, \tag{4.6}$$

where $\hat{\mu}_k$ and $\hat{\sigma}_k$ are the maximum likelihood estimates. For those variables which have parents, the parameters of the conditional distribution of $X_k|\mathbb{X}_{pa(k)}$ will be replaced with $\mu_{k|pa(k)}$ and $\sigma_{k|pa(k)}$. An example of a simple network $\mathcal{G}$ is shown in Fig. 4.10. Assume that $\mathcal{G}$ is learned from an $n$ by three matrix $\mathcal{D} = [X_1 \quad X_2 \quad X_3]$. The joint distribution of $\mathcal{G}$ can be decomposed into two univariate distribution of the root nodes, and the conditional distribution of the child node. As described in chapter 2, any node can be expressed as a linear regression model involving its causal parents. Thus, a regression model can be derived from $\mathcal{G}$, $X_3 = [X_1 \quad X_2]\beta_{LR} + \varepsilon$, where $\varepsilon = (\varepsilon_1 \quad \cdots \quad \varepsilon_n)$ is the error term. The conditional distribution of $X_3|X_1, X_2$ is equal to the distribution of the error term.

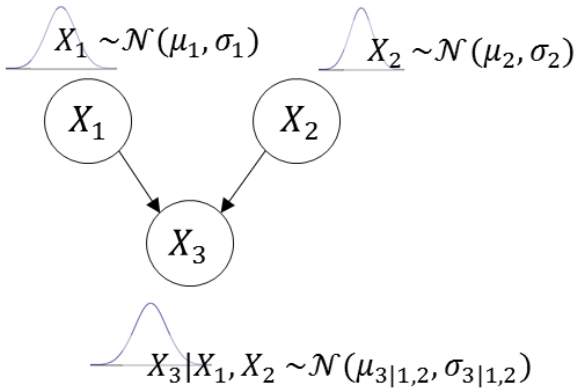


Figure 4.10 An example of a network and its local distribution.

As shown in the previous chapter, the historical data $\mathcal{D}$ is used to learn the structure $\mathcal{G}$ and the corresponding parameters $\hat{\theta}_{\mathcal{G}}$. This structure $\mathcal{G}$ then will be used for online applications. Considering a new data point $d_{n+1}$, i.e., a new wafer, its individual log likelihood can be computed based on equation 4.1, only the data $\mathcal{D}$ is replaced by $d_{n+1}$,

$$l_{n+1} = l(\hat{\theta}_{\mathcal{G}}: d_{n+1}). \tag{4.7}$$

We define this individual log likelihood as *Global Likelihood Index* (GLI), which can be used to evaluate if the new data point is similar to the historical data. Note that this index is employed under the assumption that the relationships among variables are expressed well by $\mathcal{G}$. As described in equation 4.5, this log likelihood is computed by the sum of local log likelihood. Thus, the local log likelihood of variable $X_k$ of new data $d_{n+1}$, is defined as *Local Likelihood Index* (LLI),

$$\mathrm{LLI}(X_k, d_{n+1}) = l\left(\hat{\mu}_k, \hat{\sigma}_k; x_{n+1,k}\right) \tag{4.8}$$
$$= -\frac{n}{2}\ln(2\pi) - \frac{n}{2}\ln(\hat{\sigma}_k^2) - \frac{1}{2\hat{\sigma}_k}\left(x_{n+1,k} - \hat{\mu}_k\right)^2.$$

With both GLI and LLI, the performance of new wafers can be easily monitored. The GLI firstly provides an overall evaluation. A high value of GLI indicates that this new wafer is similar to the historical wafers. If GLI is low, there might be some abnormality present. In this case, LLIs can help to quickly identify the locations of the abnormalities. If the LLI of $X_k$ is low, it means that the value is far from the center of its distribution. An example is illustrated in Fig. 4.11. If all the values are close to the center of its distribution, high GLI and high LLIs will be obtained (see Fig. 4.11a). If there are some extreme values occurred, then some of LLIs will be lower than usual and GLI as well (see Fig. 4.11b).

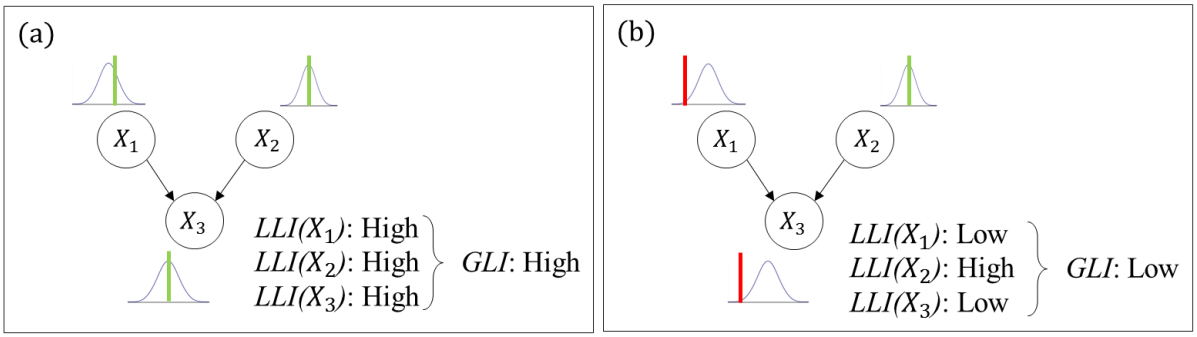


Figure 4.11 Figure 4.11 (a) The GLI and the LLIs of this wafer are high; (b) the GLI of this wafer is low, and some of LLIs are low.

To establish the appropriate control limits for anomaly detection, the kernel density estimation (Silverman, 1986), a non-parametric method, is employed to estimate the probability density function of each LLI variable. The kernel density estimation of $\mathrm{LLI}(X_k)$ is defined as

$$\hat{f}_{KDE}(\ell) = \frac{1}{nh}\sum_{j=1}^{n} \mathcal{K}\left(\frac{\ell - LLI(X_k, d_j)}{h}\right),$$

where $\mathcal{K}$ is the kernel function, and $h$ is bandwidth. The control limits can be determined by $\int_{-\infty}^{\mathrm{LLI}_{UCL}} \hat{f}_{KDE}(\ell)dx = 1 - \frac{\alpha_I}{2}$ and $\int_{-\infty}^{\mathrm{LLI}_{LCL}} \hat{f}_{KDE}(\ell)dx = \frac{\alpha_I}{2}$, where $\alpha_I$ is the predefined Type I error.

## 4.5.2 Model Update Mechanism

If the anomaly is just a random case, then this wafer should be tagged as a risky wafer and trigger an additional inspection. If this anomaly is sustained to several wafers, the reason can be that the equipment conditions have changed and the current model no longer applies. To clarify these two possibilities, the monitoring loop incorporates a model updating mechanism (see Fig. 4.12). However, the update mechanism is not always suitable considering the magnitude of the drifting behavior. Large scale drifting may require a more urgent action, such as maintenance. And updating model without consider the critical level may mask the crucial irregularity and increase the potential risk of the process. Therefore, besides the model update

mechanism, another control system of measuring the critical level should be considered in practice. In this thesis, we aim to demonstrate the possibility of incorporating a monitoring mechanism. We start with a simple approach which assumes that the model update mechanism is operated under minor drift. In the rest of manuscript, the risk of model update will not be considered, but more discussion can be found in Section 4.8 and Chapter 6.

Assume wafer $w$ has been processed, and $d_w$ is the data of this wafer. For each variable $X_k$, the likelihood LLI($X_k, d_w$) will be computed. Compare to its $\text{LLI}_{UCL}(X_k)$ and $\text{LLI}_{LCL}$, if the value is out of control limit, the alarm $\mathcal{F}_{(X_k)}$will be triggered, and the number of alarm $a_{(X_k)}$ will be set to 1. Assume the length of alarm window is $\mathcal{S}$ wafers. The following $\mathcal{S}-1$ wafers will be treated as suspicious wafers, if LLI of one of these wafers exceeds the control limit, $a_{(X_k)}$ increases by 1. After processing those suspicious wafers, the risk level denoted as $h_k = a_{(X_k)}/\mathcal{S}$ will be examined to see if it is more than a specified threshold, $h_{risk}$. If $h_k < h_{risk}$, then the alarm can be relived. If $h_k > h_{risk}$, the local update model will be activated.

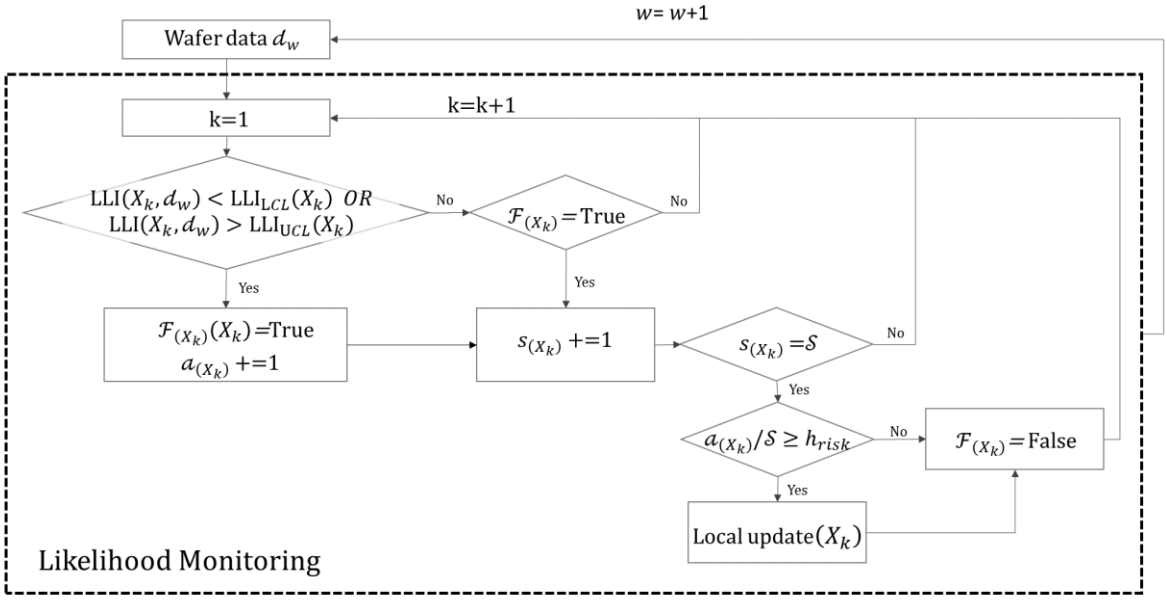


Figure 4.12 The flow chart of the likelihood monitoring mechanism.

The local update is defined as only updating the coefficients of variable $X_k$ based on the latest $\mathcal{S}$ wafers. The local update is feasible because of the conditional independent property of Bayesian Network. The regression model of a node can be learned separately from other nodes, which is one of the advantages of the Bayesian network. Especially when the change of equipment conditions is observed at the early stage, usually the sample size is limited. In this case, instead of using a small sample to update the whole model, update part of parameters should be a relatively low risk approach.

## 4.6 Online Stage - Prognosis

As described in Section 4.4, for each FDC variable in $\mathcal{G}$, there is a linear function $f_{X_k}: \mathbb{V}_{pa(X_k)} \to X_k$, where $X_k$ is dependent variable and $X \in \mathbb{V}_{pa(X_k)}$ are independent variables; similarly, there is a function $f_y: \mathbb{V}_{pa(y)} \to Y$. Each function can be used for online

prediction given new dataset. In this section, two function modules of prognosis will be introduced, a VM model and equipment condition prediction procedure. The latter will be used for the R2R control in Section 4.8.

### 4.6.1 Virtual Metrology

Given the fact that the metrology data of most wafers are usually unavailable, the VM model becomes one of the most common prognosis models. A VM model considers the metrology variable $Y^{(t)}$ as the target, and the output of the model will be the predictive value of the target. In a DBN, a target variable can be expressed by its parent nodes through a linear regression model. Therefore, a VM model can be easily extracted from a DBN without additional learning, i.e., $f_y: \mathbb{V}_{pa(y)} \rightarrow Y$. Two examples are shown in Fig. 4.13. The sub-network which consists of several blue nodes is the VM model.

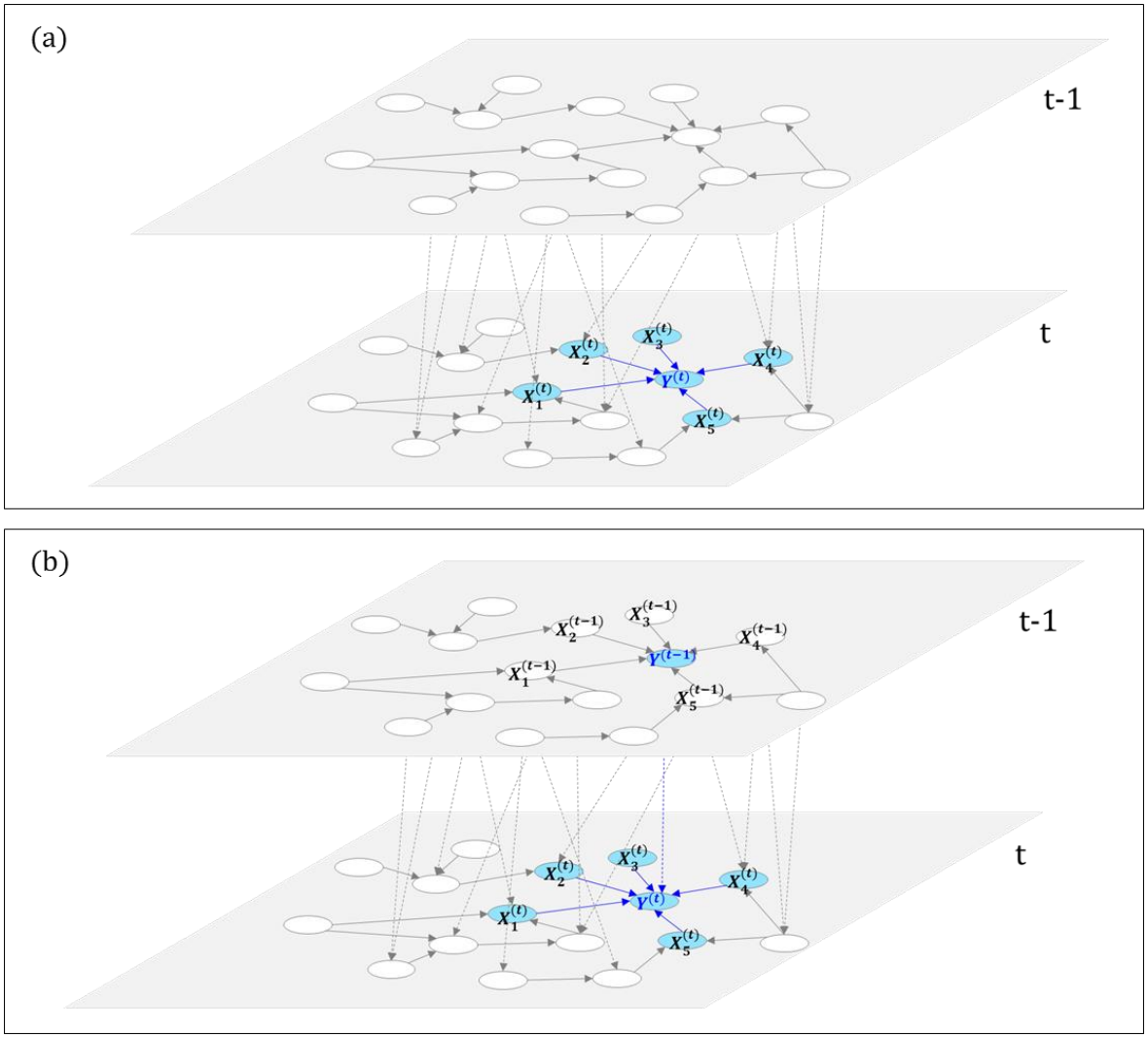


Figure 4.13 Examples of VM model.(a) Parents of $Y^{(t)}$ are located at the same time slice; (b) the metrology variable $Y^{(t)}$ is affected by the metrology of the previous wafer.

The objective of the VM model is to provide the predicted metrology after processing a wafer. Considering a real-time production line and assuming run *w* is completed, the data

vector of this wafer is denoted as a 1 by $q$ vector $\mathcal{d}_w = \left[\mathcal{d}_w^{(t-1)} \quad \mathcal{d}_w^{(t)}\right]$, where $\mathcal{d}_w^{(t-1)} = \left[u_{w,1}^{(t-1)} \quad \cdots \quad u_{w,k}^{(t-1)} \quad x_{w,1}^{(t-1)} \quad \cdots \quad x_{w,r}^{(t-1)} \quad y_w^{(t-1)}\right]$ and $\mathcal{d}_w^{(t)} = \left[u_{w.1}^{(t)} \quad \cdots \quad u_{w.k}^{(t)} \quad x_{w.1}^{(t)} \quad \cdots \quad x_{w.r}^{(t)} \quad y_w^{(t)}\right]$. Note that $y_w^{(t)}$ is a null value and $y_w^{(t-1)}$ can be known or unknown value which depends on if the previous wafer is sampled wafer.

If all $\mathbb{V}_{pa(Y^{(t)})}$ in $\mathcal{d}_w$ are known, the predictive value $\hat{y}_w^{(t)} = f_Y\left(\mathbb{V}_{pa(Y^{(t)})}, \mathcal{d}_w\right)$ can be immediately computed (see Fig. 4.13a). Considering another scenario shown in Fig. 4.13b. Assume that $Y^{(t-1)}$ is one of the parents of $Y^{(t)}$; therefore, sometimes $Y^{(t-1)}$ is unknown when the actual metrology of the previous wafer is not available. In this case, another step is required to obtain the value of $Y^{(t-1)}$. Two possible approaches are listed below:

- $\hat{y}_w^{(t-1)} = \bar{Y}^{(t-1)}$, where $\bar{Y}^{(t-1)}$ is the mean level obtained from historical data.
- $\hat{y}_w^{(t-1)} = f_Y\left(\mathbb{V}_{pa(Y^{(t)})}, \mathcal{d}_{w-1}\right)$.

The second approach requires $y^{(t-2)}$ is known. If $y^{(t-2)}$ is unknown, then another prediction procedure will be carried out in order to get $\hat{y}^{(t-2)}$. This procedure will trace back until it reaches the latest actual measurement. Since the second approach involves several iterations which can increase the computation loading, we will consider the first approach to deal with such a case.

### 4.6.2 Equipment Condition Inference

A VM model only focuses on a target metrology variable, while the prognosis procedure can be applied to more than one unknown variable in a DBN. Considering a real-time production line and assuming run $w-1$ to be completed, we would like to predict the equipment condition for the incoming run. Since FDC variables implicitly provide information about equipment condition, this implies that the objective is to predict the FDC variables of run $w$. The latest information for the incoming run (wafer) $w$ can be expressed by a 1 by $q$ vector $\mathcal{d}_w = \left[\mathcal{d}_w^{(t-1)} \quad \mathcal{d}_w^{(t)}\right]$, where $\mathcal{d}_w^{(t-1)}$ is an non-empty vector that represents the available data of the processed wafer, and $\mathcal{d}_w^{(t)}$ is an empty vector that indicates unknown information of the incoming wafer.

An iterative procedure will be conducted to give the predictions for variables in $\mathcal{d}_w^{(t)}$. The corresponding pseudocode is shown in Algorithm 4.2. The initial input includes an initial vector $\mathcal{d}_w = \left[\mathcal{d}_w^{(t-1)} \quad \mathcal{d}_w^{(t)}\right]$ wherein each element in $\mathcal{d}_w^{(t)}$ is null. A permutation list of nodes $L_{ts}$ obtained by topological sorting and the set of functions $\mathbb{f}$ represents a list of linear regression models are denoted in the pseudocode. The procedure begins with visiting the first nodes in $L_{ts}$ during the first iteration. Note that $V_{visiting}$ represents the visiting nodes in the current iteration and it will be updated at each iteration. If the element in the vector $\mathcal{d}_w$ which presents the value of $V_{visiting}$ is null, the corresponding linear regression model of $V_{visiting}$ will be used to provide a predicted value; otherwise, the prediction step will be skipped and the procedure moves to the next iteration. The following iterations proceed in the same way as the

first iteration. After running the iterative procedure, any missing values in $\mathcal{d}_w$ will be patched by the predictive ones.

**Algorithm 4.2** Predict unknown variables

1. $\mathcal{d}_w \leftarrow$ Initial vector represents current information
2. $L_{ts} \leftarrow$ A list of variables (nodes) by topological sorting
3. $\mathbb{f} \leftarrow$ a list of regression model for variables (nodes)
4. **for** each variable $V$ in $L_{ts}$:
5. $\quad V_{visiting} \leftarrow V$
6. $\quad$ idx $\leftarrow$ IndexOfVar($\mathcal{d}_w$, $V_{visiting}$)
7. $\quad$ **if** $\mathcal{d}_w$[idx] is null
8. $\quad\quad$ lm $\leftarrow \mathbb{f}$ [idx]
9. $\quad\quad \mathcal{d}_w$[idx] $\leftarrow$ lm(ParentOfVar($V_{visiting}$))
10. $\quad$ **end if**
11. **end for**

An example of the procedure is illustrated in Fig. 4.14, note that the example presents a DBN in two dimension for simplifying the demonstration. A DBN $\mathcal{G}$ with 8 nodes is given, where nodes marked in gray indicate their corresponding values in $\mathcal{d}_w$ are missing, i.e., nodes are located at time slice $t$ (see Fig. 4.14a). A permutation list of $\mathcal{G}$ is $L_{ts} = \{V_5 \;\; V_6 \;\; V_7 \;\; V_8 \;\; V_4 \;\; V_3 \;\; V_2 \;\; V_1\}$. The first iteration starts with the visiting node $V_5$ (see Fig. 4.14b). As the value is not null, the prediction step will be skipped and moved to the second iteration. For the fifth iteration, as the value of visiting node $V_4$ is missing, a prediction step will be triggered and a predicted value is computed. For the sixth iteration, the parent nodes of $V_3$, i.e., $V_4$ and $V_7$, are now known and thus the value of $V_3$ can be predicted by its parent nodes. Since $L_{ts}$ obtained by topological sorting guarantees that all parents of $V_{visiting}$ have been visited, i.e., the values of parent nodes will not be null.

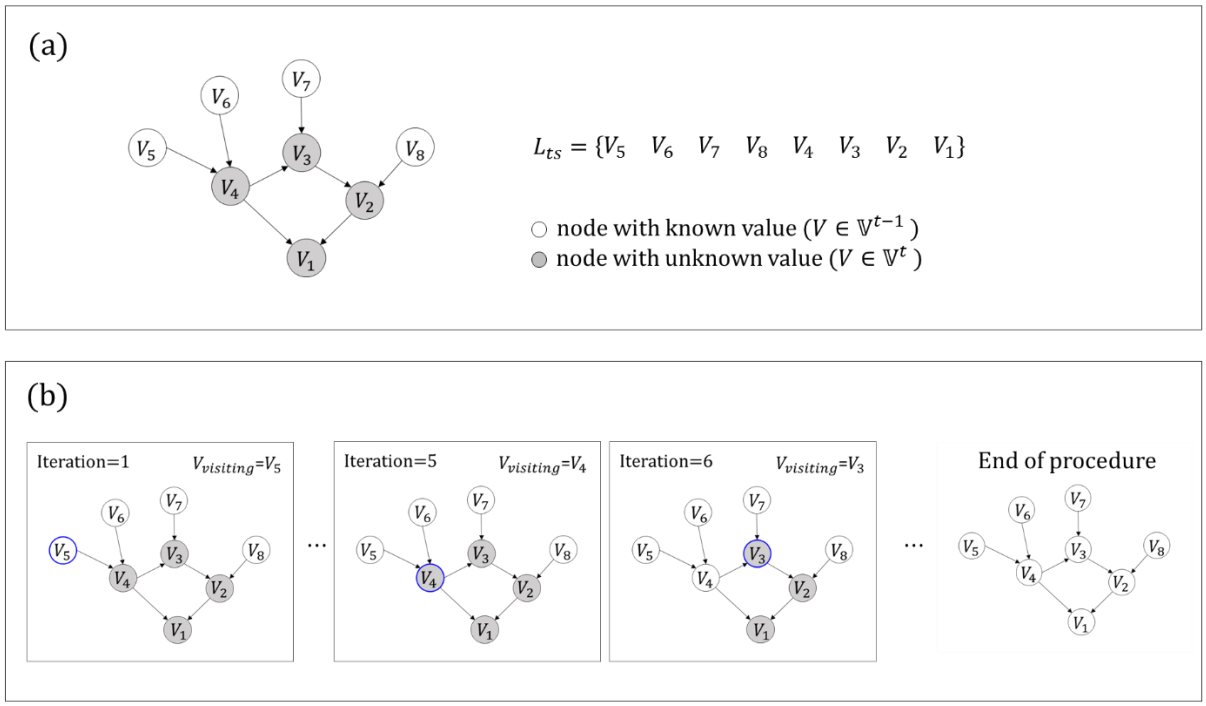


Figure 4.14 An example of the proposed prediction procedure, starting from (a) a learned DBN, $\mathcal{G}$, to (b) the predictive propagation of the nodes with unknown values.

Through the above procedure, the predictive equipment conditions can be obtained. Having such information enables more control actions to be carried out in advance. For example, a R2R control action can be made by leveraging more information about equipment; the approach will be explained in the next section. Moreover, this information may enhance the efficiency of predictive maintenance. Although predictive maintenance is not in the thesis scope, it can be an interesting subject for future investigation.

## 4.7 Online Stage - Structured R2R Controller

Conventionally, a R2R controller involves one or more controllable variables. By changing the settings of controllable variables, process conditions would change accordingly and affect the process output, i.e., the metrology. With the technique of DBN, the impact of controllable variables on the metrology measurements can be disclosed, and the R2R regulators can be designed more reasonably. In the following, the details of a Structured R2R Controller (SRC) based on DBN will be discussed.

### 4.7.1 Sub-network Identification

Although DBN can show all the variable relationships in the process system, only a sub-network is required for implementing the proposed SRC.

To specify the necessary sub-network $\mathcal{G}' = (\mathbb{V}', \mathbb{E}')$, the controllable variables $\mathbb{V}_s^{(t)}$ and metrology variable $\mathbb{V}_e^{(t)}$ will be added into $\mathbb{V}'$. Considering controllable variables as source nodes and the metrology variable as the sink node, all the paths from source nodes to the sink node are identified. The nodes along the path are denoted as the bridge nodes,

$$\mathbb{V}_{br}^{(t)} = \left\{\forall\, V, \text{where } (V_s \prec V) \wedge (V \prec V_e) \text{ and } V_s \in \mathbb{V}_s^{(t)},\ V_e \in \mathbb{V}_e^{(t)},\ V \in \mathbb{V}\right\}, \tag{4.9}$$

Furthermore, as bridge nodes and sink node are not only affected by source nodes, they also depend on their other parent nodes. Therefore, all the parent nodes of bridge nodes and all the parent nodes of the sink node will be taken into account, denoted as supporting variables,

$$\mathbb{V}_{sp} = \{\forall\, \mathrm{V} \in (\mathbb{V}_{pa(V_{br})} \cup\ \mathbb{V}_{pa(V_e)}), \text{where } V_{br} \in \mathbb{V}_{br}^{(t)},\ V_e \in \mathbb{V}_e^{(t)}, \mathrm{V} \in \mathbb{V},\ \mathrm{V} \notin \mathbb{V}_{br}^{(t)}\}. \tag{4.10}$$

As mentioned in (4.4), $\mathbb{V}_{pa(V)}$ can be decomposed into $\mathbb{V}_{pa(V)}^{(t-1)}$ and $V_{pa(V)}^{(t)}$, the set of supporting variables can be expressed as the union of the two subsets,

$$\mathbb{V}_{sp}^{(t-1)} = \left\{\mathrm{V}, \text{where } \mathrm{V}\ \in \mathbb{V}_{sp} \text{ and } \mathrm{V} \in \mathbb{V}^{(t-1)}\right\}$$

$$\mathbb{V}_{sp}^{(t)} = \left\{\mathrm{V}, \text{where } \mathrm{V}\ \in \mathbb{V}_{sp} \text{ and } \mathrm{V} \in \mathbb{V}^{(t)}\right\}$$

$$\mathbb{V}_{sp} = \mathbb{V}_{sp}^{(t-1)} \cup \mathbb{V}_{sp}^{(t-1)}. \tag{4.11}$$

In summary, the sub-network $\mathcal{G}' = (\mathbb{V}', \mathbb{E}')$, has been well defined (see Fig. 4.15), where $\mathbb{V}' = \left\{\mathbb{V}_s^{(t)} \cup \mathbb{V}_e^{(t)} \cup \mathbb{V}_{br}^{(t)} \cup \mathbb{V}_{sp}\right\}$. The set of functions corresponding to $\mathcal{G}'$ is expressed as

$$\mathbb{f}' = \{f_V, \forall\, V \in (\mathbb{V}_{br}^{(t)} \cup \mathbb{V}_e)\}. \tag{4.12}$$

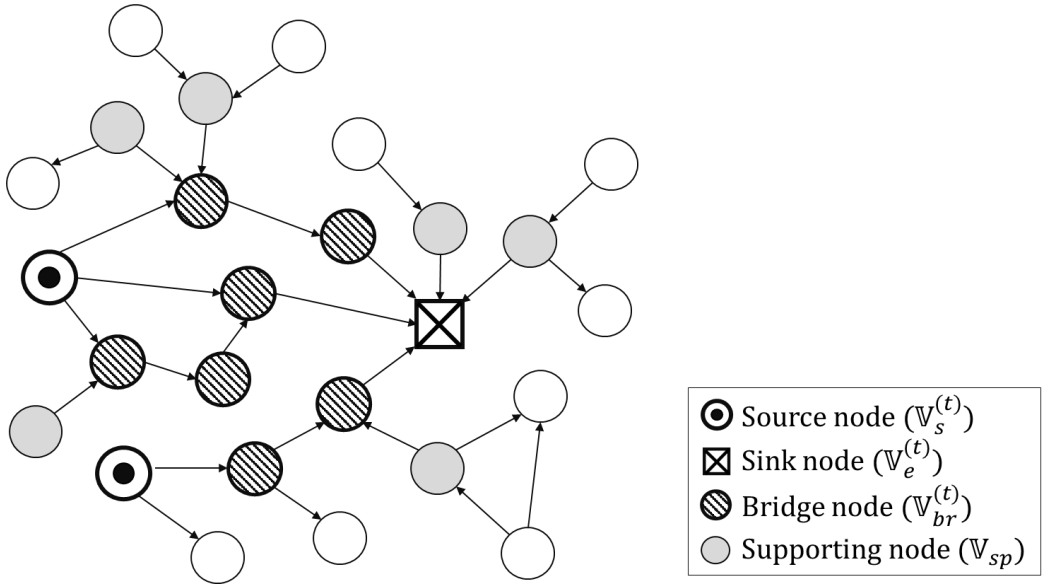


Figure 4.15 An illustration of the nodes of the sub-network.

In order to simplify the illustration of sub-network extraction, two-time-slices DBN is presented in the two-dimensional space. Supporting nodes may be located at either the time slice $t-1$ or $t$, but the sink node, source and bridge nodes, are only located at time slice $t$.

With the sub-network, $\mathcal{G}'$, all the factors and interactions concerning control decisions have been put together: The edges between source nodes and bridge nodes explain how the controllable factors affect certain process parameters. The edges between the sink node and bridges nodes indicate how the process parameters affect the metrology measurements.

Determining the appropriate controllable variable is equivalent to choosing the appropriate values of bridge nodes because they will eventually affect metrology. For each bridge node in $\mathbb{V}_{br}^{(t)}$, there is a linear regression model in relation to its parents. The parents nodes can be further classified into three types, nodes belong to the source nodes, $\mathbb{V}_{s}^{(t)}$, nodes belongs to the bridge nodes, $\mathbb{V}_{br}^{(t)}$, and parent nodes belong to the supporting nodes, $\mathbb{V}_{sp}$.

$$V_{br} = f_{V_{br}}\left(V | V \in \mathbb{V}_{pa(V_{br})}\right) = f_{V_{br}}\left(V_1, V_2, V_3 | V_1, V_2, V_3 \in \mathbb{V}_{pa(V_{br})}, V_1 \in \mathbb{V}_{s}^{(t)}, V_2 \in \mathbb{V}_{br}^{(t)}, V_3 \in \mathbb{V}_{sp}\right). \quad (4.13)$$

For each bridge node, the effects caused by other supporting nodes cannot be manipulated. Therefore, for online control value computation, the supporting nodes will be considered as the unalterable factors and their values will be given as prior information. The next step will explain how to obtain this information about supporting nodes.

### 4.7.2 Predictive Procedure

After extracting the sub-network, a three-dimensional illustration will be employed to distinguish the variables at different time slices (see Fig. 4.16). The main advantage of using DBN for R2R control is to maximize the utilization of information for decision making. Based on this concept, the next step is to leverage the available data to predict the equipment condition, so that the control action can be made which fits better the current state.

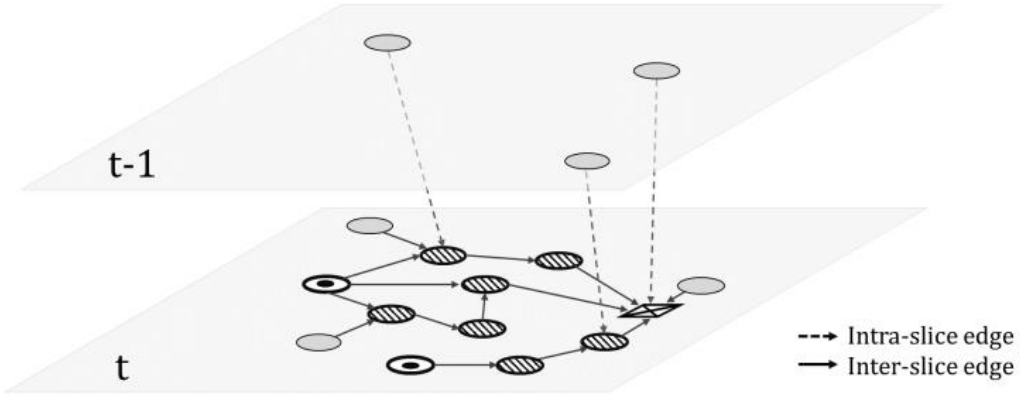


Figure 4.16 A sub-network can be illustrated in a three-dimensional space of the two-time-slice DBN.

Considering a real-time production line and assuming run $w-1$ is completed, a R2R controller should decide the controllable variable $u_{w,k}^{(t)}$ for the incoming run $w$. In order to compute $u_{w,k}^{(t)}$, input data for the SRC should be prepared. Given the wafer-based control in semiconductor manufacturing, i.e., each run indicates one wafer process, the required input data will be a collection of the latest information for the incoming wafer $w$, denoted as a 1 by $q$ vector $d_w$,

$$d_w = \left[d_w^{(t-1)} \quad d_w^{(t)}\right], \quad (4.14)$$

where $d_w^{(t-1)} = \left[u_{w,1}^{(t-1)} \quad \cdots \quad u_{w,k}^{(t-1)} \quad x_{w,1}^{(t-1)} \quad \cdots \quad x_{w,r}^{(t-1)} \quad y_w^{(t-1)}\right]$ is an non-empty vector that represents the available data of the processed wafer. $d_w^{(t)} = \left[u_{w,1}^{(t)} \quad \cdots \quad u_{w,k}^{(t)} \quad x_{w,1}^{(t)} \quad \cdots \quad x_{w,r}^{(t)} \quad y_w^{(t)}\right]$ is an empty vector that indicates unknown information of the incoming wafer. Note that the definition of all variables in $d_w$ are the same as the historical data matrix $\mathcal{D}$ in equation 4.1.

As discussed in the above steps, the values of supporting variables $\mathbb{V}_{sp}$ (10) should be given prior to the computation. The variables in $\mathbb{V}_{sp}$ are either belong to $\mathbb{V}^{(t-1)}$ or $\mathbb{V}^{(t)}$. However, all the variables in $\mathbb{V}^{(t)}$ are unknown before the wafer run $w$, i.e., $d_w^{(t)}$ in $d_w$ (4.14) is an empty vector. Thus, an iterative procedure will be conducted to give prediction for variables in $d_w^{(t)}$. This procedure is the same as equipment condition prediction which has discussed in Section 4.6.2, only the variables in $\mathbb{V}_s^{(t)} \cup \mathbb{V}_e^{(t)} \cup \mathbb{V}_{br}^{(t)}$ will be skipped in the prediction procedure.

The predication procedure is basically using the information of parent nodes. If an unknown node is a root node, i.e., node without parents, we are not able to predict its value. Thus, actions to handle different scenarios should be carefully discussed. The following flowchart shows the possible scenarios and corresponding actions.

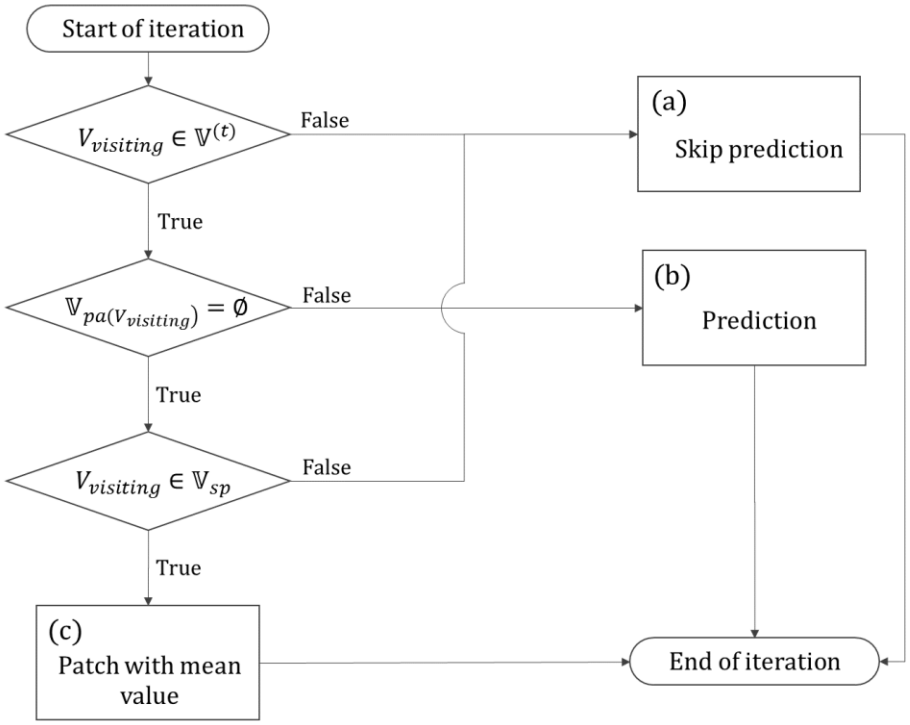


Figure 4.17 Procedure for handling different scenarios of each iteration.

For each iteration, the first logical expression will evaluate if the visiting node $V_{visiting}$ belongs to $\mathbb{V}^{(t)}$. If it is false, which means that its value is known, the prediction step can be skipped (see Fig. 4.17a); otherwise, the second logical expression will examine if $V_{visiting}$ is a root node. If $V_{visiting}$ is not a root node, a prediction procedure can be triggered and output a predicted value (see Fig. 4.17b).

If $V_{visiting}$ is a root node, the third logical expression will further check if $V_{visiting}$ belongs to $\mathbb{V}_{sp}$ and results in three possibilities:

- $V_{visiting} \in \mathbb{V}_{sp}$,
- $V_{visiting} \in \mathbb{V}_s^{(t)}$,
- $V_{visiting} \in (\mathbb{V}_{sp} \cup \mathbb{V}_s^{(t)})^c$.

The goal of the prediction procedure is to gather the information of supporting nodes. If $V_{visiting} \in \mathbb{V}_{sp}$, its missing value will be patched with the mean value (see Fig. 4.17c). If $V_{visiting} \in \mathbb{V}_s^{(t)}$, its missing value can be ignored and will be decided during the control value computation. If $V_{visiting} \in (\mathbb{V}_{sp} \cup \mathbb{V}_s^{(t)})^c$, this node is not in the scope of sub-network $\mathcal{G}'$, its missing value can be ignored as well and will not affect the final control value computation. To simplify the above statements, the third logical expression only asks if $V_{visiting} \in \mathbb{V}_{sp}$. If it is true, the missing value will be patched with the mean value; otherwise, the prediction step is skipped (Fig. 4.17a).

Imputing the missing value with the mean value can be seen as a compromising approach, considering the available information is scarce. Fortunately, iterations ended with this action are of the minority throughout the whole procedure. In the proposed framework, since all edges from time slice $t$ to time slice $t-1$ are blocked while the edges from to time slice $t-1$ to time slice $t$ are allowed (see Section 4.3.2), nodes in $\mathbb{V}^{(t-1)}$ are more likely to be root nodes than being the nodes in $\mathbb{V}^{(t)}$. In this context, the values of most of the root nodes are known. For the case study presented in Section 5.1, only two types of actions are taken: skipping prediction and giving prediction. There is no iteration ended with the last action.

### 4.7.3 Control Setting Computation

After the above prediction procedure, the updated vector $\mathcal{d}_w$ is able to provide the complete information of supporting nodes $\mathbb{V}_{sp}$ that includes the predicted value in $\mathbb{V}^{(t)}$ and the known values in $\mathbb{V}^{(t-1)}$. Replacing the support variables with their values, the equation (4.13) can be rewritten as

$$V_{br} = f'_{V_{br}}\left(V_1, V_2 | V_1, V_2 \in \mathbb{V}_{pa(V_{br})}, V_1 \in \mathbb{V}_s^{(t)}, V_2 \in \mathbb{V}_{br}^{(t)}\right) + C_{w,V_{br}},$$

where $C_{w,V_{br}}$ is the contribution of supporting nodes that can be considered as a constant for a certain run $w$. Similarly, for the sink node, i.e., the metrology measurement, the effect of supporting nodes can be isolated. Therefore, the set of functions (4.12) will be updated as

$$\mathbb{f}'_w = \{ f'_V + C_{w,V} | \, V \in (\mathbb{V}_{br}^{(t)} \cup \mathbb{V}_e)\}. \tag{4.15}$$

With the updated functions set $\mathbb{f}'_w$, we are able to compute the control value $u_{w,k}^{(t)}$. The computation can be considered as the solution of a quadratic optimization problem as shown in 4.16. A quadratic optimization problem is defined as optimizing (minimizing or maximizing) a quadratic function of several variables subject to linear constraints on these variables.

$$\min \quad \alpha\left(y_w^{(t)} - y_{taget}\right)^2 + \beta \sum_k \left(u_{w,k}^{(t)} - u_{w,k}^{(t-1)}\right)^2$$

$$\text{s.t.} \quad \mathbb{f}'_w = \{ f'_V + C_{w,V} \mid V \in (\mathbb{V}_{br}^{(t)} \cup \mathbb{V}_e)\}.$$

$$u_{k,min} \le u_{w,k} \le u_{k,max}, \tag{4.16}$$

where $y_{taget}$ is the target of metrology measurement, $u_{k,min}$ and $u_{k,max}$ specify the range of the controllable variable $u_{w,k}$. The objective function is to minimize the error of metrology while avoiding a drastic change of the controlling value between adjacent runs. The weights $\alpha$ and $\beta$ can be determined by the importance of two objective terms. For example, assume that the relative importance of the error of metrology is 90% and the relative importance of the difference of controlling value between adjacent runs is 10 %. One way to determine the weights of two terms is to divide the importance setting by the range of the variable, $= \frac{0.9}{(max(Y)-min(Y))}$ , and $\beta = \frac{0.1}{(max(U)-min(U))}$. Note that we do not standardize the data in order to obtain the control setting directly. The constraints are of a set of functions in (4.15) wherein each function indicates the relationships among the variables of sub-network $\mathcal{G}'$ except supporting nodes. Since the effects of supporting nodes cannot be manipulated, the contribution is fixed and has been computed by using the available information. A variety of methods are developed to solve the quadratic optimization problem, such as Interior-point methods, augmented Lagrangian method (Portra & Wright, 2002; Bertsekas, 1982). In this research, we use an R package *quadprog* which employs the primal-dual method (Goldfarb & Idnani, 1982). By solving this optimization problem, the suggested control value $u_{w,k}^{(t)}$ can be obtained.

## 4.8 Online Stage - Advanced Structured R2R Controller

In the previous sections, we introduced the details of each function module in the physical-informatics integrated control system in Fig. 4.1, including likelihood monitoring, a VM model, and a SRC. In this section, an Advanced Structured R2R Controller (A-SRC) will be introduced, which combines the likelihood monitoring mechanism and the SRC.

Before introducing the proposed A-SRC, the relationships among those function modules will be discussed. To understand how these modules are connected to each other, it is important to clarify the outputs of each modules, and the moment of executing these modules. A flow chart based on process time is presented in Fig. 4.18. Suppose there are two time points, time $t_0$ and time $t_1$. Two sequential wafers $w_0$ and $w_1$ are processed at time $t_0$ and $t_1$, respectively. The FDC data of wafer $w_0$, denoted as $\mathcal{d}_{w_0}$, can be obtained after process wafer $w_0$. Then the monitoring index GLI can be calculated based on $x_{w_0}$. If the GLI is acceptable, then the subsequent prognosis and control tasks proceed in the usual way. The VM model will output the predictive metrology of $w_0$, which is $\hat{y}_{w_0}$. The equipment condition prediction procedure will generate the predictive equipment values of the next wafer, denoted as $\hat{x}_{w_1}$. The essential input of SRC is $\hat{x}_{w_1}$, and $\hat{y}_{w_0}$ may or may not be the input data, which depends on the availability of actual metrology and the structure of DBN. Finally, the SRC will output the suggested control value for $w_1$, which is $u_{w_1,SRC}$.

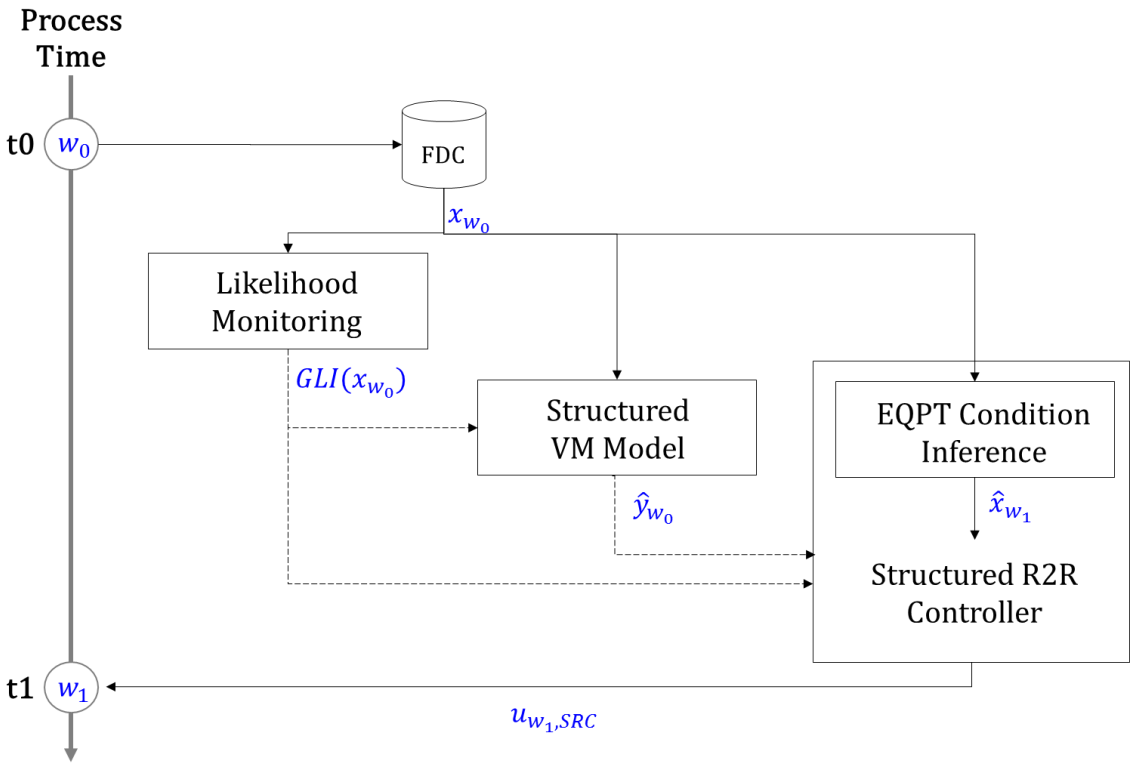


Figure 4.18 The process flow of the function modules in the framework.

The proposed framework integrates multiple function modules together while keeping the flexibility that these modules can work independently, except for the preliminary element of SRC — predictive equipment condition.

As described in Section 4.5, a monitoring loop based on the likelihood estimation can be used to trigger a model updating mechanism. The A-SRC which combines SRC and likelihood monitoring is presented in Fig. 4.19. After processing wafer $w$, if the online update is enabled, then the wafer data $d_w$ will be sent to likelihood monitor loop. The monitor loop will give the updated graph $\mathcal{G}_{cur}$ if necessary. Then, the SRC based on $\mathcal{G}_{cur}$ will compute the control value for the next wafer.

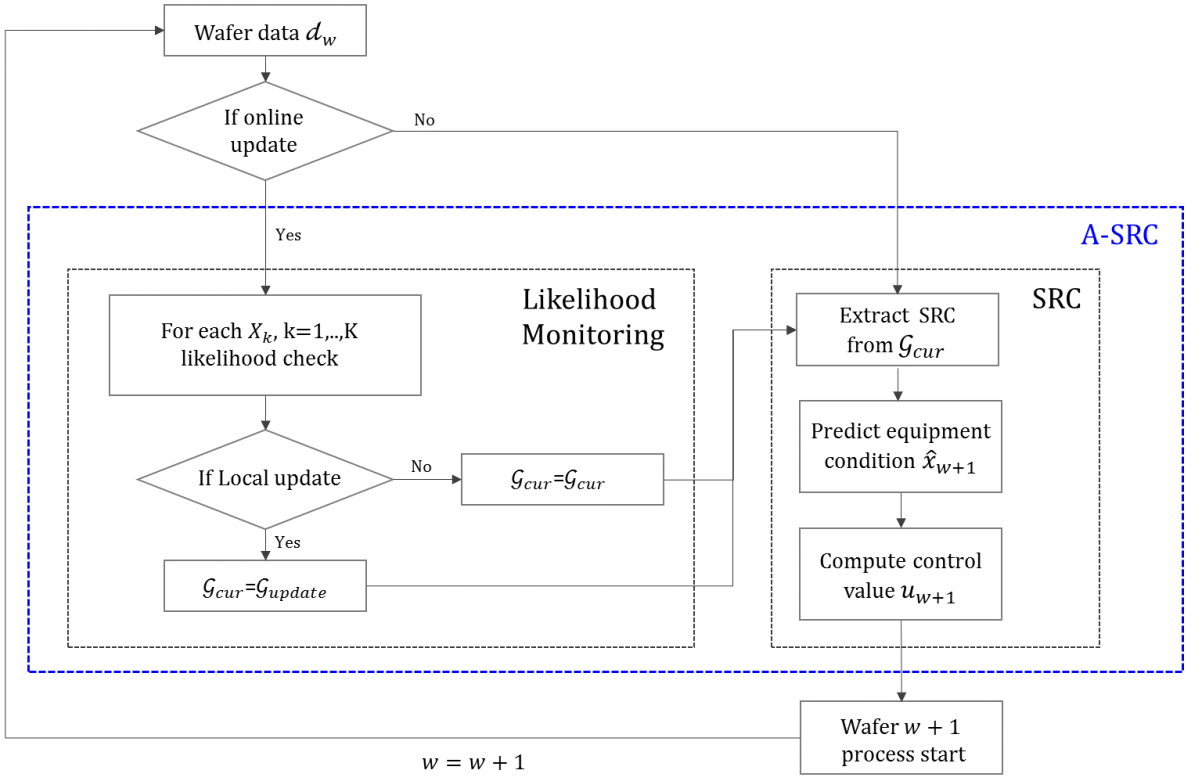


Figure 4.19 The procedure of the A-SRC.

In Section 4.5.2, the risk of updating a model has been discussed. The A-SRC should be operated under the strict condition that the drift or shift of process is still in the tolerant range; in this case, the control actions can be made by slightly updating the model. However, if the magnitude of process disturbance is high, updating the model may mask the underlying problem. These limitations should be clarified before employing A-SRC in practice. For instance, the updating magnitude should be smaller than a specified threshold. If the magnitude exceeds the limit, more urgent actions should be taken, such as preventive maintenance or corrective maintenance. Therefore, more preliminary work needs to be done before A-SRC can be used to automatically update the model. Choosing between a basic SRC and an A-SRC depends on several factors, such as the critical level of the process or the cost of model maintenance.

The main advantage of the proposed integrated framework is that all function modules are based on one data-driven model so that the work of constructing and maintaining the model can be greatly simplified. Besides, as this framework is designed to maximize the utilization of the available data, the decision is made in an information-rich manner.

# 5 Framework Assessment

In the previous chapter, the details of the proposed framework have been explained, including the approach of each function module. To validate the proposed framework, several datasets will be used to evaluate the efficiency of these function modules. A case study conducted on the real dataset provided by our industrial partner will be firstly presented, which assesses the performance of VM and SISO R2R control. The approach and the result will be shown in Section 5.1. As mentioned in Section 3.4, the R2R control is the main focus of this thesis, more assessment should be executed to see if the controller is able to handle various scenarios. In section 5.2, several simulated datasets which incorporate different disturbances will be created to evaluate the capability of the proposed R2R controllers in the MIMO system.

## 5.1 Case Study 1 – Chemical-Mechanical Polishing Process

In this section, the proposed framework will be applied to a real dataset, which includes the FDC data from the Chemical-Mechanical Polishing (CMP) process, the metrology measurements after the process, and the polishing time of each wafer, considered as the regulating data. More descriptions about the dataset will be introduced in the first subsection. The characteristics of the CMP process and necessary assumptions will be explained as well. Following the diagram shown in Fig. 4.1, a DBN will be learned in the offline stage. Then, two sub-networks for online VM and R2R control can be extracted from this DBN. The assessment will be done by comparing their performance with other existing approaches.

### 5.1.1 Case Description

As shown in Fig. 5.1, a typical CMP tools includes a rotating platen with a pad, a carrier that holds wafer upside-down, a head to press the wafer against the pad, and a pad conditioner. Through the proper settings of the chemical slurry and the mechanical force, unwanted materials and surface topologies on the wafer surface can be removed.

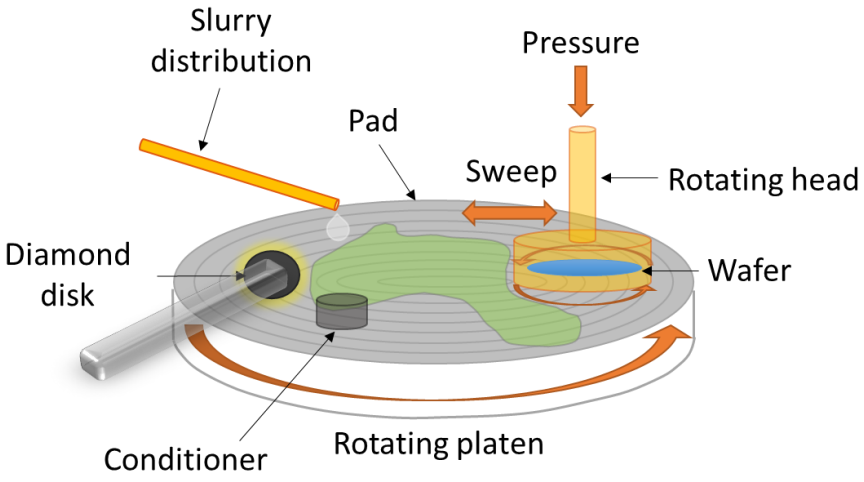


Figure 5.1 An illustration of a typical CMP tool.

The FDC parameters in the dataset cover the platen usage, the head speed, the conditioning pressure, etc. A deterministic sampling policy was put in practice for the CMP process. Two wafers per lot are sampled to take the metrology measurements, which are noted as the post-polishing thicknesses. Furthermore, it is known that the performance of the precedent processes before CMP will inevitably influence the output of CMP, including etching and Chemical Vapor Deposition (CVD). Two pre-CMP measurements, the post-etching depth and post-CVD thickness are also considered in the variable set.

In this case, there are four operations involved: the metrology measured after etching, the metrology measured after deposition, CMP process, and the metrology measured after CMP. In the CMP operation, each wafer has to pass three sequential platens. The corresponding hierarchical structure is presented in Fig. 5.2.

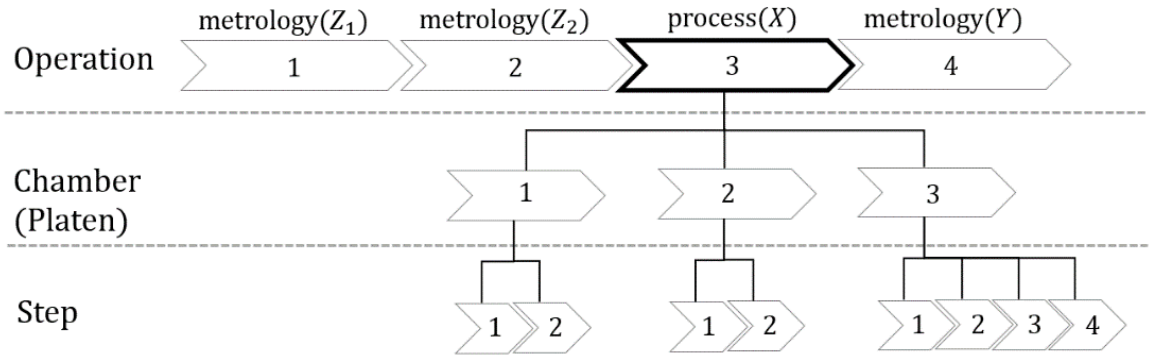


Figure 5.2 The hierarchical structure of the CMP process.

Although the studied CMP process does not apply any R2R control, the polishing time of the second platen will be considered as the controllable variable for the future R2R control system. Note that the first platen is an end-point mode which is not controllable, but the polishing time of the first and third platen will be included as the process features in the structure learning.

Generally, a R2R controller for CMP aims to compensate the quality variation caused by the unsteady removal rate. The remove rate cannot be observed directly but computed based on the posterior metrology data. The removal rate of the wafer at run $w$ is defined as

$$r_w = \frac{\text{removal thickness}}{\text{polishing time}}. \tag{5.1}$$

The conventional R2R controller works to estimate the removal rate $\hat{r}_w$ based on the result of the previous run. Given the post-CMP thickness and post-CVD thickness, the suggested polishing time for the incoming run is computed. In this case study, the polishing time $u_{w,SRC}$ will be computed based on the proposed SRC.

## 5.1.2 Offline Stage

As shown in Fig. 4.1, three steps are required in the offline stage, historical data pre-processing, knowledge-based configuration, and structure learning. The detailed approaches for this case study are presented as the following parts.

*A. Historical Data Pre-processing*

Following the procedure described in Section 3.2, temporal FDC data are transformed into 122 features, denoted as $\{X_1 \quad \cdots \quad X_{122}\}$, at the wafer level. All the other data, including the metrology of our concern, the thicknesses oxidation ($Y$), the post-etching depth ($Z_1$), the post-CVD thickness ($Z_2$), the polishing time of the first platen ($Z_3$), the third platen ($Z_4$), and the polishing time of second platen ($U$) as the controllable variable, are consolidated in the same granularity, i.e., at the wafer level. A data matrix $\mathcal{D} = [\mathcal{D}^{(t-1)} \quad \mathcal{D}^{(t)}]$, where $\mathcal{D}^{(t)} = \left[U^{(t)} \quad X_1^{(t)} \quad \cdots \quad X_{122}^{(t)} \quad Z_1^{(t)} \quad Z_2^{(t)} \quad Z_3^{(t)} \quad Z_4^{(t)} \quad Y^{(t)}\right]$ and $\mathcal{D}^{(t-1)} = \left[U^{(t-1)} \quad X_1^{(t-1)} \quad \cdots \quad X_{122}^{(t-1)} \quad Z_1^{(t-1)} \quad Z_2^{(t-1)} \quad Z_3^{(t-1)} \quad Z_4^{(t-1)} \quad Y^{(t-1)}\right]$, is defined to represent the 545 wafers with 256 variables. The wafers are ordered by the real processing sequence.

As described in 5.1, the removal rate is computed by the removal thickness and polishing time, and we assume that only the polishing time of second platen can be control, thus, the removal rate of wafer $w$ in this case can be computed by the following formula:

$$r_w = \frac{\text{removal thickness}}{\text{polishing time}} = \frac{z_{w,2}^{(t)} - y_w^{(t)}}{\left(u_w^{(t)} + z_{w.3}^{(t)} + z_{w,4}^{(t)}\right)}. \tag{5.2}$$

In this case study, the real removal rate $R = [r_1 \quad \cdots \quad r_{545}]$ are first computed based on 5.2. The vector $R$ will not be used during the model construction but revealed for validating the efficiency of the suggested polishing time.

To assess the efficiency of the proposed VM model and the SRC, data matrix $\mathcal{D}$ will be divided into two datasets: the first 70% of wafers is assigned to be the training set $\mathcal{D}_1$ for offline stage; the remaining 30% is kept in the testing set $\mathcal{D}_2$ for simulating the online prognosis and control. For VM model evaluation, both $\mathcal{D}_1$ and $\mathcal{D}_2$ will used for comparing with other models. For R2R control, the simulating control value will be computed for $\mathcal{D}_1$ and $\mathcal{D}_2$, but only the performance of $\mathcal{D}_2$ will be considered for comparing with other controllers.

*B. Knowledge-based Configuration*

In order to better present the physical phenomenon through the DBN, more existing physical information should be included by knowledge-based configuration. As described in Section 4.3.1, the intra-slices edges regarding time dependency can be generated based on the hierarchical structure of CMP process shown in Fig. 5.2, including operation level, chamber level. In addition, a pre-defined association matrix $\mathcal{M}$ was provided by the SME (see Table 5.1), so that several impossible edges can be defined as well.

Next, following the description of Section 4.3.2, several blocking inter-slice edges can be generated regarding the process nature. Finally, by consolidating those configurations, 48,083 blocking rules are obtained and added into the blacklist $\mathcal{L}$.

Table 5.1 An association matrix provided by the SME.

| From/To | | Pressure | | | | Speed | | | Slurry | | | Status | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ | $p_6$ | $p_7$ | $p_8$ | $p_9$ | $p_{10}$ | $p_{11}$ | $p_{12}$ |
| Pressure | $p_1$ | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | $p_2$ | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | $p_3$ | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| | $p_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Speed | $p_5$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| | $p_6$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| | $p_7$ | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| Slurry | $p_8$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| | $p_9$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| | $p_{10}$ | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| Status | $p_{11}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| | $p_{12}$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

*C. Structure Learning*

As described in Section 4.4, structure learning includes learning phase and repairing phase. First, with the training set, $\mathcal{D}_1$, and the blacklist, $\mathcal{L}$, an initial structure of BN regardless of the time factor can be learned through Hill-Climbing approach. As Hill-Climbing is a greedy search algorithm, the global optimum will not be easily reached. To explore the search space more completely, a restart procedure is implemented. The optimal restart frequency and the number of perturbed edges are determined empirically. A grid search approach procedure is employed. The following Table 5.2 shows that the Relative Score $RS_{h_1,h_2}$ of each $\mathcal{G}_{h_1,h_2}$ before repairing phase, and as expected, the score of $\mathcal{G}_{100,100}$ is the highest one. The $RS$ after repairing phase are presented in Table 5.3. Given 100 times of restart with 50 perturbed edges, the score is the highest one in this experiment, i.e., $\mathcal{G}_{best} = \mathcal{G}_{100,50}$. Among all the settings, $\mathcal{G}_{60,20}$ seems to provide comparable performance as well, as $RS_{60,20}$ is only 0.65%. When the computational time is taken into account, $\mathcal{G}_{best}$ takes 149.28 minutes for learning while $\mathcal{G}_{60,20}$ takes 30.91 minutes. Thus, $(h_1, h_2) = (60, 20)$ is determined to be the best setting in this study. And the final DBN $\mathcal{G}(\mathcal{D}_1)$ can be determined (see Fig. 5.3).

With the DBN $\mathcal{G}(\mathcal{D}_1)$ as a general-purpose model, some function modules can be derived based on this structure in the next online stage. In the following subsection, a VM model and an SRC will be presented.

Table 5.2 The Relative Score of each combination before repairing phase.

| $h_1$ \ $h_2$ | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 2.16% | 1.72% | 1.73% | 1.92% | 1.53% | 1.60% | 1.11% | 1.54% | 1.46% | 1.50% |
| 20 | 2.18% | 1.50% | 1.59% | 1.27% | 1.19% | 1.41% | 1.40% | 1.44% | 1.45% | 1.42% |
| 30 | 1.73% | 1.50% | 1.12% | 1.58% | 1.40% | 0.97% | 0.92% | 0.75% | 0.84% | 1.02% |
| 40 | 2.11% | 1.58% | 1.48% | 0.89% | 0.99% | 1.37% | 0.91% | 1.29% | 1.23% | 0.40% |
| 50 | 1.57% | 1.56% | 1.37% | 1.37% | 0.94% | 0.83% | 0.60% | 0.80% | 0.26% | 0.71% |
| 60 | 1.70% | 1.06% | 1.06% | 1.05% | 1.28% | 0.67% | 0.45% | 1.13% | 0.46% | 0.72% |
| 70 | 2.13% | 1.15% | 1.33% | 1.33% | 0.51% | 0.42% | 0.29% | 0.40% | 0.14% | 0.44% |
| 80 | 1.70% | 0.81% | 1.01% | 0.86% | 0.80% | 0.81% | 0.31% | 0.65% | 0.16% | 0.15% |
| 90 | 1.68% | 1.34% | 0.82% | 0.48% | 0.61% | 0.73% | 0.38% | 0.35% | 0.31% | 0.25% |
| 100 | 1.47% | 1.06% | 0.92% | 0.85% | 0.57% | 0.83% | 0.19% | 0.46% | 0.20% | **0%** |

Table 5.3 The Relative Score of each combination after repairing phase.

| $h_1$ \ $h_2$ | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 1.60% | 1.56% | 0.80% | 1.52% | 1.49% | 1.87% | 0.95% | 1.40% | 1.37% | 1.96% |
| 20 | 1.61% | 1.73% | 1.54% | 0.87% | 1.15% | 1.89% | 1.90% | 1.85% | 1.43% | 1.47% |
| 30 | 1.59% | 1.48% | 0.71% | 1.30% | 1.26% | 0.45% | 0.79% | 0.22% | 0.45% | 1.30% |
| 40 | 1.47% | 1.54% | 1.44% | 1.06% | 1.68% | 1.26% | 0.37% | 2.58% | 1.23% | 0.75% |
| 50 | 2.08% | 1.51% | 1.48% | 1.32% | 0.50% | 0.91% | 0.57% | 0.58% | 0.28% | 0.31% |
| 60 | 1.58% | **0.65%** | 0.74% | 0.73% | 1.65% | 1.01% | 1.11% | 1.67% | 0.33% | 1.03% |
| 70 | 1.54% | 0.59% | 1.72% | 1.19% | 0.41% | 1.19% | 0.65% | 0.94% | 1.47% | 0.73% |
| 80 | 1.62% | 0.70% | 0.66% | 0.46% | 0.92% | 0.88% | 1.00% | 0.69% | 0.87% | 0.68% |
| 90 | 1.51% | 1.32% | 0.75% | 0.84% | 0.80% | 0.90% | 0.80% | 1.45% | 0.63% | 1.49% |
| 100 | 1.36% | 1.06% | 1.05% | 0.83% | **0%** | 1.06% | 0.57% | 0.71% | 0.88% | 0.50% |

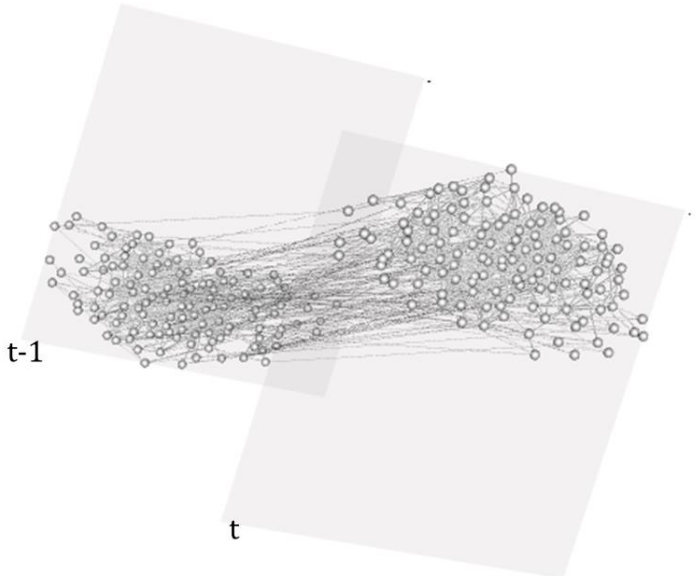


Figure 5.3 The resulted DBN $\mathcal{G}(\mathcal{D}_1)$.

### 5.1.3 Online Stage – VM Model Evaluation

By looking at the target metrology variable $Y^{(t)}$ and its parents $\mathbb{V}_{pa(Y^{(t)})}$, the sub-network which presents a VM model can be easily extracted (see Fig. 5.4). As expected, one of the pre-CMP measurements, i.e., $Z_1$, directly impacts the metrology. Other eight FDC variables were also found to significantly explain the variation of the target metrology variable.

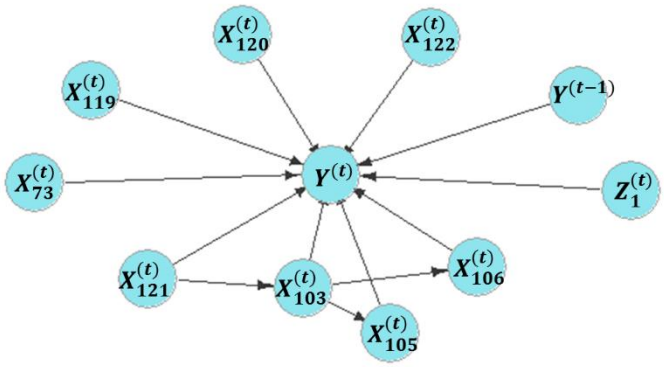


Figure 5.4 A VM model extracted from the fitted DBN $\mathcal{G}(\mathcal{D}_1)$.

The comparison of observed metrology and predicted metrology for both training set and testing set is presented in Fig. 5.5, where the good fit can be appreciated. The performance of the model is evaluated using the Root Mean Square Error (RMSE), expressed in Eq. (5.3). The RMSEs for the training set and testing set are 28.87 and 32.33, respectively.

$$RMSE = \sqrt{\frac{\sum_{w=1}^{n}(\hat{y}_w - y_w)^2}{n}}. \tag{5.3}$$

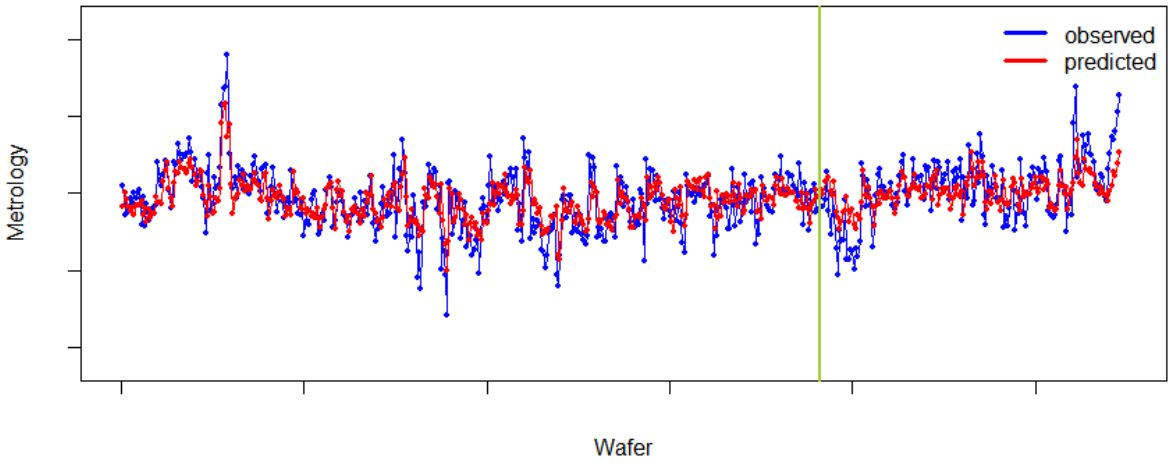


Figure 5.5 Compare the observed values and predicted values of the DBN model. Training set and testing set are separated by green line.

To validate the proposed structured VM model, four other common techniques applied in the VM modeling literature are benchmarked. *k*-fold cross-validation is used to determine the optimal hyperparameters for each method and *k* is set to 10 in this research. In each

iteration, data from the $k$-1 folds are used for training and the RMSE is evaluated in the $k^{th}$ left-out fold. The $k^{th}$ fold is also called a validation set. LASSO considers a regularization parameter $\lambda$ to restrict the variation of coefficients, the optimal $\lambda$ in this study is chosen by the one standard error rule (1 SE rule) (Breiman et al., 1984). Regression trees have two hyperparameters: maximum depth of the tree and minimum observations per split, which are determined by a grid search approach. The performance is assessed by the averaged RMSE on the validation set. The same procedure is applied to random forests, considering the number of trees and the maximum number of terminal nodes.

Table 5.4 summarizes the performance of different VM models for this case study. By evaluating RMSE, DBN, only showing causality dependencies between all variables but also providing the flexibility for incorporating domain knowledge. The lack of causal consistency has been a permanent criticism of data-driven approaches and the proposed methodology circumvents this criticism and provides an effective solution that incorporates all existing sources of knowledge about the process under analysis.

Table 5.4 The performance evalutaion of different VM models.

| RMSE | Training Set | Testing Set | #of variables selected |
|---|---|---|---|
| DBN | 28.87 | 32.33 | 10 |
| Stepwise Regression | 22.85 | 35.55 | 42 |
| LASSO Regression | 28.59 | 32.66 | 25 |
| Regression Trees | 32.50 | 32.79 | 15 |
| Random Forest | 16.13 | 36.03 | -- |

The effects of introducing SME-specified blacklist can be discussed from three perspectives: the fitness of the learned structure, the performance of the VM model and the interpretability of process interactions. The first two perspectives are evaluated by the BIC score and the RMSE, respectively. As shown in Table 5.5, the restriction of edges leads to a lower score of the overall structure, i.e., $Score(\mathcal{G}_{wi\_SME}) < Score(\mathcal{G}_{wo\_SME})$. In this case, incorporating the SME-specified blacklist does not impact RMSE.

Table 5.5 The evalutation of impact of SME-spcified blacklist.

| DBN | SME-specified blacklist | Score | RMSE of testing set |
|---|---|---|---|
| $\mathcal{G}_{wo\_SME}$ | No | 28149.5 | 32.33 |
| $\mathcal{G}_{wi\_SME}$ | Yes | 21616.7 | 32.33 |

Since the performance of VM model is not affected by edge restrictions, i.e., $\mathbb{V}_{pa(Y^{(t)})}$ did not change, the last assessment is to verify if $\mathcal{G}_{wo\_SME}$ can provide a better understanding of related causalities. An example is shown in Fig. 5.6. One of the variables in $\mathbb{V}_{pa(Y^{(t)})}$ is the conditioner pressure. The result of $\mathcal{G}_{wo\_SME}$ shows that the conditioner pressure can be affected by the platen speed and head speed, which does not fit the process physics. By considering existing domain knowledge, $\mathcal{G}_{wi\_SME}$ can provide better causal structure. The explicit cause-effect representation among variables not only allows process engineers to interpret the VM

model from a physical viewpoint, but also provides the possibility for further root cause tracking.

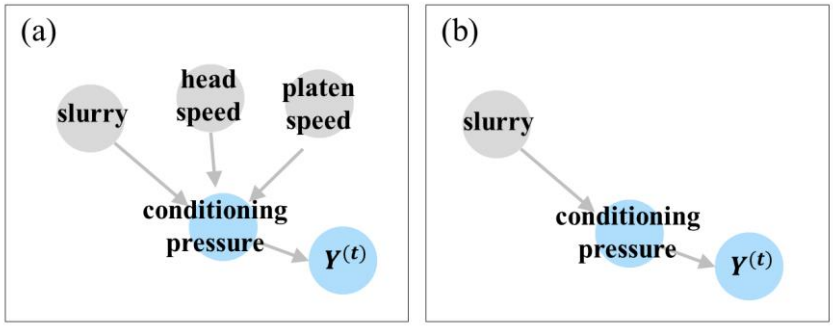


Figure 5.6 An example of casualties between variables: (a) $\mathcal{G}_{wo\_SME}$; (b) $\mathcal{G}_{wi\_SME}$.

In summary, a VM model based on the DBN was able to consolidate the domain knowledge with data, so that the final structured model could describe the existing relationships in an accurate and physically sound manner. We have verified the obtained DBN structure with a process SME, who confirmed that the revealed associations do have a relevant process meaning.

### 5.1.4 Online Stage – Controller Evaluation

Following the instruction shown in Section 4.7, the sub-network for implementing the SRC can be extracted (see Fig. 5.7). This sub-network includes one source node, one sink node, 16 bridge nodes, 29 supporting nodes, and a corresponding $\mathbb{f}'$ of 17 regression models.

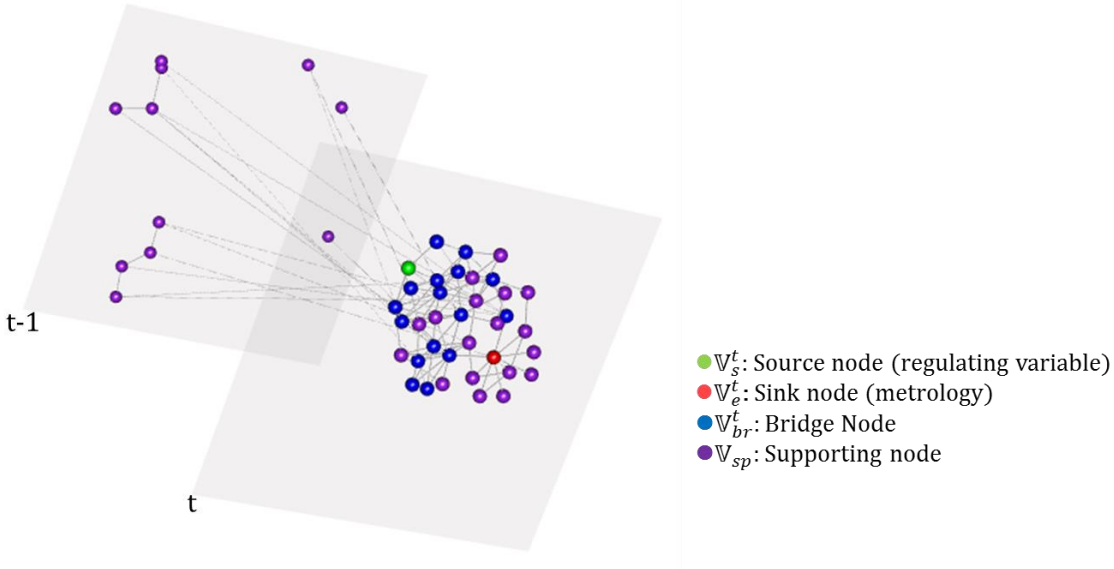


Figure 5.7 The sub-network for implementing the SRC.

The VM model, which already presented in the previous subsection also can be seen here, as shown in Fig. 5.8a, where the target metrology variable is considered as the sink node. Multiple effects can be explicit through this sub-network as well. Fig. 5.8b shows an example of one bridge node and its parent nodes. This bridge node can be not only affected by

controllable variables but also influenced by a variable at time slice $t-1$ and a variable at time slice $t$.

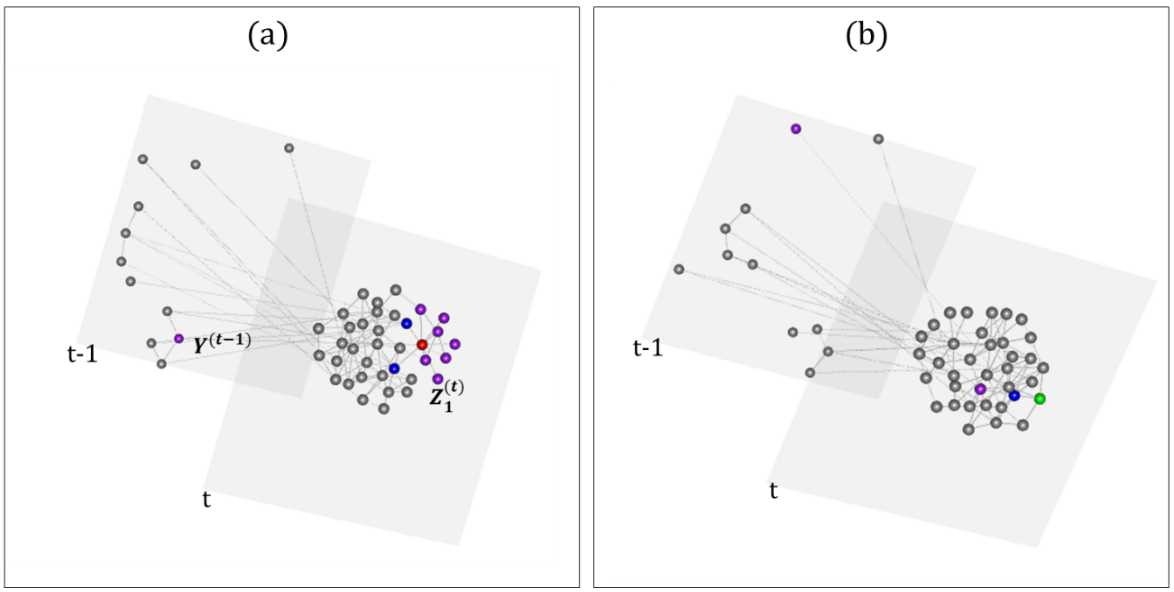


Figure 5.8 (a) The sink node and its parent nodes; (b) one of the bridge nodes and its parent nodes.

Within the set of supporting nodes, 18 nodes belong to $\mathbb{V}^{(t)}$ which are unknown before run $w$. Through the iterative prediction procedure described in Section 4.7.2, the unknown values are updated by the predicted ones. The set of functions for run *w*, expressed as $\mathbb{f}'_w$, can be obtained and used for solving the following quadratic optimization problem:

$$u_{w,SRC}^{(t)}: \min_{u_{w,SRC}^{(t)}} \alpha\left(y^{(t)} - y_{taget}\right)^2 + \beta\left(u_{w,SRC}^{(t)} - u_{SRC}^{(t-1)}\right)^2$$

$$\text{s.t.} \quad \mathbb{f}'_w = \{ f'_Y, f'_{X_{24}}, f'_{X_{25}}, f'_{X_{26}}, f'_{X_{28}}, f'_{X_{29}}, f'_{X_{38}}, f'_{X_{39}}, f'_{X_{43}}, f'_{X_{44}}, f'_{X_{45}}, f'_{X_{55}},$$
$$f'_{X_{66}}, f'_{X_{74}}, f'_{X_{75}}, f'_{X_{97}}, f'_{X_{109}}, f'_{X_{120}}\}$$

$$u_{min} \leq u_{w,SRC} \leq u_{max}.$$

The suggested control value, denoted as $u_{w,SRC}^{(t)}$, for wafer *w* will be computed.

To evaluate the efficiency of the new controller, $u_w^{(t)}$ in Equation (5.2) will be replaced by $u_{w,SRC}^{(t)}$. Given the pre-measurement $z_{w,2}^{(t)}$ and the real removal rate $r_w$, the expected metrology measurement $y_{w,SRC}^{(t)}$ can be obtained as follows:

$$y_{w,SRC}^{(t)} = z_{w,2}^{(t)} - r_w \times \left(u_{w,SRC}^{(t)} + z_{w,3}^{(t)} + z_{w,4}^{(t)}\right).$$

Two controllers are adopted as benchmarks, namely a dEWMA controller (Butter and Stefani, 1994), and a dEWMA controller incorporating a PLS model (Khan et al. 2008). To simplify the notation, the dEWMA controller incorporating a PLS model is denoted as PLS-RC in the rest of the thesis. The process flows of these controllers are presented in Appendix A. The suggested control value proposed by the dEWMA controller is denoted as $u_{w,dEWMA}^{(t)}$ and the expected

metrology will be denoted as $y_{w,dEWMA}^{(t)}$. Similarly, the suggested control setting proposed by the PLS-RC is denoted as $u_{w,PLS-RC}^{(t)}$ and the expected metrology will be referred to as $y_{w,PLS-RC}^{(t)}$. In this study, the control value of controllers will be computed at the wafer level. Considering the metrology of the previous wafer in the same lot may not be obtained in time, we average all the wafers in the same lot and get a lot-based control value. In other words, wafers in the same lot will be applied with the same control value.

Generally, only a few wafers are sent to the metrology tools for measuring; therefore, the metrology data of regular run is missing in the collected dataset (see Fig.5.9a). Since we need real measurements of each wafer to assess the controller performance, only the data of metrology run are used in this case study, as shown in Fig 5.9b. To simulate the scenario where only partial metrology data are available, we assume that sampling takes place every three lots. The metrology data of non-sampled lots are masked and will not be used for computing the control setting of each controller (see Fig 5.9c).

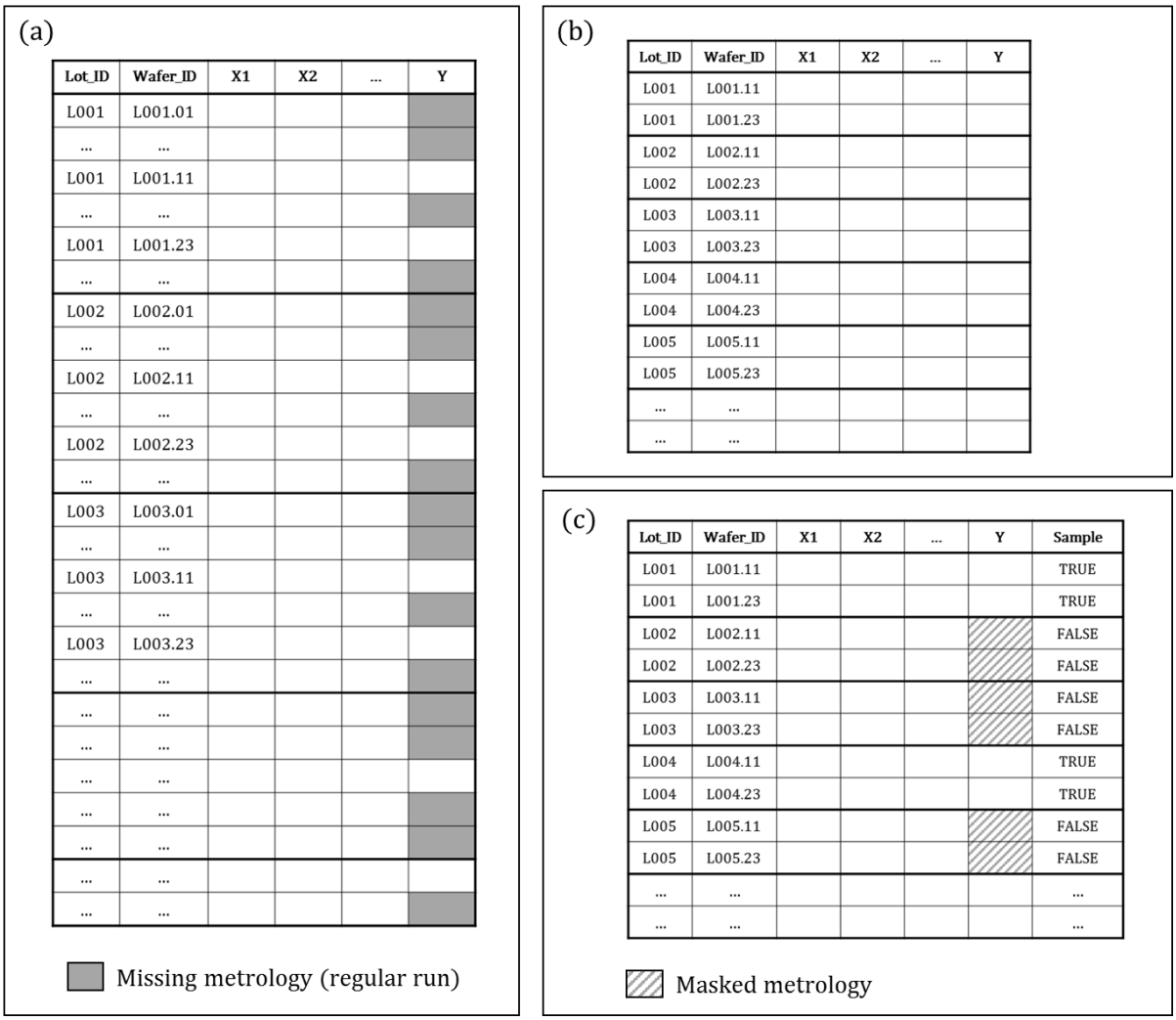


Figure 5.9 (a) The collected dataset which includes both regular run and metrology run; (b) the dataset of metrology run; (c) the dataset for evaluation..

The weights of two filters of dEWMA controller are determined by a grid search approach. By evaluating the MSE of each setting, the optimized weights were set to 0.45 and

0.5. As PLS-RC is based on dEWMA, the weights of filters are set to 0.45 and 0.5 as well. The number of components of PLS-RC is determined by cross-validation on the training set $\mathcal{D}_1$. As SRC and PLS-RC made use of the training set $\mathcal{D}_1$ to learn the model, only $\mathcal{D}_2$ will be evaluated. To compare the efficiency of different approaches, Mean Square Error (MSE) is used to be the performance index, defined as

$$MSE = \frac{1}{w}\sum_{w=1}^{n}\left(y_w^{(t)} - y_{target}\right)^2. \quad (5.4)$$

To evaluate the performance of controllers, $y_w^{(t)}$ will be replaced by either $y_{w,dEWMA}^{(t)}$, $y_{w,PLS-RC}^{(t)}$, $y_{w,SRC}^{(t)}$ or $y_{w,ASRC}^{(t)}$.

As shown in Table 5.6, the new controller SRC can reduce the MSE of testing set by 22.1%, while the dEWMA controller and PLS-RC only reduce 13.7% and 18.1, respectively. Thus, we are able to conclude that SRC which considers the overall process system is capable of efficiently reducing the variability of the process. As the local update mechanism is not triggered in this case study, the result of A-SRC is the same as the result of SRC.

Table 5.6 Comparison among different control schemes.

| Approach | MSE of the testing set $\mathcal{D}_2$ | Improvement (%) |
|---|---|---|
| Without control | $\text{MSE} = \frac{1}{n_2}\sum_{w=1}^{n_2}(y_w^{(t)} - y_{target})^2 = 693.89$ | -- |
| dEWMA controller | $\text{MSE} = \frac{1}{n_2}\sum_{w=1}^{n_2}(y_{w,dEWMA}^{(t)} - y_{target})^2 = 598.68$ | 13.7% |
| PLS-RC | $\text{MSE} = \frac{1}{n_2}\sum_{w=1}^{n_2}\left(y_{w,PLS-RC}^{(t)} - y_{target}\right)^2 = 568.18$ | 18.1% |
| SRC | $\text{MSE} = \frac{1}{n_2}\sum_{w=1}^{n_2}(y_{w,SRC}^{(t)} - y_{target})^2 = 533.75$ | 22.1% |
| A-SRC | $\text{MSE} = \frac{1}{n_2}\sum_{w=1}^{n_2}(y_{w,ASRC}^{(t)} - y_{target})^2 = 533.75$ | 22.1% |

## 5.2 Case Study 2 – Simulated Process with Disturbances

An ideal R2R controller should work well under different types of disturbances, such as the gradual drift or unexpected shift. As obtaining the real data with these behaviors can be difficult, simulated data will be used for a more delicate assessment. Besides, a R2R controller should be able to deal with both SISO and MIMO system. The real case study has shown that the

efficiency of SRC in SISO system. In this section, the simulated case of a MIMO system will be examined.

The study on the simulated data will be presented following the procedures in Fig. 5.10. To generate different types of variables in the control system, the relationship between those variables should be predefined. This preliminary work for data simulation is presented in Section 5.2.1. A universal structure for three synthetic datasets is specified as well as different disturbances are defined. To simulate the process under different controllers, specific setting of each controller is specified, explained in Section 5.2.3, under the general settings. Finally, the performance of three controllers under various disturbances is assessed in Section 5.2.4.

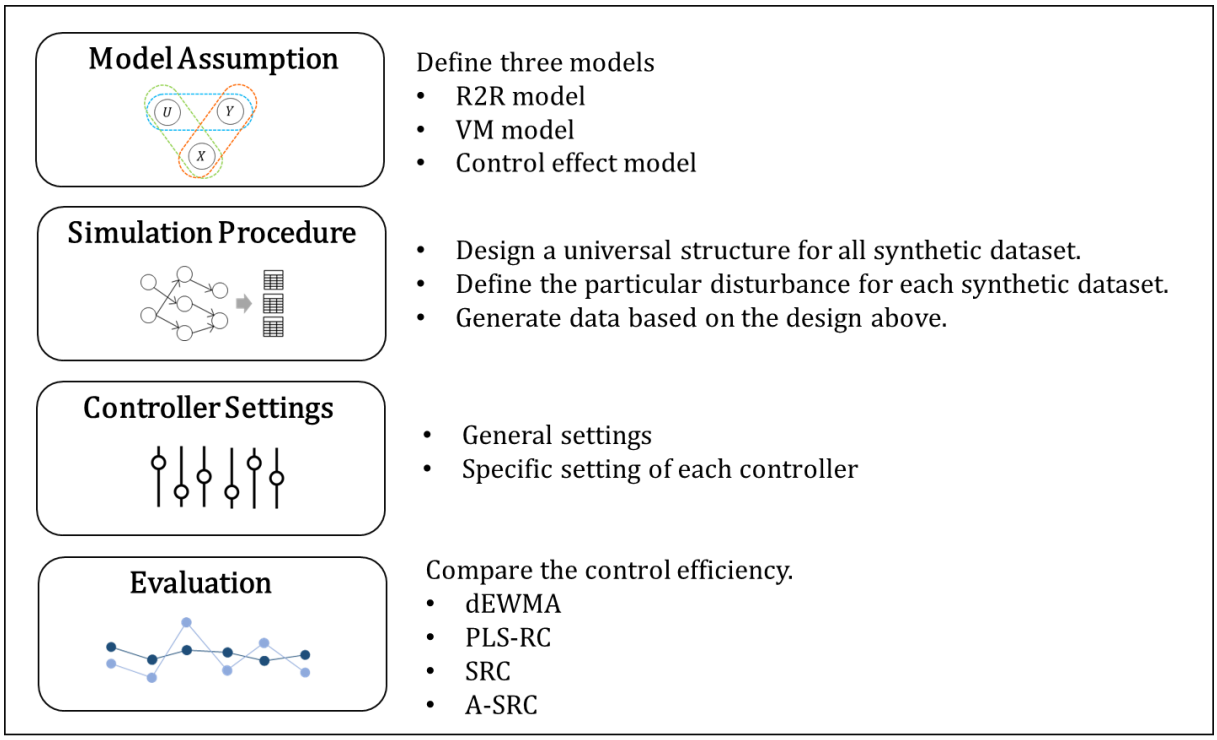


Figure 5.10 The analytical procedure of the simulated case.

## 5.2.1 Model Assumptions

In this section, we consider a simple MIMO system with $k$ process inputs and $\delta$ process outputs. This system can be described by several sets of variables (see Fig. 5.11a). Assume $U$ is a set of controllable variables, i.e., process inputs, and $Y$ is a set of metrology variables, i.e., process outputs. Let $X$ be the set of FDC variables which consists of two types of variables: $X_{br}$ and $X_{sp}$, and $X = [X_{br} \quad X_{sp}]$. The bridge variables $X_{br}$ link controllable variables and metrology variables, and the support variables $X_{sp}$ affect the metrology variables but cannot be manipulated by controllable variables.

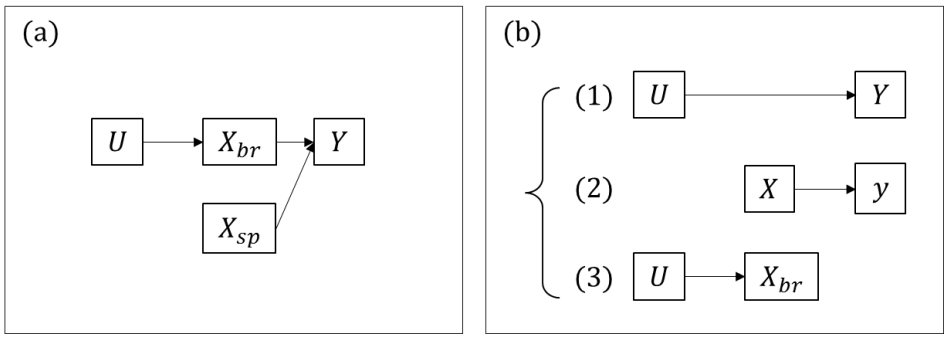


Figure 5.11 (a) A process control system; (b) decompose the system in terms of the three sets of variables: (1) R2R model; (2) VM model; (3) control effect model.

As shown in Fig. 5.11(b), the structure can be further decomposed into three individual models:

(1) R2R model describes the relationship between controllable variables and metrology variables. It is conventionally determined by performing the DOE in the R&D phase.
(2) VM model characterizes the relationship between process variables and metrology variables.
(3) Control effect model explains how the impact from controllable variables on the process variables.

To generate the synthetic datasets, these models should be clearly defined. Assume that the R2R model is a linear system described as

$$y_w = u_w A + d_{w,o} + \varepsilon_{w,uy}, \tag{5.5}$$

where $y_w \in \mathbb{R}^{1\times\delta}$ denotes the metrology values, i.e., the outputs, obtained at the $w^{\text{th}}$ run, $u_w \in \mathbb{R}^{1\times k}$ is the corresponding controllable values of run $w$, i.e., the inputs, $A \in \mathbb{R}^{k\times\delta}$ is the system gain matrix that relates the inputs to outputs, $d_{w,o}$ is an offset, and $\varepsilon_{w,uy}$ is a multivariate white noise.

The process outputs can be described by a set of FDC variables, which is also known as the VM model:

$$y_w = x_w B + \varepsilon_{w,xy}, \tag{5.6}$$

where $x_w \in \mathbb{R}^{1\times r}$ is the FDC variable vector at run $w$, $B \in \mathbb{R}^{r\times\delta}$ is the estimated regression coefficient matrix obtained from historical data, $\varepsilon_{w,xy}$ is a multivariate prediction error of the VM model. To address the effect that is contributed by controllable variables, the above VM model can be reformulated as

$$y_w = x_{w,br} B_1 + x_{w,sp} B_2 + \varepsilon_{w,xy}, \tag{5.7}$$

where $x_{w,br} \in \mathbb{R}^{1\times r_1}$ is the bridge variable vector, $x_{w,sp} \in \mathbb{R}^{1\times r_2}$ is the support variable vector, and $B_1 \in \mathbb{R}^{r_1\times\delta}$ and $B_2 \in \mathbb{R}^{r_2\times\delta}$ are regression coefficient matrix of $x_{w,br}$ and $x_{w,sp}$, respectively.

The control effect model describes how FDC variables can be affected by the values of controllable variables and is expressed as

$$x_{w,br} = u_w C + \varepsilon_{w,ux}, \tag{5.8}$$

where $C \in \mathbb{R}^{k \times r_1}$ is the mapping matrix which projects controllable variables to bridge variables, and $\varepsilon_{w,ux}$ is the noise vector.

From the model descriptions, the relationships among those variables are established. These models enable us to generate variables sequentially based on the procedures illustrated in Fig. 5.12. Some configuration should be specified in advance, including the distribution setting of some variables and the values of some matrices. In the next section, three datasets will be generated following this procedure.

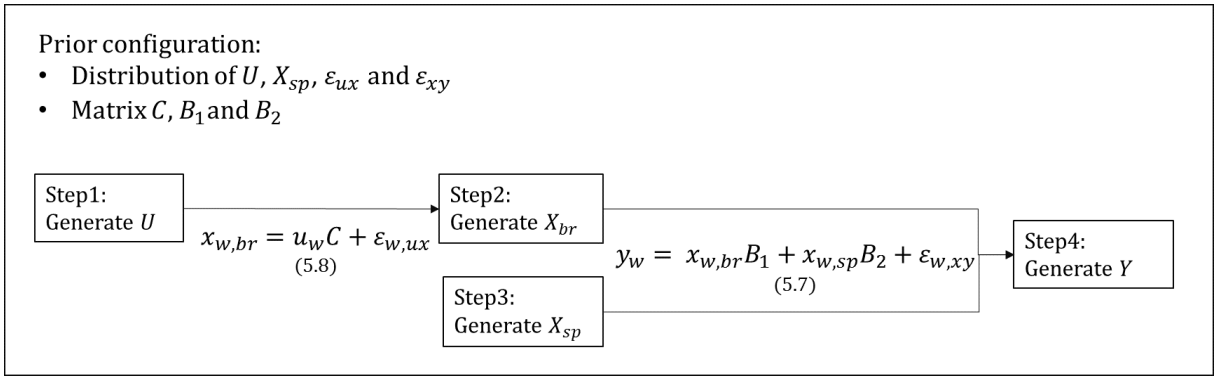


Figure 5.12 The procedure of variable generation.

As shown in Fig. 5.12, two models – the control effect model and the VM model are employed to simulate the data. Since three types of variables are connected to each other, the third model – R2R model can be derived based on the setting of the other two models. The bridge variables in 5.7 are replaced with the control effect model in 5.8 and derive the following expression.

$$\begin{aligned} y_w &= \left(u_w C + \varepsilon_{w,ux}\right) B_1 + x_{w,sp} B_2 + \varepsilon_{w,xy} \\ &= u_w C B_1 + x_{w,sp} B_2 + \left(\varepsilon_{w,ux} B_1 + \varepsilon_{w,xy}\right) \\ &= u_w A + d_{w,o} + \varepsilon_{w,uy}. \end{aligned} \tag{5.9}$$

Comparing (5.9) with the R2R model in (5.5), we are able to know the system gain matrix $A = CB_1$, $d_{w,o} = x_{w,sp} B_2$ is the offset term, which cannot be manipulated by controllable variables, and the noisy term $\varepsilon_{w,uy} = \varepsilon_{w,ux} B_1 + \varepsilon_{w,xy}$. In the Section 5.2.2, three synthetic datasets will be firstly simulated based on the procedure in Fig. 5.11. In Section 5.2.3, various R2R controllers will give the suggested control settings of each run, and (5.9) can be used to compute the expected metrology values based on those settings.

Let the difference between the original and suggested control values be $\Delta_u = u'_w - u_w$. Based on (5.9), the expected metrology will be calculated as

$$y'_w = u'_w C B_1 + x_{w,sp} B_2 + \left(\varepsilon_{w,ux} B_1 + \varepsilon_{w,xy}\right)$$

$$= (u_w + \Delta_u)CB_1 + x_{w,sp}B_2 + \left(\varepsilon_{w,ux}B_1 + \varepsilon_{w,xy}\right)$$

$$= y_w + \Delta_u\, CB_1.$$

Since bridge variables are affected by controllable variables, the new control value will lead to the changes of their FDC values. Given (5.8), the FDC values of the bridge variables are updated as

$$x'_{w,br} = u'_w C + \varepsilon_{w,ux} = (u_w + \Delta_u)C + \varepsilon_{w,ux} = x_{w,br} + \Delta_u\, C.$$

With the updated FDC data, the new control values of the SRC and the VM-based controller can be computed based on the current conditions.

### 5.2.2 Simulation Procedures

The objective of the simulation case study is to assess the performance of the SRC under different types of disturbances. Three datasets with embedded disturbances will be generated according to the simulation procedures shown in Fig. 5.10.

*A. Design a universal structure*

As shown in Fig. 5.13, considering the MIMO system data with two input variables ($k = 2$) and two output variables ($\delta = 2$). Suppose there are 6 FDC variables observed in this process, denoted as $X = [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6] = [X_{br} \quad X_{sp} \quad X_{other}]$, where $X_{br} = [X_1 \quad X_2]$ is the matrix of bridge variables, $X_{sp} = [X_3 \quad X_4]$ is the matrix of support variables, $X_{other} = [X_5 \quad X_6]$ includes the system variables which are independent of the controllable variables and metrology. Three datasets with different disturbances, simulated via the disturbance variable $X_3$, will be generated based on this structure. To simplify the simulation procedure, we generate the summarized FDC variables instead of the temporal FDC raw data.

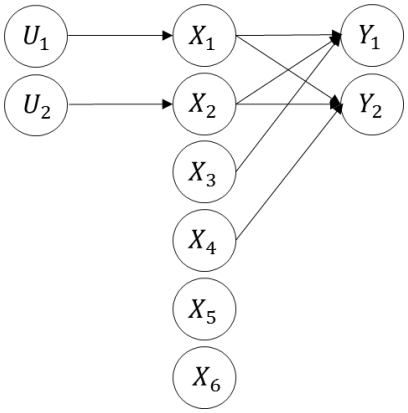


Figure 5.13 The relationship structure of the simulated MIMO system.

*B. Definition of the disturbances*

Generally, the process can be affected by different types of disturbances. Some common disturbances are listed in Table 5.7. The first type is the drift, usually caused by the machine or part degradation, and can be fixed by regular maintenance. The second type is the random impulse, which can be treated as an abnormal case. The last type is the shift. The machine

parameters may shift after the calibration during the maintenance activity. This change can cause the significant shift on the process output.

Table 5.7 The three types of disturbances.

| Disturbance | Drift | Impulse | Shift |
|---|---|---|---|
| Root cause | Machine or parts degradation. | Random case. | Parameters calibration. |
| Process output ($Y$) |  |  |  |

In the following paragraphs, the disturbances, denoted as $\eta$, will be embedded into one of the FDC variables.

### C. *Generate data*

Given the designed structure of the MIMO system and the definition of disturbances, the next step is to specify the configuration details. Table 5.8 shows the configuration of three desirable datasets. As described in Fig. 5.12, the simulation procedure will start with the controllable variables, followed by FDC variables. At the end, the metrology can be generated. The details will be explained below.

Table 5.8 The configuration of each dataset.

| Data | $\mathcal{D}_{drift}$ | $\mathcal{D}_{impulse}$ | $\mathcal{D}_{shift}$ |
|---|---|---|---|
| $\eta$ | $\eta_1$= (1,2, …,100,1,2, ...100, …) | $\eta_2 = (0,0, \dots,5,0, \dots,10,0, \dots)$ | $\eta_3 = (0,0, \dots,4,4, \dots,1,1, \dots)$ |
| $U$ | $U_1, U_2 \sim N(5,0.5)$ | | |
| $X_{br}$ | $X_{br} = [X_1 \quad X_2] = UC + \varepsilon_{uv}$ | | |
| $X_{sp}$ | $X_{3,ini} \sim N(5,0.5)$<br>$X_3 = X_{3,ini} + \eta_1$<br>$X_4 \sim N(5,0.5)$ | $X_{3,ini} \sim N(5,0.5)$<br>$X_3 = X_{3,ini} + \eta_2$<br>$X_4 \sim N(5,0.5)$ | $X_{3,ini} \sim N(5,0.5)$<br>$X_3 = X_{3,ini} + \eta_3$<br>$X_4 \sim N(5,0.5)$ |
| $X_{other}$ | $X_5, X_6 \sim N(5,0.5)$ | | |
| $Y$ | $Y = XB + \varepsilon_{xy} + \varepsilon_{ar1,t}$ | | |
| $\varepsilon_{ux}$ | $\varepsilon_{ux,1}, \varepsilon_{ux,2} \sim N(0,0.02)$ | | |
| $\varepsilon_{xy}$ | $\varepsilon_{xy,1}, \varepsilon_{xy,2} \sim N(0,0.02)$ | | |
| $\varepsilon_{ar1}$ | $\varepsilon_{ar1,t} = 0.5 \times \varepsilon_{ar,t-1}$ | | |
| $B$ | $\begin{bmatrix} 6 & -4 & 0.02 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0 & 0.1 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 & 1 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0 & 0.1 & 0 & 0 \end{bmatrix}$ | $\begin{bmatrix} 6 & -4 & 0.5 & 0 & 0 & 0 \\ 0.4 & 0.2 & 0 & 0.1 & 0 & 0 \end{bmatrix}$ |
| $C$ | $\begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ | | |

Considering the process has been operated for 600 runs and $w = 1, \ldots, 600$. The controllable variables $U = [U_1 \quad U_2]$ are firstly generated as $U_1 \sim N(5, 0.5)$ and $U_2 \sim N(5, 0.5)$. According to the control effect model in (5.7), the bridge variables, $X_{br} = UC + \varepsilon_{uv}$, can be generated given the mapping matrix $C = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$ and the control noise vector $\varepsilon_{ux} = [\varepsilon_{ux,1} \quad \varepsilon_{ux,2}]$, where $\varepsilon_{ux,1} \sim N(0, 0.02)$ and $\varepsilon_{ux,2} \sim N(0, 0.02)$.

As $X_3$ is assumed to be a disturbance variable which includes the disturbance vector $\eta$ and affects the process output $Y_1$. Let $X_3 = X_{3,ini} + \eta$, where $X_{3,ini} \sim N(0, 0.5)$. The disturbance variable $X_3$ for the different datasets is illustrated in Fig. 5.14, respectively. $X_4 \sim N(0, 0.5)$, which impacts on output $Y_2$.

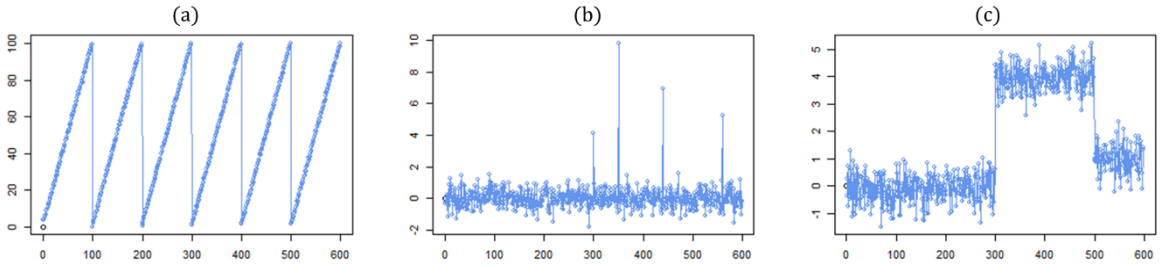


Figure 5.14 Three different simulated disturbances: (a) a drift; (b) sudden impulses; and (c) a shift.

The two FDC variables are generated as $X_5 \sim N(5, 0.5)$ and $X_6 \sim N(5, 0.5)$. By consolidating all the generated FDC variables, the outputs can be generated based on (5.6). Since the process output is usually correlated to the previous run, another autocorrelated error term $\varepsilon_{ar1,t}$ will be integrated as well. The process outputs are illustrated in Fig. 5.15. Let $B$ be the coefficient matrix of VM model, and $\varepsilon_{xy} = [\varepsilon_{xy,1} \quad \varepsilon_{xy,2}]$ be the prediction error $\varepsilon_{xy} = [\varepsilon_{xy,1} \quad \varepsilon_{xy,2}]$, where $\varepsilon_{xy,1} \sim N(0, 0.05)$ and $\varepsilon_{xy,2} \sim N(0, 0.05)$. As shown in (5.7), the coefficient matrix can be decomposed into $B_1$ and $B_2$ for bridge variables and support variables, respectively.

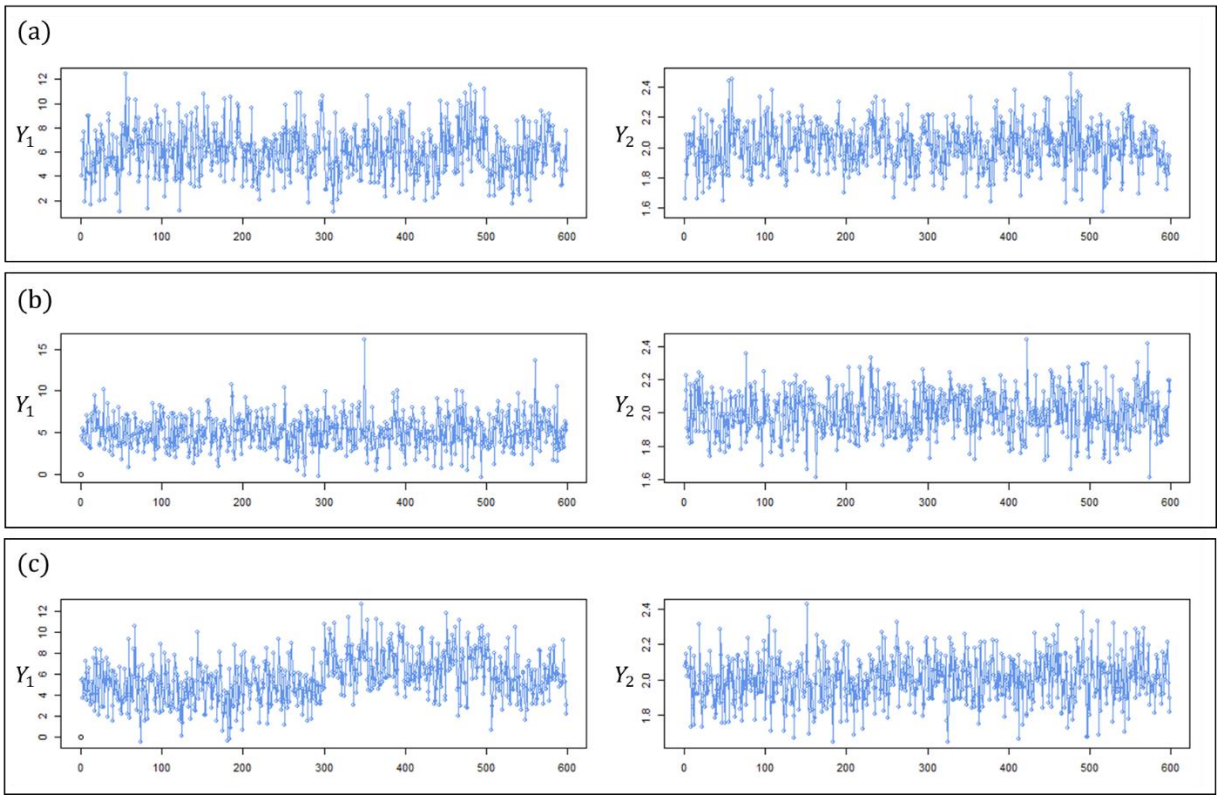


Figure 5.15 Process output of each dataset after introducing the (a) drift; (b) impulses; and (c) shift.

By combining all the variables above to be $\mathbb{D} = \{\mathcal{D}_{drift} \quad \mathcal{D}_{impulse} \quad \mathcal{D}_{shift}\}$, where each dataset $\mathcal{D}$ in $\mathbb{D}$ is a 600 by 10 matrix, three datasets are ready for evaluation.

$$\mathcal{D} = [U_1 \quad U_2 \quad X_1 \quad \cdots \quad X_6 \quad Y_1 \quad Y_2]. \tag{5.10}$$

If the two-time-slice DBN is considered, all variables with one-period lag should be included as well (see Section 4.2.4). The combined dataset for SRC is denoted as $\mathbb{D}_{DBN}$, which consists of the disturbed datasets $\mathcal{D} = [\mathcal{D}^{(t-1)} \quad \mathcal{D}^t]$.

Following the settings above, the R2R model in (5.9) can be deduced, where the system gain matrix $A = CB_1 = \begin{bmatrix} 3 & 0.2 \\ -2 & 0.1 \end{bmatrix}$ can be derived. The error term $\varepsilon_{uy}$ is a composited noise where $\varepsilon_{uy} = \varepsilon_{ux}B_1 + \varepsilon_{xy}$. As discussed in Section 5.2.1, the R2R model is not used for data generation but for evaluating the control efficiency.

### 5.2.3 Controller Settings

In this case study, the efficiency of four controllers: the dEWMA controller (Butter and Stefani, 1994), the PLS-RC (Khan et al. 2008), the SRC (Section 4.7), and the A-SRC (Section 4.8), will be examined. The necessary settings to complete the study will be specified in the following paragraphs, and the process flows of these controllers are illustrated in Appendix A.

*A. General settings*

For each dataset $\mathcal{D}$ in $\mathbb{D}$, the first 200 wafers, defined as $\mathcal{D}_1$, will be used for model training, and the rest, denoted as $\mathcal{D}_2$, will be used for model testing. For the testing wafers in

$\mathcal{D}_2$, assume that the metrology is measured every 12 runs (see Fig. 5.16). This means that variable $Y$ of the regular run is masked during the control value computation and will be revealed for evaluating the controller efficiencies. As defined in the previous subsection, Instead of $\mathbb{D}$, the dataset $\mathbb{D}_{DBN}$ will be applied to the A-SRC with the same settings as $\mathbb{D}$, wherein the training set is $\mathcal{D}_{1,DBN}$ and testing set is $\mathcal{D}_{2,DBN}$.

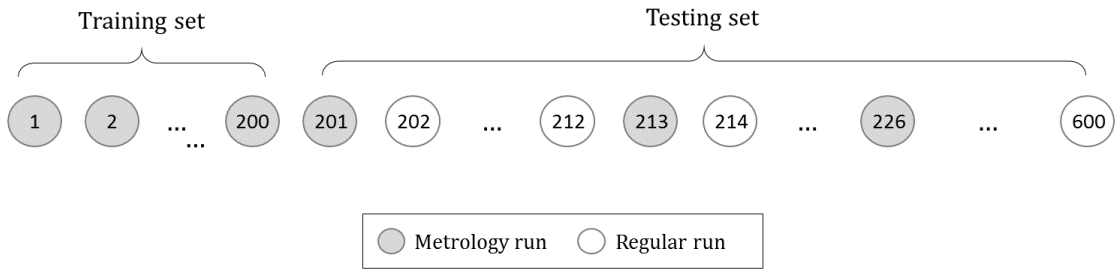


Figure 5.16 The metrology run takes the measurements every 12 regular runs.

*B. dEWMA controller Setting*

The dEWMA controller aims to estimate the process trend by considering a double exponential filter. The process flow is presented in Appendix A. In this case study, the discount factors for the filter is defined as $\lambda = (\lambda_1, \lambda_2) = (0.35, 0.65)$. The setting is referred to that in Khan's study (Khan et al, 2008).

*C. PLS-RC Setting*

In this case study, if the previous wafer is a metrology run, the control value for the next run will be computed based on the actual metrology with $\lambda = (\lambda_1, \lambda_2) = (0.35, 0.65)$. If the previous wafer is a regular run, a predicted value will be used in the filter with $\lambda = (\lambda_1, \lambda_2) = (0.65, 0.55)$ (Khan et al. 2008).

*D. SRC and A-SRC Setting*

A DBN is learned based on training set $\mathcal{D}_1$ and a blacklist $\mathcal{L}$. Note that the association matrix $\mathcal{M}$ is not considered in this simulation study. Only the time-dependency configuration and inter-slice configuration will be considered. The length of the alarm window is set to be 10, and the threshold $h_{risk}$ is 0.8. If the risk level of $h_k$ exceeds $h_{risk}$, the local update of $V_k$ will be triggered.

## 5.2.4 Controller Evaluation

Given the three types of dataset and the above setting, the controllers can be evaluated. The efficiency of controllers is assessed by comparing the improving percentage of the MSE of $\mathcal{D}_2$ of two process outputs, $Y_1$ and $Y_2$. The results are presented in the following sections.

*A. Drift*

Fig. 5.17 shows that expected output of four controllers under the drift disturbance. Since the actual measurements are only available every 12 runs, without considering real-time equipment information, the dEWMA controller is not able to quickly adjust the control value

for the drifted output $Y_1$, while the three other controllers incorporating the equipment information perform quite well. As shown in Table 5.9, both SRC and A-SRC controller outperform the other two controllers for output $Y_1$ and $Y_2$ in terms of reduced MSE. Since the likelihood monitoring of A-SRC did not enable model update (see Fig. B1 in Appendix B), the results of SRC and A-SRC are the same.

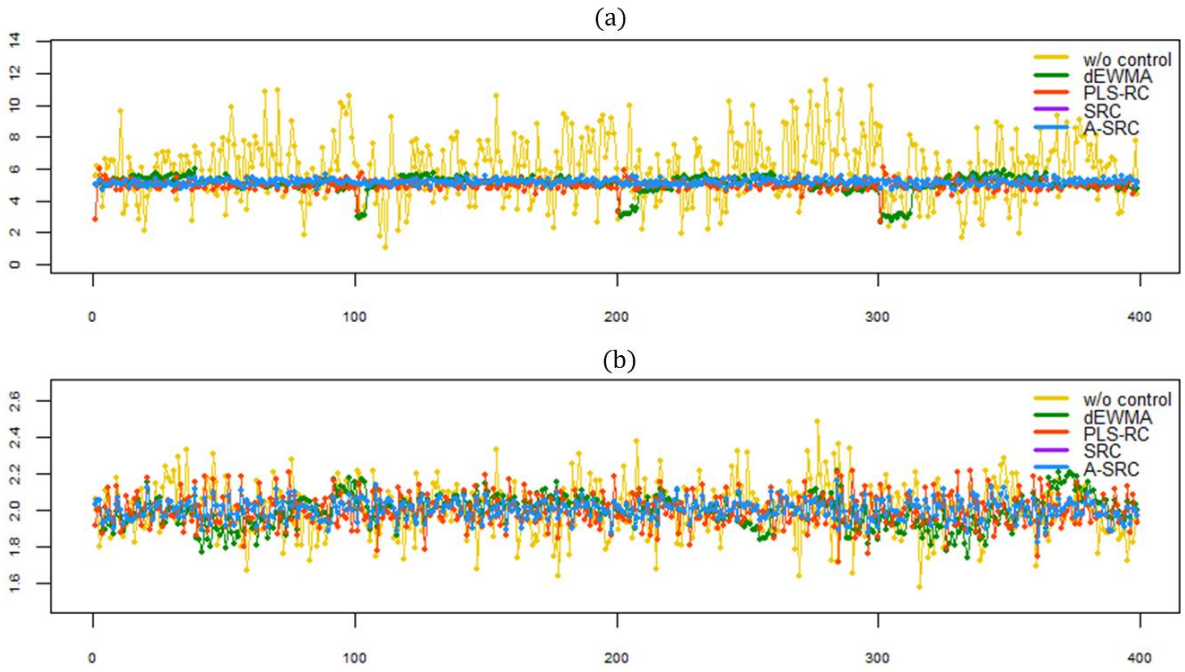


Figure 5.17 The expected metrologies of (a) $Y_1$ and (b) $Y_2$ of the testing sets under the drift disturbance.

Table 5.9 The performance evaluation of the drift disturbance.

| Improve % MSE of $\mathcal{D}_2$ | dEWMA | PLS-RC | SRC | A-SRC |
|---|---|---|---|---|
| $Y_1$ | 92.98% | 97.88% | 98.72% | 98.72% |
| $Y_2$ | 61.66% | 59.57% | 85.10% | 85.10% |
| average | 77.32% | 78.73% | 91.91% | 91.91% |

*B. Impulse*

Fig. 5.18 shows that the expected output for four different control schemes under the impulse disturbance. All controllers cannot prevent randomly occurred cases. Furthermore, the control value of PLS-RC for the next wafer is misled. The reason is that the PLS-RC is capable of estimating the metrology of impulse case, but it does not gauge if this case is appropriate to be the input of control setting computation for the next wafer. Since the SRC replaces the missing metrology $Y^{(t-1)}$ with the mean level of historical data (See Section 4.6.1), it can avoid the misled control due to a random case. The results of SRC and A-SRC are the same as shown in Table 5.10, because such random disturbance is not sustained so that the monitoring mechanism does not trigger the model update (see Fig. B2 in Appendix B). By comparing the

improved MSE of both $Y_1$ and $Y_2$, the SRC and A-SRC again outperform the other two controllers.

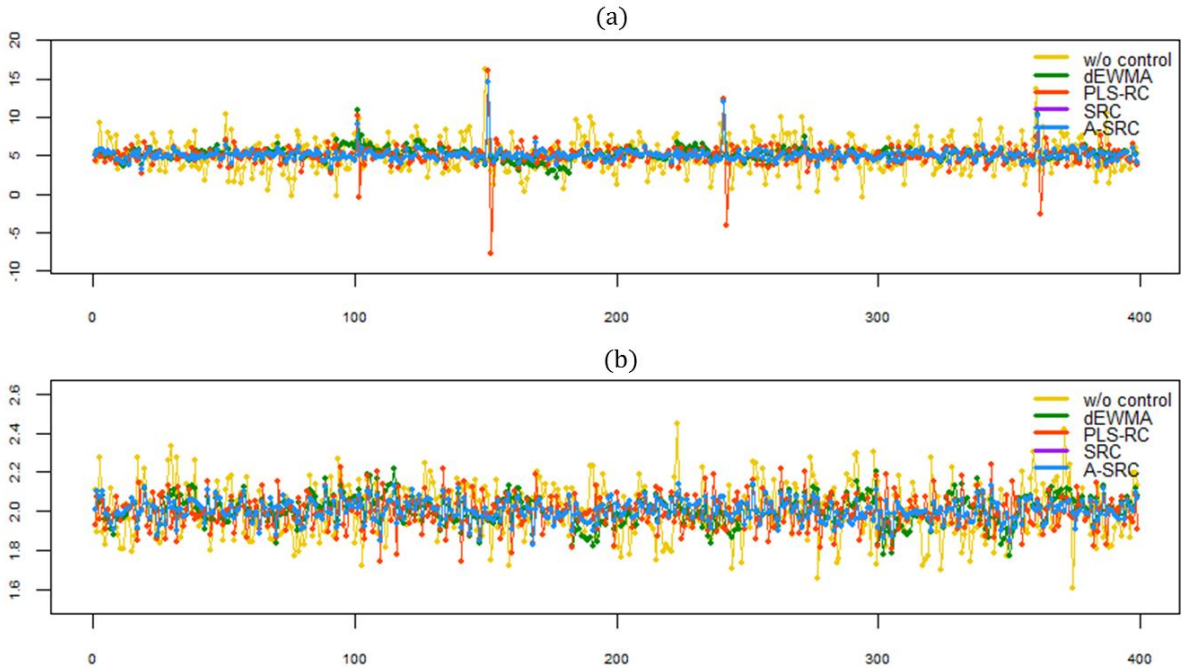


Figure 5.18 The expected metrologies of (a) $Y_1$ and (b) $Y_2$ of the testing sets under the impulse disturbance.

Table 5.10 The performance evaluation of the impulse disturbance.

| Improve % MSE of $\mathcal{D}_2$ | dEWMA | PLS-RC | SRC | A-SRC |
|---|---|---|---|---|
| $Y_1$ | 75.22% | 54.46% | 83.70% | 83.70% |
| $Y_2$ | 69.00% | 58.95% | 83.27% | 83.27% |
| average | 72.11% | 56.71% | 83.49% | 83.49% |

*C. Shift*

Fig. 5.19 shows the expected outputs of the four different control schemes under the shift disturbance. The SRC fails to correct the shit process output $Y_1$ as shown in Fig. 5.19a, while dEWMA and PLS-RC are able to deal with such disturbance. Since the mean level of one variable has changed, the DBN no longer presents well the current equipment status. Therefore, the control settings based on SRC are outdated. However, the A-SRC can give the most efficient control on the two outputs (see Table 5.11). The monitoring mechanism embedded in A-SRC is able to detect the shifting distribution and enable to locally update the coefficients of disturbed variable $X_3$. The likelihood monitoring of each variable is shown in Fig. B.3 in Appendix B. As a result, A-SRC demonstrates the high efficiency of control by providing the timely control setting, which fits well the current equipment status.

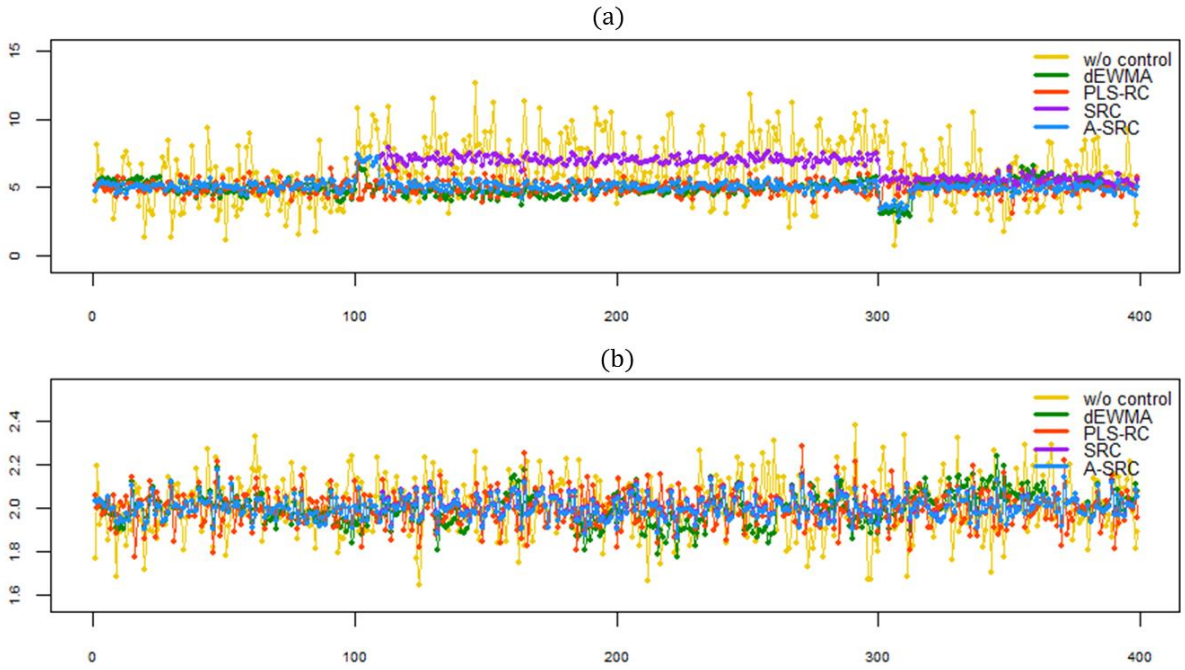


Figure 5.19 The expected metrologies of (a) $Y_1$ and (b) $Y_2$ of the testing sets under the shift disturbance.

Table 5.11 The performance evaluation of the shift disturbance.

| Improve % MSE of $\mathcal{D}_2$ | dEWMA | PLS-RC | SRC | A-SRC |
|---|---|---|---|---|
| $Y_1$ | 94.38% | 96.02% | 62.90% | 96.20% |
| $Y_2$ | 65.48% | 60.76% | 84.13% | 84.37% |
| average | 79.93% | 78.39% | 73.52% | 90.29% |

*D. Comparison*

Based on the experiments presented in this section, the proposed SRC and A-SRC show their capability of dealing with the process under different disturbances. The differences between those controllers can be discussed in several aspects (see Table 5.12). All controllers are capable of controlling the SISO and the MIMO systems. Two data-driven based controllers: SRC and PLS-RC, incorporate the real-time equipment information, can timely provide the critical control even when the actual metrology is unknown. Both controllers include the VM models so that they can predict the measurement of the regular run. The SRC further considers one step ahead, the latest FDC data are not only used for the VM model, but also utilized to predict the equipment conditions for next run. Therefore, the control value of the next run will be calculated more effectively. As the A-SRC further incorporates a likelihood monitoring mechanism, the model can be slightly adjusted which better fits the current equipment state. The results of likelihood monitoring of this case study are presented in Appendix B.

Table 5.12 The similarities and differences among the three controllers.

| Controller | dEWMA | PLS-RC | SRC/A-SRC |
|---|---|---|---|
| SISO/MIMO | Yes | Yes | Yes |
| Require training model | No | Yes | Yes |
| Include EQPT info | No | Yes | Yes |
| Missing metrology of a regular run | Skip | $\hat{y}_{w-1}$=$f_{VM-PLS}(x_{w-1})$ | $\hat{y}_{w-1}$=$f_{VM-DBN}(x_{w-1})$ |
| EQPT condition prediction | No | No | $\hat{x}_w \leftarrow x_{w-1}$ |
| Control (when $w-1$ is a metrology run) | $u_w \leftarrow y_{w-1}$ | $u_w \leftarrow y_{w-1}$ | $u_w \leftarrow (y_{w-1}, x_{w-1}, \hat{x}_w)$ |
| Control (when $w-1$ is a regular run) | $u_w$=$u_{w-1}$ | $u_w \leftarrow \hat{y}_{w-1}$ | $u_w \leftarrow (\bar{y}, x_{w-1}, \hat{x}_w)$ |

# 6 Conclusions and Perspectives

## 6.1 Conclusions

In this thesis, a holistic framework for process control in the semiconductor sector is proposed. The thesis starts with an overview of the integrated control framework and then drills down to the individual function modules. In Chapter 1, background of process control in semiconductor manufacturing is given, from conventional control systems to advanced data-driven solutions. Then, the motivation and scope of the thesis are established, regarding the development of an integrated control framework that is able to consolidate numerous models. It is based on a Dynamic Bayesian Network (DBN), in order maximize the utilization of information, by combine multiple data sources with existing domain knowledge. Based on such structured model, the function modules can be constructed with a more transparent view of process interactions, including process monitoring, Virtual Metrology (VM), and Run-to-Run (R2R) control. The state-of-the-art of such applications was reviewed in Chapter 2. The current process monitoring solutions focus on improving the efficiencies of both fault detection and fault diagnosis, under high-dimensional scenarios. Many researchers have investigated different machine learning models for VM, and it is possible to verify that choosing the best model with the smallest prediction error is highly case dependent. Due to the dynamic nature of the process, the maintenance of VM models can be challenging too. Several R2R controllers have been proposed to deal with complex processes. Control systems based on the VM model have been studied as well. As the core of the proposed framework is a DBN model, the theoretical background of Bayesian Networks was provided in Chapter 3, including types of networks, inference methods, and learning algorithms. The details of the proposed framework were presented in Chapter 4. Starting with the online stage, which aims to consolidate the available information and construct the DBN model. Then, the DBN model can be adopted in the scope of different applications, in the online stage. A DBN-based monitoring mechanism can be adopted without requiring complex computations. Based on a DBN, the underlying VM model can be simply extracted. By identifying a sub-network, a Structured R2R Controller (SRC) can be obtained, which includes all causalities involving in the control problem. By considering the nature of equipment aging or process drift, an Advanced-SRC (A-SRC) is proposed, which incorporates a local updating mechanism. To validate the efficiency of the propose framework, some case studies were presented in Chapter 5. An industrial case was employed to evaluate the performance of the VM model, SRC and A-SRC. To further assess the capability of the proposed controller under various disturbances, synthetic datasets were generated and employed. The evaluation was done by comparing the performance of the proposed controllers with several alternatives controllers. Based on the results of the case study, we can conclude that the proposed framework provides efficient predictions and stable control. Furthermore, this physics-informed framework also gives a better interpretable model that enhances the existing data-driven approach.

The contributions of this research can be analyzed from two perspectives: the framework and its applications. By looking at the overall framework, one can verify that it successfully unifies various function modules in a single integrated framework. This framework allows modules to be operated either together or independently. A DBN is employed as the fundamental model core of the framework, which is able to consolidate

multiple data sources and incorporate domain knowledge. This is a novel approach for process control in semiconductor manufacturing, as the existing solutions mainly focus on pure data-driven methods. As a DBN is presented in a structured form, the process interactions can be easily illustrated, which makes the DBN superior to other approaches in terms of interpretability. Furthermore, a DBN can be decomposed to different sub-networks depending on the tasks, and this characteristic enables a DBN to support various applications. Therefore, works requiring different sub-models, including data pre-processing and model learning, can be addressed by the same data pre-processing and model learning procedure.

The contributions of each function module are listed in Table 6.1, where DBN-based modules are compared with the conventional approaches. A DBN consolidates multiple information sources so that its applications can consider more information than what has been done in the past. The proposed monitoring mechanism not only can monitor the behavior of the univariate but also can detect the change of relationships among variables, while the conventional approaches focus on monitoring either univariate or compressed features. The traditional VM approaches concentrate on finding the important variables that affect the process output. A structured VM model starts with connecting all process variables and discovers all conditional dependencies. With a global view, the direct impact and indirect impact of the process output can be clearly presented. In this way, the important variables of the VM model can be identified, as well as their indirect impacts. Conventional R2R controllers primarily rely on metrology data; some equipment information may be taken into account through VM models. The proposed SRC considers more equipment conditions related to the control decision.

The causal structure inherent to DBN makes it superior to the conventional approaches in terms of interpretability. Many process monitoring methods compress the features for detection and later decompose those features for diagnosis. With a causal structure, the fault diagnosis can be accomplished without complex computation. Instead of accuracy-oriented VM methods that might neglect the physical meanings, a structured VM model provides a better interpretation of the process interactions. The conventional R2R controllers focus on the relationships between controllable variables and metrology variables. When a controller incorporates a VM model, the relationships between equipment state and metrology are taken into account as well. An SRC can further consider how the control action affects the equipment states, which has not been contemplated in past studies.

In summary, we proposed an *Integrated Physics-Informed Control Framework*, which consolidates various process information sources and can be applied to different applications. By leveraging more information and considering physical causalities, we are able to enhance the existing process control systems. The effectiveness of this framework has been demonstrated through case studies. In this thesis, we introduced this new framework to deal with the process control problems, but this new idea requires further work in the future before moving to the shop floor. Besides, the proposed framework should be able to scale to broader scopes. These topics can be interesting areas for future research. More details will be presented and discussed in the next section.

Table 6.1 The thesis contributions in terms of function modules.

| Contributions | Application | Differentiation | |
|---|---|---|---|
| | | Conventional approaches | DBN-based module |
| Leverage multiple information sources | Monitoring | • Univariate<br>• Compressed features | • Univariate<br>• Correlations |
| | VM | • Direct effects | • Direct effects<br>• Indirect effects |
| | R2R | • Previous measurement<br>• Equipment condition related to the measurement (VM) | • Previous measurement (depending on the structure)<br>• Equipment condition related to the measurement (VM)<br>• Equipment condition related to control decision |
| Employ a causal structure | Monitoring | • Feature compression for fault detection<br>• Feature decomposition for diagnosis | • Global likelihood for monitoring<br>• Local likelihood for diagnosis |
| | VM | • Results may lack physical meaning | • The result is consistent with available physical knowledge |
| | R2R | • How metrology is affected by control setting<br>• How metrology is affected by equipment states | • How metrology is affected by control setting<br>• How metrology is affected by equipment states<br>• How equipment state is affected by control settings |

## 6.2 Perspectives

In this research, we introduce an Integrated Physics-Informed Control Framework and present its advantages. Through both simulated examples and the industrial case study presented in Chapter 5, the effectiveness of the proposed framework has been validated and therefore it can be extended to a larger scale. However, some assumptions made in the thesis may require some adaptation to fit real industrial environments. In this section, we would like to address these challenges and future opportunities. Following the diagram in Fig 4.1, the future work under each block is discussed. Finally, a vision of the future framework, with a broader scope, will be presented.

### 6.2.1 Offline Stage

*A. Historical data pre-processing*

The procedure for transforming temporal FDC data into wafer-based data is presented in Section 4.2.1. This procedure is the most common approach for FDC analysis, which takes the statistics of specific windows. The appropriate statistics should be determined by the profile of the raw temporal data (for example, taking the average for a steady profile and taking the

slope for an incremental profile). Many approaches for extracting the appropriate features from temporal data have been proposed in literature and commercial solutions (Moyne et al., 2015; Rendall et al., 2017). In this thesis, we did not focus on investigating feature generation and extraction. Nevertheless, the efficiency of the VM model and SRC have been validated through the case study presented in Section 5.1, where only average and standard deviation are taken into account. Since only two types of statistics may not be sufficient to describe the behavior of the process, we suggest that the future framework should incorporate a feature-oriented approach to extract the indicators (Rendall et al., 2017).

*B. Knowledge-based configuration*

In this thesis, a physics-informed framework is proposed by introducing a blacklist defined based on domain knowledge (see Section 4.3). This blacklist is employed to prevent the appearance of infeasible anti-causalities in the network. In this context, the learning of process causality is more data-driven than knowledge-driven. For cases with plenty of unknown causalities, this approach can be advantageous. On the other hand, for some cases with less unknown causalities, i.e., most causalities have been validated through experiments or SME, it is also possible to introduce a white list for structure learning so that the network will include more knowledge-driven edges. The edges on a whitelist are fixed before the structure learning and these edges cannot be eliminated even when this movement can obtain a higher score. Furthermore, if the coefficients for described the causalities have been established, parameter learning can be skipped as well. In summary, the proportion of data-driven learning or knowledge-driven specification of the network can be tailored one a case by case basis.

*C. Structure learning*

In this thesis, the simplest search algorithm – Hill Climbing (HC) was employed. Since the objective of this thesis is to demonstrate the capability of DBN as an informatics tool, we did not focus on addressing the efficiency of the learning algorithm. Nevertheless, the issue of choosing a better and more efficient learning algorithm shall be investigated in the future. Several search algorithms have been studied, such as Tabu search, simulated annealing, genetic algorithm, and some hybrid algorithms combining constraints-based and score-based approaches have been proposed as well (Larranaga et al., 1996; Tsamardinos et al., 2006). The performance comparison between the proposed two-phase learning method and the conventional approach was not addressed in our assessment, as we focus more on the efficiency of the function modules. But the investigation across different learning algorithms regarding the process control problem surly can be a future study as well. Furthermore, the future framework may consider a more complicated structure which will be later on described in Section 6.2.3. A more sophisticated learning procedure should be developed to meet the need.

### 6.2.2 Online Stage

*A. Process monitoring*

In this thesis, we demonstrate that likelihood can be used for detecting abnormalities, and a simple mechanism was proposed for updating the model (see Section 4.5). However, it may not be sufficient for real production lines. As mentioned in Section 4.8, if the magnitude of process disturbance is large, updating the model may mask the underlying fault. Besides,

different disturbances should require different treatments, which was not address in this thesis. Several researchers have investigated solutions for detecting and handling various disturbances (Sachs and Hu, 1995; Wang & He, 2007). A brief review can be found in Section 2.4.2. We suggest that the future framework should incorporate these solutions to better fit the needs of the industry.

*B. Prognosis*

As mentioned in Section 3.6, to simplify the computations, we assume that all variables are continuous and following Gaussian distributions, and the relationships among variables are linear. Therefore, the inference method for GBNs was employed in this thesis. However, such assumptions should be relaxed for a more general framework, which is capable of handling both continuous variables and discrete variables (Madsen & Jensen, 1999; Cowel, 2005; Salmerón et al.,2018). Several approximate inference methods can be used for a more general network, such as when the relationships among variables are non-linear, or the distributions of variables are not identical. A brief review of these inference methods has been given in Section 3.3.

*C. R2R Control*

In semiconductor manufacturing, some advanced process equipment implement Integrated Metrology (IM) tools, where the measurements can be obtained in the same machine. This kind of technology provides better real-time process control and monitoring. As IM tools haven't been widely implemented in every equipment, most measurements are still collected from stand-alone metrology tools. Consequently, most processes metrology present significant delays (Khan et al., 2008). The delay may be caused by the transporting time and queuing time. Conventional R2R controllers highly depend on metrology data as the input. Thus, the level of metrology delay can affect the performance of controllers. The proposed SRC primary considers equipment information. The measurement of the previous run can be an input as well, which depends on the structured learned from data. When the actual metrology is not available, the predicted one will be used in substitution. Therefore, comparing to conventional controllers, the impact of metrology delay on SRC should be smaller. We did not address this effect in this thesis, but it surely should be taken into account and assessed through more experiments.

### 6.2.3 The Scope of the Network

*A. Cross-Equipment or cross-product*

Various data sources are related to process control in semiconductor manufacturing as shown in Fig1.1. In this thesis, we focus on the data collected from process equipment and metrology tools. To construct a more general model, more information should be included as well, such as equipment or product class. The context information is usually in the form of discrete variables. Since the fundamental model, DBN, can handle both continuous variables and discrete variables, the original DBN can be extended to a broader scope as shown in Fig 6.1.

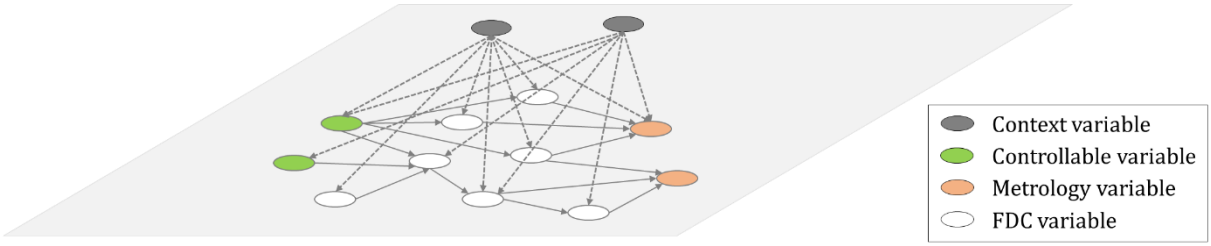


Figure 6.1 A hybrid network consists of both continuous variables and discrete variables.

As described in Section 2.4.2, R2R control can be challenging in high-mix semiconductor manufacturing processes, because the conventional controllers are usually designed for particular equipment and product type. Many researchers have investigated the solutions for such an environment (Firth et al., 2006; Wang et al., 2009; Tan et al., 2015). With a hybrid DBN, the SRC can be employed in the high-mix production line. To be more specific, assume that the hybrid network in Fig 6.1 is a Conditional Linear Gaussian Bayesian Network (CLGBN) (see Section 3.2.3). Let the context variables be equipment type and product group, denoted as $C_1$ and $C_2$, respectively. Assume an FDC variable $X_i$ is affected by both context variables and another variable $X_j$. Then, the local distribution of $X_i$ is expressed as $f(X_i|C_1 = c_1, C_2 = c_2, X_j = x_j) = \mathcal{N}(\alpha_{c_1,c_2} + \beta_{c_1,c_2}X_j, \sigma_{c_1,c_2})$. After defining the conditional local distribution, the approach presented in this thesis can be applied. Either the VM model or SRC can switch between different equipment and product types while sharing the same universal structure.

*B. Multiple Process*

Although only one process operation is considered in this thesis, the network can be extended to multiple processes. An example is shown in Fig. 6.2, where two sequential processes are included in the network with two time-slices. Two slices at the same vertical axis indicate different wafers in the same process. The two slices at the same horizontal axis indicate the same wafer passes through two sequential processes. In some cases, the variability of the metrology can be affected by the metrology of its former process. For example, the CD measurement after the etching process is influenced by the CD after the lithography process. This effect can be presented by linking the two metrology variables of different processes (see Fig. 6.2).

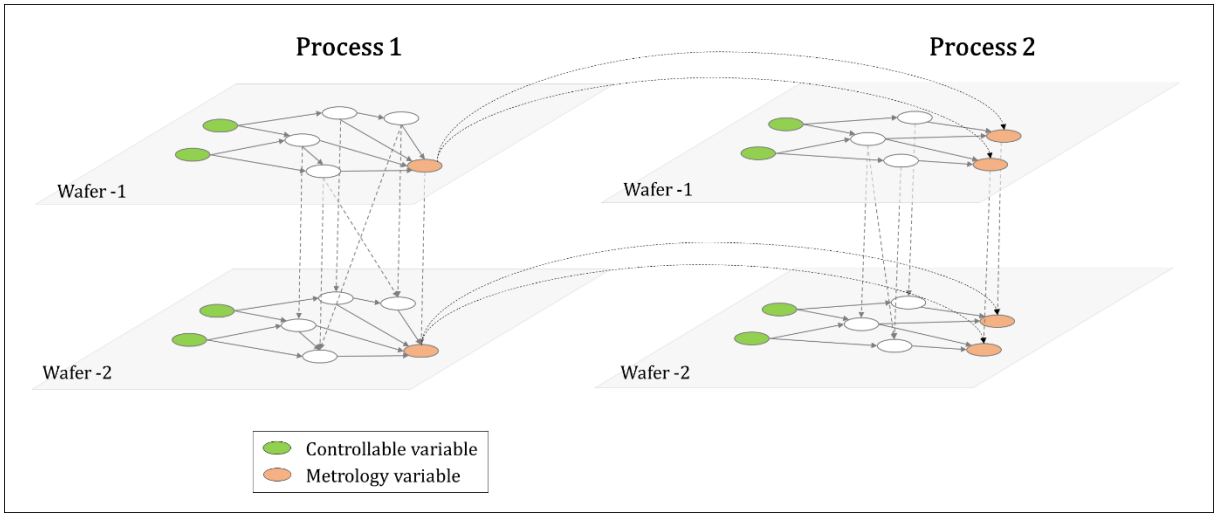


Figure 6.2 A DBN consists of two sequential processes.

A structured R2R controller based on such a cross-process DBN is presented in Fig. 6.3. The green line means that the metrology data of the previous wafer are taken into account for the control setting computation, which can be considered as feedback control. The blue line shows that the pre-measurement is included, which equivalent to feed-forward control. Therefore, a cross-process DBN can present more complex interaction among processes and provide the possibility for more delicate control.

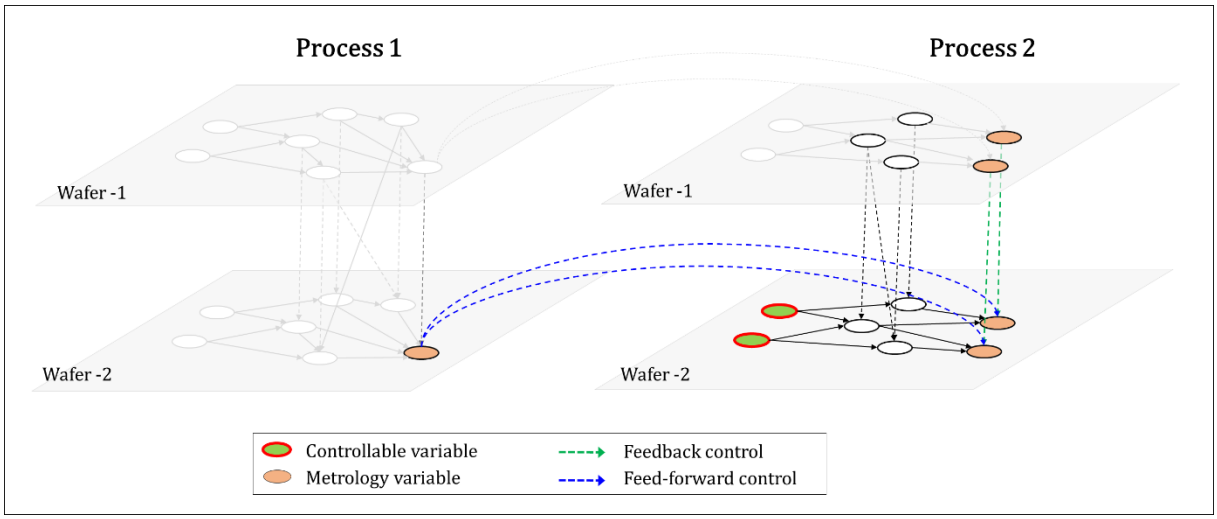


Figure 6.3 A SRC based on a cross-process DBN.

## *C. Multiblock*

In this research, all variables are treated as individual variables. However, considering their physical characteristics, we may categorize them into several variable groups, where the variables in the same group share similar properties. This type of approaches are called multiblock methods, and have been explored in the literature (Campos et al., 2017). The edges of the multiblock DBN indicate the connections between groups instead of single variables. Besides, the edges within the block can be either directed or undirected,depending on their relationships. An example is illustrated in Fig. 6.4. We expect that such a multiblock DBN can fit better the physical nature of the process and provide more straightforward interpretation.

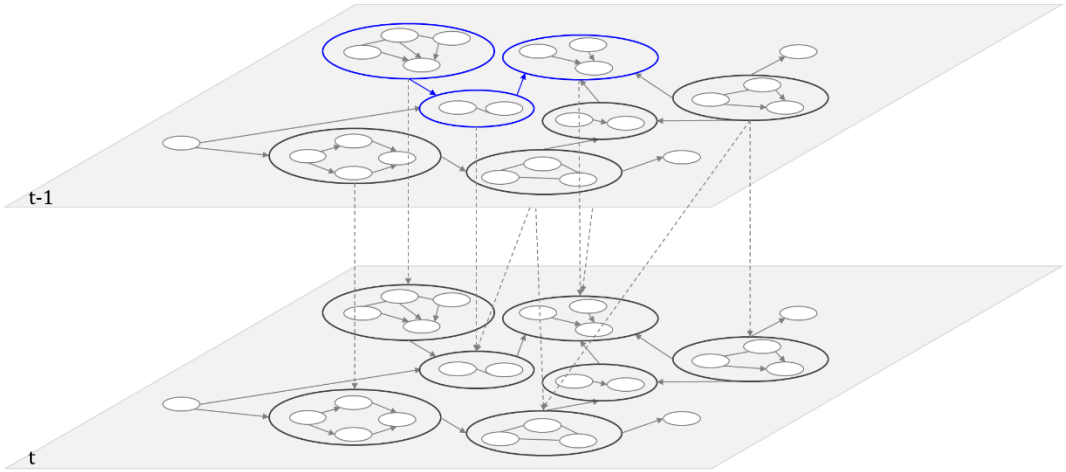


Figure 6.4 An example of a multiblock DBN.

## *D. Consolidation*

By consolidating all the extensions above, the future framework can encompass a huge network (see Fig 6.5). This network links all information during the wafer fabrication, which can possibly be used for global control decision making and future yield analysis. The potential value of such framework is very high for a company, and therefore its development should be considered as a strategic asset and differentiating advantage.

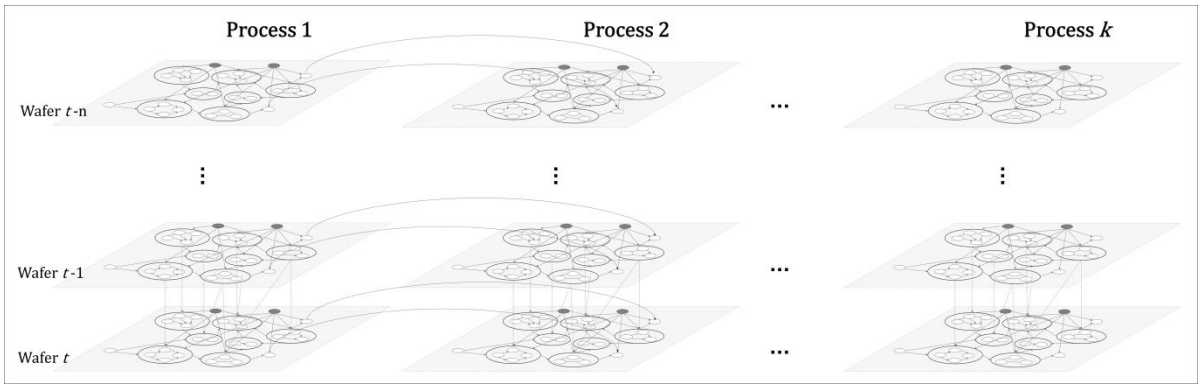


Figure 6.5 An example of the future network.

# Bibliography


Adivikolanu, S., and Zafiriou, E. (2000). Extensions and performance/robustness tradeoffs of the EWMA run-to-run controller by using the internal model control structure. IEEE Transactions on Electronics Packaging Manufacturing *23*, 56–68.

Andersen, C.M., and Bro, R. Variable selection in regression—a tutorial. Journal of Chemometrics *24*, 728–737.

Arnborg, S., and Proskurowski, A. (1989). Linear time algorithms for NP-hard problems restricted to partial k-trees. Discrete Applied Mathematics 23, 11–24.

Bertsekas, D. P. (1982). Constrained Optimization and Lagrange MultiplierMethods. New York: Academic.

Besnard, J., Gleispach, D., Gris, H., Ferreira Porto Rosa, A., Roussy, A. agnès, Kernaflen, C., and Hayderer, G. (2012). Virtual Metrology Modeling for CVD Film Thickness. International Journal of Control Science and Engineering *2*, 26–33.

Biton, T., and Ratner, H. (2005). Multivariate fault detection (MVFD) EP/FDC implementation. In ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005., pp. 377–380.

Blue, J., Roussy, A., Thieullen, A., and Pinaton, J. (2012). Efficient FDC based on hierarchical tool condition monitoring scheme. In 2012 SEMI Advanced Semiconductor Manufacturing Conference, pp. 359–364.

Boning, D.S., Moyne, W.P., Smith, T.H., Moyne, J., Telfeyan, R., Hurwitz, A., Shellman, S., and Tayor, J. (1996). Run by run control of chemical-mechanical polishing. IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C *19*, 307–314.

Botre, C., Mansouri, M., Nounou, M., Nounou, H., and Karim, M.N. (2016). Kernel PLS-based GLRT method for fault detection of chemical processes. Journal of Loss Prevention in the Process Industries *43*, 212–224.

Box, G., & Jenkins, M. (1974). Time series analysis—forecasting and control. Oakland, CA: Holden-Day.

Boyen, X., and Koller, D. (2013). Tractable Inference for Complex Stochastic Processes. ArXiv:1301.7362 [Cs].

Bunkofske, R., Colt, J., McGill, J., Pascoe, N., Surendra, M., Taubenblatt, M., and Ghias, A. (2003). User configurable multivariate time series reduction tool control method.

Butler, S.W., and Stefani, J.A. (1994). Supervisory run-to-run control of polysilicon gate etch using in situ ellipsometry. IEEE Transactions on Semiconductor Manufacturing *7*, 193–201.

Cai, B., Huang, L., and Xie, M. (2017). Bayesian Networks in Fault Diagnosis. IEEE Transactions on Industrial Informatics *13*, 2227–2240.

Campos, L.M.D. (2006). A Scoring Function for Learning Bayesian Networks Based on Mutual Information and Conditional Independence Tests. J. Mach. Learn. Res. *7*, 2149–2187.

Campos, M.P., Sousa, R., Pereira, A.C., and Reis, M.S. (2017). Advanced predictive methods for wine age prediction: Part II – A comparison study of multiblock regression approaches. Talanta 171, 132–142.

Chang, Y.-C., and Cheng, F.-T. (2005). Application development of virtual metrology in semiconductor industry. In 31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005, pp. 124-129.

Charitos, T., van der Gaag, L.C., Visscher, S., Schurink, K.A.M., and Lucas, P.J.F. (2009). A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients. Expert Systems with Applications *36*, 1249–1258.

Chen, A., and Guo, R.-S. (2001). Age-based double EWMA controller and its application to CMP processes. IEEE Transactions on Semiconductor Manufacturing *14*, 11–19.

Chen, J., and Wang, F. (2007). PLS based dEWMA run-to-run controller for MIMO non-squared semiconductor processes. Journal of Process Control *17*, 309–319.

Chen, J., Jia, H., Huang, Y., and Liu, D. (2012). Learning the structure of Dynamic Bayesian Network with domain knowledge. In 2012 International Conference on Machine Learning and Cybernetics, pp. 372–375.

Chen, P., Wu, S., Lin, J., Ko, F., Lo, H., Wang, J., Yu, C.H., and Liang, M.S. (2005). Virtual metrology: a solution for wafer to wafer advanced process control. In ISSM 2005, IEEE International Symposium on Semiconductor Manufacturing, 2005., pp. 155–157.

Cheng, J., and Druzdzel, M.J. (2000). AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks. Jair *13*, 155–188.

Cheng, F., Chen, Y., Su, Y., and Zeng, D. (2008). Evaluating Reliance Level of a Virtual Metrology System. IEEE Transactions on Semiconductor Manufacturing *21*, 92–103.

Cherry, G.A., and Qin, S.J. (2006). Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. IEEE Transactions on Semiconductor Manufacturing *19*, 159–172.

Chiang, L.H., Russell, E.L., and Braatz, R.D. (2000). Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis. Chemometrics and Intelligent Laboratory Systems *50*, 243–252.

Chickering, D.M., Geiger,D., Heckerman, D. (1994) Learning Bayesian networks is NP-hard. Technical ReportMSR-TR-94-17, Microsoft Research, Microsoft Corporation.

Chickering, D.M. (2002). Optimal Structure Identification With Greedy Search. Journal of Machine Learning Research *3*, 507–554.

Cofiño, A.S., Cano, R., Sordo, C., and Gutiérrez, J.M. (2002). Bayesian networks for probabilistic weather prediction. In In Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002, (Press), pp. 695–699.

Cooper, G.H. (1990). Bayesian belief-network inference using recursive decomposition. p.

Cowell, R. (2005). Local Propagation in Conditional Gaussian Bayesian Networks. Journal of Machine Learning Research *6*, 1517–1550.

Cowell, R.G., Dawid, P., Lauritzen, S.L., and Spiegelhalter, D.J. (1999). Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks (New York: Springer-Verlag).

Darwiche, A. (2009). Modeling and Reasoning with Bayesian Networks by Adnan Darwiche.

Del Castillo, E., and Hurwitz, A.M. (1997). Run-to-Run process control: Literature review and extensions. Journal of Quality Technology *29*, 184–196.

Del Castillo, E., and Yeh, J.-Y. (1998). An adaptive run-to-run optimizing controller for linear and nonlinear semiconductor processes. IEEE Transactions on Semiconductor Manufacturing *11*, 285–295.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. IEEE Transactions on Systems, Man, and Cybernetics *25*, 804–813.

Deventer, R., Denzler, J., and Niemann, H. Control of Dynamic Systems Using Bayesian Networks. 7.

Dunia, R., and Qin, S.J. (1998). Subspace approach to multidimensional fault identification and reconstruction. AIChE Journal *44*, 1813–1831.

Elidan, G., Ninio, M., Friedman, N., and Schuurmans, D. (2002). Data Perturbation for Escaping Local Maxima in Learning. In AAAI/IAAI, p.

Fan, S.-K.S., and Chang, Y.-J. (2013). An integrated advanced process control framework using run-to-run control, virtual metrology and fault detection. Journal of Process Control *23*, 933–942.

Ferreira, A., Roussy, A., Kernaflen, C., Gleispach, D., Hayderer, G., Gris, H., and Besnard, J. (2011). Virtual metrology models for predicting avera PECVD oxide film thickne. In 2011 IEEE/SEMI Advanced Semiconductor Manufacturing Conference, pp. 1–6.

Firth, S.K., Campbell, W.J., Toprac, A., and Edgar, T.F. (2006). Just-in-time adaptive disturbance estimation for run-to-run control of semiconductor processes. IEEE Transactions on Semiconductor Manufacturing *19*, 298–315.

Friedman, N., Murphy, K., and Russell, S. (1998). Learning the Structure of Dynamic Probabilistic Networks. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), pp. 139–147.

Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000). Using Bayesian Networks to Analyze Expression Data. Journal of Computational Biology *7*, 601–620.

Gao, S., Xiao, Q., Pan, Q., and Li, Q. (2007). Learning Dynamic Bayesian Networks Structure Based on Bayesian Optimization Algorithm. In Advances in Neural Networks – ISNN 2007, D. Liu, S. Fei, Z. Hou, H. Zhang, and C. Sun, eds. (Springer Berlin Heidelberg), pp. 424–431.

Geman, S., and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence *PAMI-6*, 721–741.

Goldfarb, D., & Idnani, A. (1982). Dual and primal-dual methods for solving strictly convex quadratic programs. In Numerical Analysis (pp. 226-239). Springer, Berlin, Heidelberg.

Good, R.P., and Qin, S.J. (2006). On the stability of MIMO EWMA run-to-run controllers with metrology delay. IEEE Transactions on Semiconductor Manufacturing *19*, 78–86.

Good, R.P., Kost, D., and Cherry, G.A. (2010). Introducing a Unified PCA Algorithm for Model Size Reduction. IEEE Transactions on Semiconductor Manufacturing *23*, 201–209.

Goodlin, B.E., Boning, D.S., Sawin, H.H., and Wise, B.M. (2003). Simultaneous Fault Detection and Classification for Semiconductor Manufacturing Tools. J. Electrochem. Soc. *150*, G778–G784.

Guo, H., and Hsu, W. (2002). A Survey of Algorithms for Real-Time Bayesian Network Inference. In In In the Joint AAAI-02/KDD-02/UAI-02 Workshop on Real-Time Decision Support and Diagnosis Systems, p.

Hankinson, M., Vincent, T., Irani, K.B., and Khargonekar, P.P. (1997). Integrated real-time and run-to-run control of etch depth in reactive ion etching. IEEE Transactions on Semiconductor Manufacturing *10*, 121–130.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (New York: Springer-Verlag).

He, Q.P., and Wang, J. (2007). Fault Detection Using the k-Nearest Neighbor Rule for Semiconductor Manufacturing Processes. IEEE Transactions on Semiconductor Manufacturing *20*, 345–354.

Heckerman, D., and Geiger, D. (1995). Learning Bayesian Networks: A Unification for Discrete and Gaussian Domains. In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence, (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), pp. 274–284.

Henrion, M. (1988). Propagating Uncertainty in Bayesian Networks by Probabilistic Logic Sampling. In Machine Intelligence and Pattern Recognition, J.F. Lemmer, and L.N. Kanal, eds. (North-Holland), pp. 149–163.

Heskes, T., and Zoeter, O. (2002). Expectation propagation for approximate inference m dynamic Bayesian networks. In UAI 2002, p.

Hong, S.J., Lim, W.Y., Cheong, T., and May, G.S. (2012). Fault Detection and Classification in Plasma Etch Equipment for Semiconductor Manufacturing$e$-Diagnostics. IEEE Transactions on Semiconductor Manufacturing *25*, 83–93.

Hotelling, H. (1947). Multivariate Quality Control-illustrated by the air testing of sample bombsights. p.

Huang, Y., Cheng, F., and Hung, M. (2009). Developing a product quality fault detection scheme. In 2009 IEEE International Conference on Robotics and Automation, pp. 927–932.

Hung, M.H., Lin, T.H., Cheng, F.T., and Lin, R.C. (2007). A Novel Virtual Metrology Scheme for Predicting CVD Thickness in Semiconductor Manufacturing. IEEE/ASME Transactions on Mechatronics *12*, 308–316.

Jazwinski, A.H., Stochastic Processes and Filrering Theory. New York: Academic, 1970.

Jebri, M.A., Adel, E.M.E., Graton, G., Ouladsine, M., and Pinaton, J. (2017). Virtual Metrology applied in Run-to-Run Control for a Chemical Mechanical Planarization process. J. Phys.: Conf. Ser. *783*, 012042.

Kang, S., and Kang, P. (2017a). An intelligent virtual metrology system with adaptive update for semiconductor manufacturing. Journal of Process Control *52*, 66–74.

Kang, S., and Kang, P. (2017b). An intelligent virtual metrology system with adaptive update for semiconductor manufacturing. Journal of Process Control *52*, 66–74.

Kang, P., Lee, H., Cho, S., Kim, D., Park, J., Park, C.-K., and Doh, S. (2009). A virtual metrology system for semiconductor manufacturing. Expert Systems with Applications *36*, 12554–12561.

Kang, P., Kim, D., Lee, H., Doh, S., and Cho, S. (2011). Virtual metrology for run-to-run control in semiconductor manufacturing. Expert Systems with Applications *38*, 2508–2522.

Kao, C.A., Cheng, F.T., Wu, W.M., Kong, F.W., and Huang, H.H. (2013). Run-to-Run Control Utilizing Virtual Metrology With Reliance Index. IEEE Transactions on Semiconductor Manufacturing *26*, 69–81.

Kass, R.E., Carlin, B.P., Gelman, A., and Neal, R.M. (1998). Markov Chain Monte Carlo in Practice: A Roundtable Discussion. The American Statistician *52*, 93–100.

Khan, A.A., Moyne, J.R., and Tilbury, D.M. (2007). An Approach for Factory-Wide Control Utilizing Virtual Metrology. IEEE Transactions on Semiconductor Manufacturing *20*, 364–375.

Khan, A.A., Moyne, J.R., and Tilbury, D.M. (2008). Virtual metrology and feedback control for semiconductor manufacturing processes using recursive partial least squares. Journal of Process Control *18*, 961–974.

Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models. Journal of Computational and Graphical Statistics *5*, 1–25.

Knuth, D.E. (1997). The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms (Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc.).

Kortenkamp, D., and Weymouth, T. (1994). Topological Mapping for Mobile Robots Using a Combination of Sonar and Vision Sensing. In Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2), (Menlo Park, CA, USA: American Association for Artificial Intelligence), pp. 979–984.

Kourti, T., and MacGregor, J.F. (1996). Multivariate SPC Methods for Process and Product Monitoring. Journal of Quality Technology *28*, 409–428.

Larranaga, P., Poza, M., Yurramendi, Y., Murga, R.H., and Kuijpers, C.M.H. (1996). Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. IEEE Transactions on Pattern Analysis and Machine Intelligence 18, 912–926.

Lauritzen, S.L. (1992). Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models. Journal of the American Statistical Association *87*, 1098.

Lauritzen, S.L., and Spiegelhalter, D.J. (1988). Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. Journal of the Royal Statistical Society. Series B (Methodological) *50*, 157–224.

Lauritzen, S.L., and Wermuth, N. (1989). Graphical Models for Associations between Variables, some of which are Qualitative and some Quantitative. Ann. Statist. *17*, 31–57.

Lerner, U., Parr, R., Koller, D., and Biswas, G. (2000). Bayesian Fault Detection and Diagnosis in Dynamic Systems. In AAAI/IAAI, p.

Li, C., and Mahadevan, S. (2018). Efficient approximate inference in Bayesian networks with continuous variables. Rel. Eng. & Sys. Safety *169*, 269–280.

Lynn, S., Ringwood, J., and MacGearailt, N. (2010). Gaussian process regression for virtual metrology of plasma etch. In IET Irish Signals and Systems Conference (ISSC 2010), pp. 42–47.

MacGregor, J.F., Jaeckle, C., Kiparissides, C., and Koutoudi, M. (1994). Process monitoring and diagnosis by multiblock PLS methods. AIChE Journal *40*, 826–838.

Madsen, A.L., and Jensen, F.V. (1999). Lazy propagation: A junction tree inference algorithm based on lazy evaluation. Artificial Intelligence *113*, 203–245.

Margaritis, D. (2003). Learning Bayesian Network Model Structure From Data.

Mason, R.L., Tracy, N.D., and Young, J.C. (1995). Decomposition of T2 for Multivariate Control Chart Interpretation. Journal of Quality Technology *27*, 99–108.

Mastrangelo, C.M., Runger, G.C., and Montgomery, D.C. (1996). Statistical process monitoring with principal components. Quality and Reliability Engineering International *12*, 203–210.

May, G.S., and Spanos, C.J. (2006). Fundamentals of semiconductor manufacturing and process control (Newark, NJ: Wiley).

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. J. Chem. Phys. *21*, 1087–1092.

Miller, P., Swanson, R.E., and Heckler, C.E. (1998). Contribution plots: a missing link in multivariate quality control. Applied Mathematics and Computer Science *Vol. 8*, 775–792.

Minka, T. (1999). From hidden markov models to linear dynamical systems (pp. 1-10). Technical report, MIT

Moyne, J., Del Castillo, E., and Hurwitz, A.M. (2000). Run-to-run control in semiconductor manufacturing. Boca Raton: CRC Press.

Moyne, J., and Iskandar, J. (2017). Big Data Analytics for Smart Manufacturing: Case Studies in Semiconductor Manufacturing. Processes *5*, 39.

Moyne, J., Iskandar, J., and Armacost, M. (2015). "Next-Generation Fault Detection for Improved Quality and Reduced Cost," NanoChip Magazine, Vol. 10, No. 2.

Murphy, K.P. (2002). Dynamic Bayesian Networks: Representation, Inference and Learning. PhD Thesis. University of California, Berkeley.

Murphy, K., and Mian, S. (1999). Modelling gene expression data using dynamic bayesian networks.

Ning, Z., Moyne, J.R., Smith, T., Boning, D., Del Castillo, E., Yeh, J.-Y., and Hurwitz, A. (1996). A comparative analysis of run-to-run control algorithms in the semiconductor manufacturing industry. In IEEE/SEMI 1996 Advanced Semiconductor Manufacturing Conference and Workshop. Theme-Innovative Approaches to Growth in the Semiconductor Industry. ASMC 96 Proceedings, pp. 375–381.

Olson, K., and Moyne, J. (2010). Adaptive Virtual Metrology applied to a CVD process. In 2010 IEEE/SEMI Advanced Semiconductor Manufacturing Conference (ASMC), pp. 353–358.

Palmer, E., and Spanos, and C.J. (1996). Control of photoresist properties: a Kalman filter based approach. IEEE Transactions on Semiconductor Manufacturing *9*, 208–214.

Pampuri, S., Schirru, A., Fazio, G., and Nicolao, G.D. (2011). Multilevel Lasso applied to Virtual Metrology in semiconductor manufacturing. In 2011 IEEE International Conference on Automation Science and Engineering, pp. 244–249.

Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.).

Perrin, B.-E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d'Alché–Buc, F. (2003). Gene networks inference using dynamic Bayesian networks. Bioinformatics *19*, ii138–ii148.

Potra, F.A., and Wright, S.J. (2000). Interior-point methods. Journal of Computational and Applied Mathematics 124, 281–302.

Qin, S.J. (2003). Statistical process monitoring: basics and beyond. Journal of Chemometrics *17*, 480–502.

Qin, S.J. (2012). Survey on data-driven industrial process monitoring and diagnosis. Annual Reviews in Control *36*, 220–234.

Raich, A., and Çinar, A. (1996). Statistical process monitoring and disturbance diagnosis in multivariable continuous processes. AIChE Journal *42*, 995–1009.

Raich, A., and Çinar, A. (1997). Diagnosis of process disturbances by statistical distance and angle measures. Computers & Chemical Engineering *21*, 661–673.

Raissi, M. (2018). Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations. J. Mach. Learn. Res. *19*, 932–955.

Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2017). Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. ArXiv:1711.10561 [Cs, Math, Stat].

Rajapakse, J.C., and Zhou, J. (2007). Learning effective brain connectivity with dynamic Bayesian networks. Neuroimage *37*, 749–760.

Rasmussen, C.E. (2003). Gaussian Processes in Machine Learning. In Advanced Lectures on Machine Learning, (Springer, Berlin, Heidelberg), pp. 63–71.

Rendall, R., and Reis, M.S. (2018). Which regression method to use? Making informed decisions in "data-rich/knowledge poor" scenarios – The Predictive Analytics

Comparison framework (PAC). Chemometrics and Intelligent Laboratory Systems *181*, 52–63.

Rendall, R., Lu, B., Castillo, I., Chin, S.-T., Chiang, L.H., and Reis, M.S. (2017). A Unifying and Integrated Framework for Feature Oriented Analysis of Batch Processes. Ind. Eng. Chem. Res. 56, 8590–8605.

Rendall, R., Pereira, A.C., and Reis, M.S. (2017). Advanced predictive methods for wine age prediction: Part I – A comparison study of single-block regression approaches based on variable selection, penalized regression, latent variables and tree-based ensemble methods. Talanta *171*, 341–350.

Roy, A., Govil, S., and Miranda, R. (1995). An algorithm to generate radial basis function (RBF)-like nets for classification problems. Neural Networks *8*, 179–201.

Sachs, E., Hu, A., and Ingolfsson, A. (1995). Run by run process control: combining SPC and feedback control. IEEE Transactions on Semiconductor Manufacturing *8*, 26–43.

Salmerón, A., Rumí, R., Langseth, H., Nielsen, T.D., and Madsen, A.L. (2018). A Review of Inference Algorithms for Hybrid Bayesian Networks. Journal of Artificial Intelligence Research *62*, 799–828.

Scutari, M. (2009). Learning Bayesian Networks with the bnlearn R Package. ArXiv:0908.3817 [Stat].

Scutari, M. (2014). Bayesian Network Constraint-Based Structure Learning Algorithms: Parallel and Optimised Implementations in the bnlearn R Package. ArXiv:1406.7648 [Cs, Stat].

Shachter, R.D. (1986). Intelligent Probabilistic Inference. In Machine Intelligence and Pattern Recognition, L.N. Kanal, and J.F. Lemmer, eds. (North-Holland), pp. 371–382.

Silverman, B.W. (1986). Density Estimation for Statistics and Data Analysis, Chapman & Hall/CRC. London.

Smith, T.H., and Boning, D.S. (1997). A self-tuning EWMA controller utilizing artificial neural network function approximation techniques. IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part C *20*, 121–132.

Smith, T.H., Boning, D.S., Stefani, J., and Butler, S.W. (1998). Run by run advanced process control of metal sputter deposition. IEEE Transactions on Semiconductor Manufacturing *11*, 276–284.

Smola, A.J., and Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and Computing *14*, 199–222.

Spanos, C.J. (1992). Statistical process control in semiconductor manufacturing. Proceedings of the IEEE *80*, 819–830.

Spirtes, P., Glymour, C. N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., & Richardson, T. (2000). Causation, prediction, and search. MIT press.

Stoddard, K., Crouch, P., Kozicki, M., and Tsakalis, K. (1994). Application of feedforward and adaptive feedback control to semiconductor device manufacturing. In Proceedings of 1994 American Control Conference - ACC '94, pp. 892–896 vol.1.

Strobl, C., Malley, J., and Tutz, G. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. Psychol Methods *14*, 323–348.

Susto, G.A., and Beghi, A. (2013). A virtual metrology system based on least angle regression and statistical clustering. Appl. Stochastic Models Bus. Ind. *29*, 362–376.

Susto, G.A., Beghi, A., and Luca, C.D. (2011). A Virtual Metrology system for predicting CVD thickness with equipment variables and qualitative clustering. In ETFA2011, pp. 1–4.

Susto, G.A., Pampuri, S., Schirru, A., Nicolao, G.D., McLoone, S., and Beghi, A. (2012). Automatic Control and Machine Learning for Semiconductor Manufacturing: Review and Challenges. 8.

Tan, F., Pan, T., Li, Z., and Chen, S. (2015). Survey on Run-to-Run Control Algorithms in High-Mix Semiconductor Manufacturing Processes. IEEE Transactions on Industrial Informatics *11*, 1435–1444.

Tartakovsky, A.M., Marrero, C.O., Perdikaris, P., Tartakovsky, G.D., and Barajas-Solano, D. (2018). Learning Parameters and Constitutive Relationships with Physics Informed Deep Neural Networks. ArXiv:1808.03398 [Physics].

Teyssier, M., and Koller, D. (2005). Ordering-based search: a simple and effective algorithm for learning Bayesian networks. In Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence (UAI-05), Bacchus, F. & Jaakkola, T. (eds). AUAI Press, 584–590.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. Journal of the Royal Statistical Society. Series B (Methodological) *58*, 267–288.

Tobon-Mejia, D.A., Medjaher, K., and Zerhouni, N. (2012). CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. Mechanical Systems and Signal Processing *28*, 167–182.

Trabelsi, G., Leray, P., Ben Ayed, M., and Alimi, A.M. (2013). Dynamic MMHC: A Local Search Algorithm for Dynamic Bayesian Network Structure Learning. In Advances in Intelligent Data Analysis XII, A. Tucker, F. Höppner, A. Siebes, and S. Swift, eds. (Springer Berlin Heidelberg), pp. 392–403.

Tsamardinos, I., Brown, L.E., and Aliferis, C.F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning 65, 31–78.

Tseng, S.-T., Chou, R.-J., and Lee, S.-P. (2002). A study on a multivariate EWMA controller. IIE Transactions *34*, 541–549.

Verma, T., and Pearl, J. (1992). An Algorithm for Deciding if a Set of Observed Independencies Has a Causal Explanation. In Uncertainty in Artificial Intelligence, D. Dubois, M.P. Wellman, B. D'Ambrosio, and P. Smets, eds. (Morgan Kaufmann), pp. 323–330.

Wan, J., Pampuri, S., O'Hara, P.G., Johnston, A.B., and McLoone, S. (2014). On regression methods for virtual metrology in semiconductor manufacturing. In 25th IET Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014), pp. 380–385.

Wang, J., and He, Q.P. (2007). A Bayesian Approach for Disturbance Detection and Classification and Its Application to State Estimation in Run-to-Run Control. IEEE Transactions on Semiconductor Manufacturing *20*, 126–136.

Wang, J., Peter He, Q., and Edgar, T.F. (2009). State estimation in high-mix semiconductor manufacturing. Journal of Process Control *19*, 443–456.

Wang, Y., Zheng, Y., Gu, X. g, and Huang, L. (2015). Multi-objective Fault Monitoring for Semiconductor Manufacturing Process with DEWMA Run-to-Run Controller. In Industrial Information Integration 2015 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, pp. 152–155.

Wise, B.M., and Gallagher, N.B. (1996). The process chemometrics approach to process monitoring and fault detection. Journal of Process Control *6*, 329–348.

Wold, S., Ruhe, A., Wold, H., and Dunn, I., W. (1984). The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. SIAM J. Sci. and Stat. Comput. *5*, 735–743.

Wu, M., Lin, C., Wong, D.S., Jang, S., and Tseng, S. (2008). Performance Analysis of EWMA Controllers Subject to Metrology Delay. IEEE Transactions on Semiconductor Manufacturing *21*, 413–425.

Yang, L., and Lee, J. (2012). Bayesian Belief Network-based approach for diagnostics and prognostics of semiconductor manufacturing systems. Robotics and Computer-Integrated Manufacturing *28*, 66–74.

Yang, W.-T., Blue, J., Roussy, A., Pinaton, J., and Reis, M.S. (2019). A Structure Data-Driven Framework for Virtual Metrology Modeling. IEEE Transactions on Automation Science and Engineering 1–10.

Yang, X., Tartakovsky, G., and Tartakovsky, A. (2018). Physics-Informed Kriging: A Physics-Informed Gaussian Process Regression Method for Data-Model Convergence. ArXiv:1809.03461 [Cs, Stat].

Yue, H.H., and Tomoyasu, M. (2004). Weighted principal component analysis and its applications to improve FDC performance. In 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), pp. 4262-4267 Vol.4.

Zeng, D., and Spanos, C.J. (2009). Virtual Metrology Modeling for Plasma Etch Operations. IEEE Transactions on Semiconductor Manufacturing *22*, 419–431.

Zhang, N. L., & Poole, D. (1994). A simple approach to Bayesian network computations. In Proceedings of the Biennial Conference-Canadian Society for Computational Studies of Intelligence (pp. 171-178).

Zou, M., and Conzen, S.D. (2005). A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data. Bioinformatics *21*, 71–79.

Zweig, G., and Russell, S.J. (1998). Speech recognition with dynamic Bayesian networks. In Proceedings of the 15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference, Madison, WI, pp. 173–180.

# Appendix A. Process Flows of Controllers

In the case study of this thesis, the efficiencies of four controllers are examined: the dEWMA controller (Butter and Stefani, 1994), the PLS-RC (Khan et al. 2008), the SRC (Section 4.7), and the A-SRC (Section 4.8). The process flows of these controllers are presented below.

- *dEWMA controller*

The procedure for dEWMA is presented in Algorithm A.1, where $W$ is the number of wafers of the testing dataset.

**Algorithm A.1** dEWMA

**for** $w$ in 2: $W$ :
    **if** $w-1 \in$ metrology run
        $u_w = f_{dEWMA}(y_{w-1}, \lambda)$
    **else**
        $u_w = u_{w-1}$
    **end if**
**end for**

- *PLS-RC*

The procedure of PLS-RC basically follows that in Khan's study (see Algorithm A.2). A PLS model will be first constructed based on $\mathcal{D}_1$. For online control value computation, the dEWMA filter is employed. The PLS model will be updated when the new actual metrology data is available.

**Algorithm A.2** PLS-RC

***(I) PLS Modeling***

**function** $VM(\mathcal{D})$
    $\mathcal{P}$= learn_pls($\mathcal{D}$)
    **Return** $\mathcal{P}$
**end function**

***(II) PLS-RC (Main)***

$\mathcal{D}_{current} = \mathcal{D}_1$
$\mathcal{P}_{current} = VM(\mathcal{D}_{current})$
**for** $w$ in 2: $W$
    **if** $w-1 \in$ metrology run
        $u_w = f_{R2R}(y_{w-1}, \lambda)$
        $\mathcal{D}_{current} = \mathcal{D}_{current}[2:,] + d_w$
        $\mathcal{P}_{current}$= VM ($\mathcal{D}_{current}$)
    **else**
        $\hat{y}_{w-1} = \mathcal{P}_{current}(x_{w-1})$
        $u_w = f_{dEWMA}(\hat{y}_{w-1}, \lambda)$
    **end if**
**end for**

- *SRC and A-SRC*

Following the framework in Fig. 4.1, the procedure of SRC and A-SRC is presented in Algorithm A.3. A DBN $\mathcal{G}_{current}$ based on training set $\mathcal{D}_1$ and a blacklist $\mathcal{L}$ will be learned at the offline stage. The SRC, denoted as $\mathcal{S}_{current}$, is extracted from $\mathcal{G}_{current}$. For each run in the testing set $\mathcal{D}_2$, the data of the previous wafer $\mathcal{d}_w$ will be used for likelihood monitoring of each variable $V_k$ (see Section 4.5.2). For A-SRC, if the risk level of $h_k$ exceeds $h_{risk}$, the local update of $V_k$ will be triggered. The wafers in the alarm window will be used to compute the local distribution of $V_k$, and the updated DBN $\mathcal{G}'$ is obtained. By setting the current DBN $\mathcal{G}_{current} = \mathcal{G}'$ and extract $\mathcal{S}_{current}$, the next step will follow the simple SRC, give equipment prediction, and obtain the updated $\mathcal{d}'_w$. The control value will be calculated based on $\mathcal{S}_{current}$ and $\mathcal{d}'_w$.

**Algorithm A.3** SRC and A-SRC

***(I) DBN***

**function** $DBN(\mathcal{D}, \mathcal{L})$
    $\mathcal{G}$= learn_dbn($\mathcal{D}, \mathcal{L}$)
    **return** $\mathcal{G}$
**end function**
**Function** UPDATE($\mathcal{D}, \mathcal{G}, V$)
    $\theta'_V = \text{get\_new\_distribution}(\mathcal{D}, V)$
    $\mathcal{G}' = \text{local\_update}(\mathcal{G}, \theta'_V)$
    **return** $\mathcal{G}'$
**end function**

***(II) Likelihood Monitoring***

**function** likelihood_monitoring($\mathcal{d}_w, \mathcal{G}, V$)
    $LLI$= get_likelihood($\mathcal{d}_w, \mathcal{G}, V$)
    $\mathcal{f} = \text{check\_if\_update}(LLI)$
    **return** $\mathcal{f}$
**end function**

***(III) A-SRC (Main)***

$\mathcal{D}_{current} = \mathcal{D}_{1,DBN}$
$\mathcal{G}_{current} = DBN(\mathcal{D}_{current}, \mathcal{L})$
$\mathcal{S}_{current} = \text{extract\_SRC}(\mathcal{G}_{current})$
**for** $t$ in 2: $T$
    **for each** $V_k \in \mathbb{V}$
        $\mathcal{f}_k$= likelihood_monitoring($\mathcal{d}_{w-1}, \mathcal{G}, V_k$)
        **if** $\mathcal{f}_k$ *is* $TRUE$
            $\mathcal{G}' = \text{UPDATE}(\mathcal{D}_{alarm\_window}, \mathcal{G}_{current}, V_k)$
            $\mathcal{G}_{current} = \mathcal{G}'$
            $\mathcal{S}_{current} = \text{extract\_SRC}(\mathcal{G}_{current})$
        **end if**
    **end for**
    $\mathcal{d}'_w = \text{equipmemt\_condition\_prediction}(\mathcal{d}_w, \mathcal{S}_{current})$
    $u_w = \text{control\_value\_computation}(\mathcal{d}'_w, \mathcal{S}_{current})$
**end for**

# Appendix B. Case Study 2 – Likelihood Monitoring

In Section 5.1, the Advanced-SRC (A-SRC) which incorporates a likelihood monitoring mechanism is employed in the simulated process case study. For each run in the testing set $\mathcal{D}_2$, the data will be used for likelihood monitoring of each variable $V_k$. If the risk level of $h_k$ exceeds $h_{risk}$, the local update of $V_k$ will be triggered. The results of likelihood monitoring of $\mathcal{D}_2$ of three datasets are presented in Fig. B.1, Fig. B.2, and Fig. B.3. The green lines indicate that the local update is activated. The red lines are the control limits determined based on kernel density estimation of training set $\mathcal{D}_1$, where the Gaussian is employed as kernel function, the Type I error is set to be 0.1. The bandwidth is determined by the rule-of-thumb, which is 0.9 times the minimum of the standard deviation and the interquartile range divided by 1.34 times the sample size to the negative one-fifth power (Silverman, 1986).

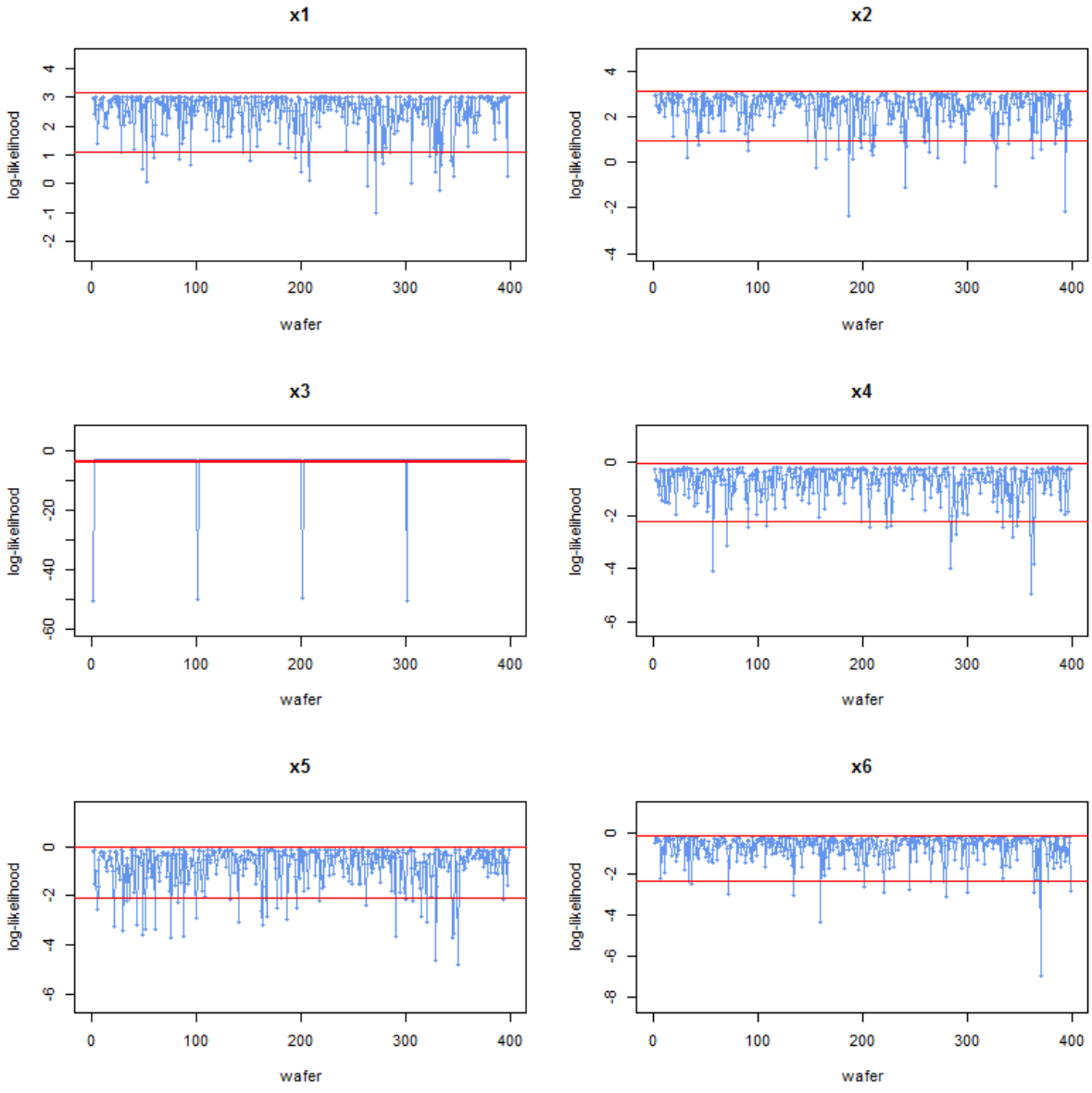


Figure B.1 The likelihood monitoring of each variable of the testing sets under the shift disturbance.

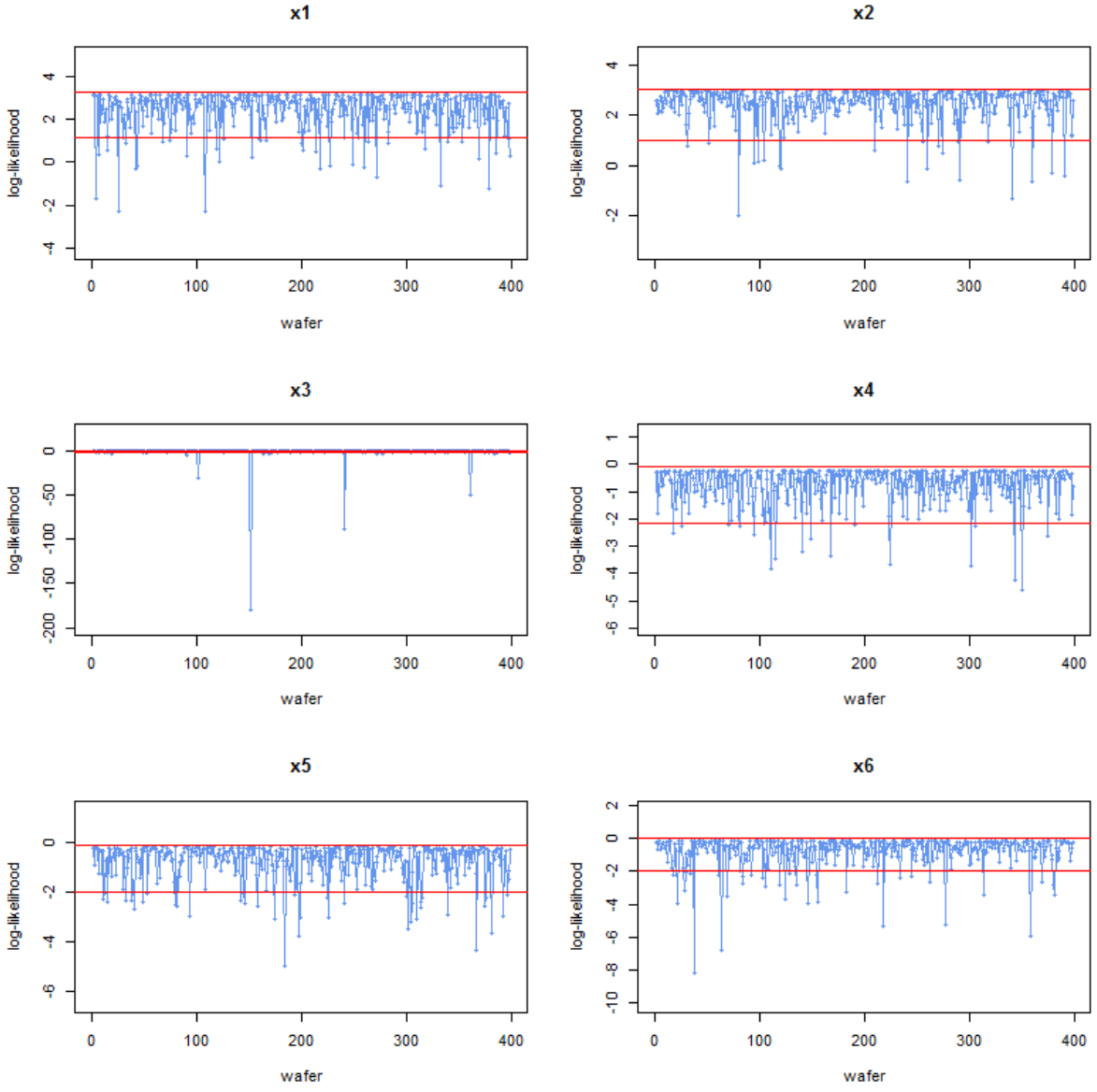


Figure B.2 The likelihood monitoring of each variable of the testing sets under the impulse disturbance.

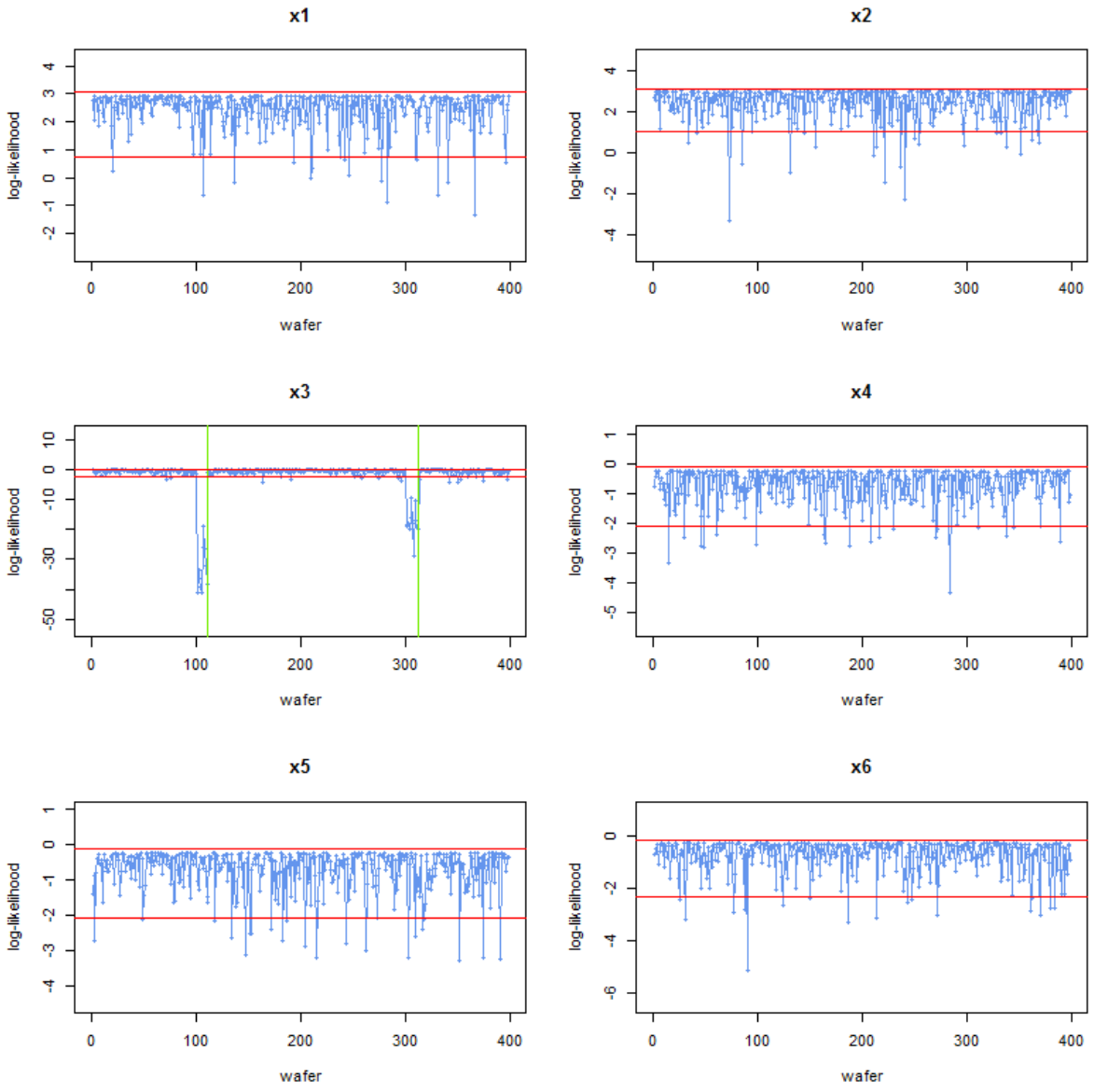


Figure B.3 The likelihood monitoring of each variable of the testing sets under the shift disturbance.

# Appendix C. Abbreviations

| Abbreviations | Full Name |
|---|---|
| APC | Advanced Process Control |
| A-SRC | Advanced Structured R2R Controller |
| BIC | Bayesian Information Criterion |
| BN | Bayesian Network |
| CD | Critical Dimension |
| CLGBN | Conditional Linear Gaussian Bayesian Network |
| CMP | Chemical-Mechanical Polishing |
| CPD | Conditional Probability Distribution |
| CVD | Chemical Vapor Deposition |
| DAG | Directed Acyclic Graph |
| DBN | Dynamic Bayesian Network |
| dEWMA | Double Exponentially Weighted Moving Average |
| DoE | Design of Experiments |
| EHM | Equipment Healthy Management |
| EWMA | Exponentially Weighted Moving Average |
| FDA | Fisher Discriminant Analysis |
| FDC | Fault Detection and Classification |
| GBN | Gaussian Bayesian Network |
| GLI | Global Likelihood Index |
| GM | Gradual Mode |
| GMt | Time-based Gradual Mode |
| GPR | Gaussian Process Regression |
| GSI | Global Similarity Index |
| HC | Hill Climbing |
| IC | Integrated Circuits |
| IC | Inductive Causation |
| IM | Integrated Metrology |
| IS | Importance Sampling |
| ISI | Individual Similarity Indexes |
| KFM | Kalman Filter Model |
| KIRC | Knowledge-based Interactive Run-to-run Control |
| KNN | K-Nearest Neighbors |
| LASSO | Least Absolute Shrinkage and Selection Operator |
| LLI | Local Likelihood Index |
| MCMC | Markov Chain Monte Carlo |
| MES | Manufacturing Execution System |
| MIMO | Multiple-Input–Multiple-Output |
| MLR | Multiple Linear Regression |
| MMHC | Max-Min Hill Climbing algorithm |
| MNN | Modular Neural Network |
| MSE | Mean Square Error |

| | |
|---|---|
| MYT | The Mason, Young and Tracy decomposition |
| NN | Neural Networks |
| OAQC | Optimizing Adaptive Quality Controller |
| OEE | Overall Equipment Effectiveness |
| OOC | Out-Of-Control |
| PCA | Principal Component Analysis |
| PCC | Predictor Corrector Control |
| PCR | Principal Component Regression |
| PDE | Partial Differential Equations |
| PdM | Predictive Maintenance |
| PECVD | Plasma Enhanced Chemical Vapor Deposition |
| PLS | Multiple Linear Regression |
| PLS-RC | the dEWMA controller incorporating a PLS model |
| PM | Preventive Maintenance |
| R2R | Run-to-Run |
| RF | Random Forest |
| RI | Reliance Index |
| RM | Rapid Mode |
| RMSE | Root Mean Square Error |
| RS | Relative Score |
| RSM | Response Surface Model |
| RT | Regression Trees |
| SISO | Single-Input-Single-Output |
| SM | Smart Manufacturing |
| SME | Subject Matter Experts |
| SPC | Statistical Process Control |
| SPE | Squared Prediction Erorr |
| SRC | Structured R2R Control |
| SVID | Status Variable IDentification |
| SVR | Support Vector Regression |
| VE | Variable Elimination |
| VM | Virtual Metrology |

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2020LYSEM004

Wei-Ting YANG

# An Integrated Physics-Informed Process Control Framework and Its Applications to Semiconductor Manufacturing

Speciality: Microelectronics

Keywords: Dynamic Bayesian Network (DBN), Physics-informed, Run-to-Run (R2R) control, Virtual Metrology (VM), Semiconductor Manufacturing.

## Abstract:

The primary task in semiconductor manufacturing is to produce chips with high quality while keeping short production cycles. To reduce the process variation, many process monitoring and control solutions have been implemented in the fabrication. To improve the product quality and reduce process variability, these solutions should seamlessly coordinate to not only share the information but also make consistent and optimal decisions. In this thesis, an *Integrated Physics-Informed Process Control Framework* is proposed, which is capable of coordinating various data sources and supporting multiple applications. Dynamic Bayesian Network (DBN) is employed as the modeling foundation of the framework, which can consolidate available data and existing physics information. As DBN is presented as connected graphs, the interpretation of the associations between process variables is intuitive and understandable. Three DBN-based applications are proposed in this thesis. A monitoring mechanism is employed for fault detection and diagnosis. A Virtual Metrology (VM) model is developed to provide efficient prediction and its underlying causalities. Finally, a Structured Run-to-Run Controller (SRC) is implemented, which aims to optimize the control decision by considering not only the product metrology but also the interactions between process parameters. The proposed approaches are evaluated through a practical case study and simulated data.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT : 2020LYSEM004

Wei-Ting YANG

# Etude d'un cadre de contrôle intégré incorporant les connaissances du domaine et son application à la fabrication des semi-conducteurs

Spécialité: Microélectronique

Mots clefs : Réseau Bayésien Dynamique (RDB), Régulateur Run-to-Run, Modèle de Métrologie Virtuelle (VM), Fabrication de Semi-conducteurs

## Résumé :

La principale tâche de la fabrication de semi-conducteurs est de produire des puces de haute qualité tout en maintenant des cycles de production courts. Afin de réduire les variations de processus, de nombreuses solutions de surveillance et de contrôle de processus ont été mises en œuvre en fabrication. Dans le même objectif d'améliorer la qualité des processus, ces solutions devraient finalement être coordonnées, non seulement par le partage d'informations, mais également par la prise de décisions optimisées et cohérentes. Cette thèse propose un cadre de contrôle intégré incorporant des informations physiques, et capables de rassembler diverses sources et de prendre en charge plusieurs applications. Un Réseau Bayésien Dynamique (souvent noté RBD, ou DBN pour Dynamic Bayesian Network) est utilisé comme base de modélisation du cadre, permettant de consolider les données disponibles et les connaissances des experts. Dans la mesure où un DBN est un graphe connecté, l'interprétation des associations entre les variables de processus est simple et intuitive. En se basant sur le DBN, trois applications sont proposées dans cette thèse. Un mécanisme de surveillance est utilisé pour la détection et le diagnostic des défauts. Un modèle de métrologie virtuelle (VM) est développé pour fournir une prédiction efficace et ses causalités sous-jacentes. Enfin, un régulateur structuré Run-to-Run (SRC) est mis en œuvre en prenant en compte les résultats de la métrologie et les interactions entre les paramètres de processus. Les approches proposées sont évaluées à travers une étude de cas réel et des données simulées.