Real-time Analysis and Control for Smart Manufacturing Systems

by

Yunyi Kang

A Dissertation Presented in Partial Fulfillment
of the Requirement for the Degree
Doctor of Philosophy

Approved June 2020 by the
Graduate Supervisory Committee:

Feng Ju, Chair
Giulia Pedrielli
Hao Yan
Teresa Wu

ARIZONA STATE UNIVERSITY

August 2020

ABSTRACT

Recent advances in manufacturing system, such as advanced embedded sensing, big data analytics and IoT and robotics, are promising a paradigm shift in the manufacturing industry towards smart manufacturing systems. Typically, real-time data is available in many industries, such as automotive, semiconductor, and food production, which can reflect the machine conditions and production systems operation performance. However, a major research gap still exists in terms of how to utilize these real-time data information to evaluate and predict production system performance and to further facilitate timely decision making and production control on the factory floor. To tackle these challenges, this dissertation takes on an integrated analytical approach by hybridizing data analytics, stochastic modeling and decision making under uncertainty methodology to solve practical manufacturing problems.

Specifically, in this research, the machine degradation process is considered. It has been shown that machines working at different operating states may break down in different probabilistic manners. In addition, machines working in worse operating stage are more likely to fail, thus causing more frequent down period and reducing the system throughput. However, there is still a lack of analytical methods to quantify the potential impact of machine condition degradation on the overall system performance to facilitate operation decision making on the factory floor. To address these issues, this dissertation considers a serial production line with finite buffers and multiple machines following Markovian degradation process. An integrated model based on the aggregation method is built to quantify the overall system performance and its interactions with machine condition process. Moreover, system properties are investigated to analyze the influence of system parameters on system performance. In addition, three types of bottlenecks are defined and their corresponding indicators are derived to provide guidelines on improving system performance. These meth-

ods provide quantitative tools for modeling, analyzing, and improving manufacturing systems with the coupling between machine condition degradation and productivity given the real-time signals.

To my family

ACKNOWLEDGEMENTS

I would like to show my deepest appreciation to my advisor, Dr. Feng Ju, for his five-year selfless support, both financially and spiritually, towards my completion of the PhD study. I can not survive this journey without his comprehensive guidance on research and constructive advise on life and personality. I can always recall the scene when he tells me to set up proactive and positive attitude, to prepare for the challenges and fight back, and to accept the outcomes mutually with no complaints. This will become the forever treasure in my life.

I also would like to thank my dissertation committee members, Dr. Teresa Wu, Dr. Giulia Pedrielli, and Dr. Hao Yan. Their fruitful comments and constructive suggestions significantly enrich the idea and rendition of this dissertation work. I am also in great honor to collaborate with them, either in courses or in research works, which widen my eyes and broaden my horizon on multiple research fields.

Many thanks go to the my peer collaborators, Logan Mathesen, Xinyu Zhao, Menghan Liu, and Viren Bhanushali. Your commitment on high quality research on our joint research projects contributes to the great research outcomes and publications.

I am also grateful to the peers in the research lab- Feifan Wang and Sepehr Fathizadan. Sepehr Fathizadan is the only office mate that I have during my PhD study. Feifan is impressive not only on his research but also on his excellent cooking.

I want to thank the other peers, Chao Wang, Houpu Yao, Dr. Zhe Gao, Dr. Guanqi Fang, Xufeng Yao, Penghui Zhang, and Jiajing Huang, and all the other professors and friends, for the help in courses and life, and for the witness on my growth.

Most of all, I am grateful to all my family members. I feel very lucky to receive unswerving support at any time for the past thirty years.

TABLE OF CONTENTS

CHAPTER Page

CHAPTER Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Motivation

During the past several centuries, three major industrial revolutions have been observed, moving the manufacturing systems from equipment-level empowerment, via mass production, to system-level automation. Currently, in the awareness of the fourth generation of industrial revolution, emerging concepts and technologies are involved, such as Internet of Things (IoT), cloud computing, and cyber physical systems, which makes the manufacturing systems towards intelligence, and bring both challenges and opportunities.

When mentioning industrial 4.0, it is commonly referred as smart manufacturing system. Such types of system create a connected and data-enriched environment. The manufacturing system therefore does not only produce parts, assemblies and final products. At the same time, it creates huge amount of real-time information. Moreover, the flow of the information is not bounded by the physical structure of the production system. The upper layers of the systems can collect and store the information in a centralized platform, through which all the component for the manufacturing processes can be monitored, analytics can be performed to evaluate the system performance and actions can be taken to solve the problems.

Furthermore, the smartness of the manufacturing system under the concept of Industrial 4.0 can be observed in three folds. First, the sensor-enabled equipment can achieve smart information processing. On one hand, real-time information can be collected with regards to the conditions and performance of the machines. Such

information, with the help of data analytics tools, can be used the evaluate the health conditions of the equipment. On the other hand, actions can be sent out to the equipment so that it can automatically execute the instructions of the operators, either manually or automatically. The second level of smartness the system performance assessment. Since all the equipment are connected and can communicate with each other, the overall system performance can be evaluated and projected, based on the information collected from each process and unit of the system. The third level is the smart decision. The system under this level knows not only how the system is performed, but also enables suggesting actions that can further project and improve the performance.

To achieve the highest level of the smartness, many efforts should be made. Specifically, system level performance analytics approaches are needed, which can make full use of the real-time information of the system for better system performance evaluations. Furthermore, based on the system conditions, advanced models are required to make the best decisions on system performance improvements. This motivates my research.

Specifically, in this research, machine condition degradation, a gradual and accumulative process on physical conditions that can be observed in numeric manufacturing assets and impact the manufacturing system operation and performance significantly, is utilized in the system modeling. It is well known that a machine working at different operating states may break down in different probabilistic manners. It has also been investigated that machines working in worse operating stage are more likely to fail, thus causing more frequent down period and reducing the system throughput (Lee and Ni (2013)). In addition, different health status of machines could result in various quality yields, thus impacting product quality directly (Inman *et al.* (2013); Ju *et al.* (2015)). In practice, the health status of a machine can be

identified and described using multiple states based on the features of the machines (Heng *et al.* (2009); Teti *et al.* (2010)). Some research has considered such multi-state models to analyze performance of production lines (e.g., Colledani and Tolio (2012); Cholette and Djurdjanovic (2014); Sloan and Shanthikumar (2002); Bian and Gebraeel (2014)). However, systematic approaches to uncover the potential impact of machine degradation on system performance are still in great demand.

Furthermore, many industrial systems, such as transportation devices, manufacturing systems, energy generation systems and oil pipeline networks, suffer from inevitable failures due to complex degradation processes and environmental conditions. In production systems, these unexpected failures may result in severe consequences, including massive production losses, high corrective replacement (repair) costs and safety hazards to workers and the environment. For this reason, preventive maintenance (PM), which emphasizes effectively avoiding the occurrence of unexpected severe failures, becomes a preferred maintenance choice for practitioners. In fact, maintenance has a great impact on production capacity, product quality and production profitability. On average, 28 % of the total production cost is attributed to maintenance activities in the modern industry (Wang (2012)). For some industries, such as marine, the number can be as high as 40% (Eruguz *et al.* (2017)). Therefore, how to develop effective maintenance policies is of critical importance.

Specifically as the manufacturing process becomes more complicated (e.g., additive manufacturing, composite material manufacturing, renewable energy manufacturing), there is a pressing need to understand the relationship between machine condition degradation and the overall system performance to make better maintenance and product dispatch decisions. Traditionally, much of the throughput analysis of production system research has been carried out based on aggregated machines, assuming machine up or down according to certain distributions and typically ignoring the ma-

chine degradation process (Kimemia and Gershwin (1983); Groenevelt *et al.* (1992); Lin and Gong (2006); Chakraborty *et al.* (2008); Li and Meerkov (2008)). However, as machine conditions impact the performance of production systems in various ways, it is critical to understand their interactions and to study the implication of their integration, which has been a significant gap between the two fields. Some early work has attempted to address this problem. For instance, Lee and Ni (2013) considers different machine condition status during operation and investigates its impact on the overall system performance. Nevertheless, it is based on numerical approach without providing much insight about the system properties. Indeed, the lack of analytical methods and high fidelity models are still the major barrier to prevent full exploitation of the integration of machine condition degradation and system performance.

Besides, due to limited resources such as budget, space, and technicians, it is important to identify the machines or states that are most critical to system performance to guide maintenance and system improvement activities. To this end, bottleneck analysis can serve as an effective tool. For instance, if a machine is in a state that has significant influence on the overall performance of the system, most likely this machine is in an unhealthy state. Thus, to improve the system performance and minimize the chance of encountering catastrophic failures, maintenance and other improvement activities need to be scheduled before its health condition further deteriorates (Fitouhi *et al.* (2017)). Moreover, since for most factories, limited resources can be devoted to the system improvement activities, bottleneck analysis enables finding the most critical component and maximizing the reward of investments. Therefore, systemic and theoretic analysis of system bottlenecks with machine condition degradation is of significant importance and needs to be conducted rigorously.

In this work, a novel integrated approach to answer these questions are introduced. Specifically, this dissertation extends the existing research work (Li, 2008) by con-

sidering multiple machine operating states. It enables modeling machines following different condition degradation process. Analytical solutions are derived to estimate the system performance and system-theoretical properties are investigated to provide insights on how system parameters affect the overall system performance. In addition, bottleneck machines and states are defined, and new novel indicators are created to facilitate bottleneck identification. These methods provide quantitative tools for modeling, analyzing, and improving manufacturing systems with the coupling between machine condition degradation and overall system productivity performance.

Furthermore, the research roadmap is provided, which is shown in Table **?**. This research begins with the system analytics on production systems. The structure and product flow information are analyzed. With the help of sensors and other measurement tools, the real-time information, such as the sensor signals of the equipment and the quality information of the products can be collected. Using the data analytics tools, the machine health conditions can be evaluated. Till this stage, the asset smartness is achieved. Then, by integrating the information of each component in the manufacturing system, together with the system information, such as the production flow information, a system-level model can be built to evaluate the performance of the entire production systems, achieving the system level smartness. Finally, using the results of the system performance and through analyzing the conditions of each component, decisions can be made on the operational actions, which help identify the potential problems and further improve the system performance. Such actions are sent back to the local process controllers and with smart assets, these actions can be executed on-time and thus smart decisions are made. Furthermore, one point worth mentioning that this is a continuous learning and recursive approach. Since the system is dynamic, evaluations and decisions should be made following the change of the system conditions to maintain the best possible evaluation accuracy and system

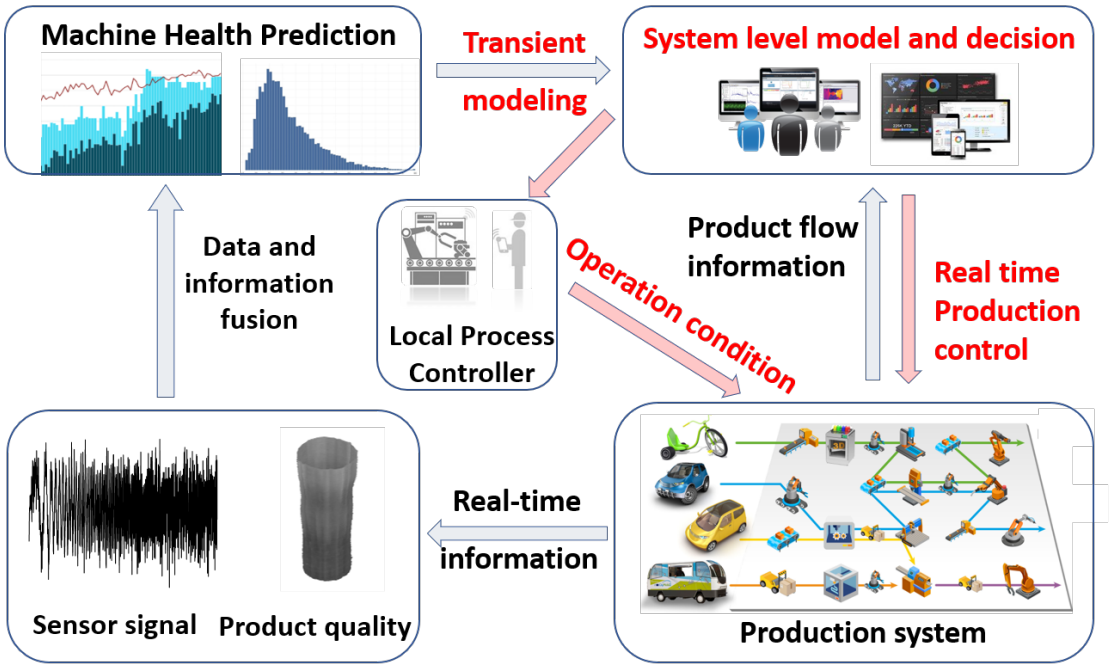


Table 1.1: The Roadmap of the Dissertation Research

performance.

1.2 Outline

This research, therefore, build up bridges from the equipment-level health monitoring to the system level performance evaluation. Furthermore, the conditional based system improvement approaches are also made to better improve the system performance. To summarize, this research develops effective system analytics and decision tools, to monitor, evaluate, improve and control the manufacturing system given real-time machine information.

The rest of this dissertation is organized as follows:

Chapter 2 is devoted to literature review of related research. The current state of the art on degradation modeling, system analytics, conditional-based preventative

maintenance, and the real-time degradation monitoring on production systems are reviewed and discussed.

In Chapter 3, the research problem is formulated and an analytical model is provided to estimate the performance of the serial production lines given machine degradation process. The exact two-machine closed-form expressions are used and aggregation approaches are developed for the serial production lines. The analytical results are the fundamentals for system performance improvements. Then, bottleneck identification tools, derived from the system performance evaluation models, are developed to identify the most critical component that can improve the system performance the most.

In Chapter 4, based on the system performance evaluation models developed in Chapter 3, system improvement tactics are developed. Specifically, the maintenance activities, which can recover the machine conditions suffered from the degradation process, is considered. New models are therefore developed and a Markovian decision process is formulated to find the best maintenance policies that can maximize the overall system production performance for small systems. Furthermore, for general serial production system, an aggregation approach is developed to obtain effective maintenance policies. Numerical experiments are also performed to compare the proposed method with state-of-the-art.

In Chapter 5, the degradation processes are further extended, considering the dynamics on the degradation signals. This work present the algorithm to incorporate the real-time degradation signals of each individual machines to best estimate the production system performance. This is the under-work chapter of the dissertation.

In Chapter 6, the degradation processes under multiple operating conditions are considered, integrating the real-time the dynamics on the degradation signals. This work propose an framework which generate the optimal operating condition control

and maintenance scheduling utilizing the degradation signals. of the machines and parts in the production systems.

Finally, conclusions and future work are presented in Chapter 7. Particularly, the research timeline to complete the entire dissertation work is presented. All the proofs are shown in the Appendix.

Chapter 2

LITERATURE REVIEW

2.1 Modeling of Degradation Process

Machine degradation problem has received significant attention in the recent two decades. To model machine degradation, various physical, statistical, and stochastic models have been developed (Colledani and Tolio (2012); Lee and Ni (2013); Gorjian *et al.* (2010)). Moreover, modern manufacturing systems enable diagnosing conditions of each machine and further classifying them into different operating states (Teti *et al.* (2010)). In literature, mounting research has been devoted to incorporating machine degradation factors into investigating the best maintenance schedule to best utilize resources and achieve optimal system performance from the long run perspectives (Cholette and Djurdjanovic (2014); Sloan and Shanthikumar (2002); Icten *et al.* (2013)). Besides productivity, the coupling of machine degradation process with product quality has also been studied extensively (Tambe and Kulkarni (2014); Kouki *et al.* (2014); Gouiaa-Mtibaa *et al.* (2015)). Some recent research considers additional issues related to machine condition degradation. For example, Colledani and Tolio (2012) considers productivity, quality, maintenance, and logistics issues in an integrated and systematic way. Sloan and Shanthikumar (2002) focuses on the machine degradation in a single stage among the wafer manufacturing process in the semiconductor manufacturing systems. Tambe and Kulkarni (2014) discusses maintenance strategies for a single machine system with continuous machine degradation process. These research typically focuses on single process and ignores the interaction between multiple machines and storage devices. It is thus still unclear how the

degradation process of a single machine could influence the performance of the whole production system.

For serial production lines with multiple machines and limited buffers, the modeling and throughput analysis have been developed in recent years (Ju *et al.* (2013a)). In general, Bernoulli machines (Jacobs and Meerkov (1995a)), geometric machines (Li and Meerkov (2000)), and exponential machines (Jacobs and Meerkov (1995b)) are the typical reliability models developed in the literature. Li and Meerkov (2005) also discusses modeling methods for non-exponential machines cases. The common characteristics of these papers are that they develop systematic approaches to investigate system properties of serial production lines. Using these research as the building block, researchers further apply these theories in different scenarios, solving problems such as quality flow in automobile paint shops (Ju *et al.* (2013b)) and producing perishable products in food factory (Kang *et al.* (2017)). Technically, for analyzing the serial production lines, closed form formulas could be derived only in systems with one or two machines due to the complexity of the problems. For longer lines, approximation algorithms have been derived, typically based on aggregation approach (Chiang *et al.* (2000)), decomposition methods (Le Bihan and Dallery (2000)), and numerical approaches for short lines (Yang *et al.* (2000)). Nevertheless, they simply assume that each machine has only one operating state and one failure state, and ignore machine's condition degradation. Some research work (Tan and Gershwin (2011); Gebennini and Gershwin (2013); Matta and Simone (2016)) have considered multiple operation states in their model. However, they limited the scope of the problems into two machines and one buffer models. Some researchers (Tolio *et al.* (1998)) even realized the meaningfulness of their work when extended to longer lines; nevertheless, they did not provide systematic studies on longer lines nor further discussions on bottleneck identifications.

2.2 Bottleneck Identifications

In addition, bottleneck analysis is regarded as an effective tool in analyzing system properties. Early research tends to focus on machines with lower isolated efficiency or high work in progress in front for performance improvements (Goldratt (1990)). Chiang *et al.* (2000, 2001) provide systematic and theoretic approaches to detect the bottlenecks with regards to system production rate and machine uptime and downtime using blockage and starvation of machines. Furthermore, generalized methods are designed on the basis of Chiang's methods for complex systems, such as production lines with due time (Li (2000)), multiple products (Zhao and Li (2015)), and rework (Biller *et al.* (2010)). In terms of bottlenecks, multiple types of definitions have been extended. Wang *et al.* (2013) discusses machine bottlenecks with respect to the quality of products. Brundage *et al.* (2016) defines the energy bottleneck as the machine with lowest energy efficiency. In addition, Zhou *et al.* (2015) and Guner *et al.* (2016) introduce bottleneck identification methods related to preventative or opportunistic maintenance. However, all these bottleneck related research does not explicitly consider the potential impact of machine condition degradation on the bottleneck identification and mitigation.

2.3 Maintenance Activities in Production Lines

Mounting research has been developed for maintenance in the production systems, starting from the analysis on system performance considering degradation processes using methods such as machine aggregation (Belmansour and Nourelfath (2010); Kang and Ju (2018)). The maintenance strategies are developed primarily focusing on improving system reliability and associated profits. For system reliability, maintenance policies are generated to improve the system performance such as average operating

time (Klutke and Yang (2002)) and machine cycle time in a certain period (Liao *et al.* (2006)). While in another line of research, cost-oriented maintenance policies are developed to minimize the maintenance cost (Alaswad and Xiang (2017)). Specifically, value based maintenance policies have been introduced, focusing on improving the profit of the whole production system (Liyanage and Kumar (2003); Liu *et al.* (2014)). Apart from these efforts, there are also case-specific problems. For example, in aircraft systems, some researchers consider minimizing the risk of violating the aviation laws and regulations (Joo (2009)); in microchip production systems, some research considers minimizing the energy consumptions and wastage caused by machine failure (Luo *et al.* (2015)). Furthermore, there are also works considering multiple objectives together, such as minimizing costs and improving machine availabilities (Liu *et al.* (2013b)). Alaswad and Xiang (2017) delivers the most recent review on developing condition-based maintenance optimization models for stochastic deteriorating systems. Reviews for models that consider replacement, repair, and inspection policies can also be found in Jardine *et al.* (2006) and Wang (2002).

PM is a coherent set of actions that can be applied when the system is still operating, aiming at retaining the system or specified components in certain conditions (Alaswad and Xiang (2017)). Among the mounting efforts on developing system maintenance for serial production lines with buffers, PM has been an emerging issue, drawing significant attention in the past two decades. Following earlier works such as Barlow and Hunter (1960), many models are developed to supervise the decision making procedures, such as determining proper maintenance intervention intervals, optimizing consumption on the spare parts, and minimizing related operating costs (Wang (2012)). Developing PM policies on a single machine system has been well studied (Douer and Yechiali (1994)). However, in recent years, due to the increasing complexity of production systems, more attention has been paid to designing poli-

cies on systems with multiple machines and complex structures (Keizer *et al.* (2017)). Specifically, de Jonge *et al.* (2017) discusses the benefit of conditional-based PM based on surveys on multiple industries.

For serial production systems, several system-level condition-based PM models are developed recently, with primary focus on small scale serial production systems, typically two machines and one buffer systems. For example, Lee *et al.* (2013b) develop optimal policies for two machine systems with continuous time flows. Fitouhi *et al.* (2017) have built an analytical model to find the threshold states for PM. However, they typically do not consider the complex interactions among machines and buffers in the system and merely focus on the performance of individual machines. Gu (2016) considers the information from the system perspective and proposes a PM model utilizing all the information of machine states and buffer levels. However, they assume deterministic PM in terms of both the time it takes to perform PM and its resulting machine status. Such a method has relatively strong requirements on maintenance, thus losing generality and may not be the optimal. Moreover, it is difficult to be generalized for systems with larger scale. Ambani *et al.* (2010) investigates the PM policies for general serial production lines with exponential machines. However, this work simply models the degradation machine process through assuming an exponential distribution on the machine operating time, which is rare supported by the real cases and thus limits the applicability in the real-world scenarios. Therefore, a more general condition-based maintenance model is needed for serial production lines considering the machine degradation process and imperfect maintenance, which motivates the proposed work.

2.4 Real-time Production Performance Including Degradation Signals

The use of real-time sensing information to predict the degrading machine performance are commonly observed in the reliability and prognostics research. The primary focus is to use degradation signals to understand equipment's health condition so as to assess its reliability. In the literature, different degradation modeling such as degradation path models Lu and Meeker (1993), random process models Gebraeel *et al.* (2005), and state space models Zuo *et al.* (1999) have been developed. There is also research on estimating the degradation signals from a single sensor Gebraeel *et al.* (2005); Zhang *et al.* (2015) and multiple sensors Liu *et al.* (2013a); Yan *et al.* (2016). However, such research mainly focuses on individual machines, without further extension to complex manufacturing systems. The closest research related to the problem under investigation in this dissertation is Hao *et al.* (2017), who develops a degradation model for parallel machines to adjust the workloads. However, such model only considers a single stage of the production system, and ignores the complication involved with multiple machines and buffers in the real systems.

2.5 Optimizing Multiple Condition Policy Based on Real-time Degradation Signals

For the work considering degradation process with multiple operating conditions, Yang *et al.* (2007) jointly considers maintenance scheduling and adjustable throughput in their model. However, the operating time are assumed to follow the exponential distribution, which is not true in many cases. The work Hao *et al.* (2015) and Li and Parlikad (2017) consider the real-time degradation signals when making decisions for capacity adjustment. Nevertheless, the maintenance are simply performed after the machine failure, and no preventative maintenance options are discussed. For the POMDP framework, Byon and Ding (2010) develops a condition based maintenance

model to generate the the most cost-effective policy that can be adapted to the wind farm operations, considering the seasonal impact on turbine capacities. However, their model treats the alternating capacities as non-controllable parameters and automatically switched throughout the year, which limits its application scenarios. The work by AlDurgam and Duffuaa (2013) considers the multiple capacity problems, but they assume that all the parameters in the POMDP models are typically provided. They provide no guidelines on utilizing the real-time degradation signals into the framework, which leads to practical concerns.

Chapter 3

INTEGRATED ANALYSIS OF PRODUCTIVITY AND MACHINE CONDITION DEGRADATION: PERFORMANCE EVALUATION AND BOTTLENECK IDENTIFICATION

3.1 Introduction

In this chapter, a novel PM framework is proposed to provide effective PM policies on serial production systems considering machine degradation. The PM plan is formulated based on the failure characteristics of the system considered. Specifically, a new inspection and repair policy is proposed to deal with both internal deterioration and external shock damages. A Markov decision model can be formulated to generate PM policies in small systems with two machines. In particular, three types of machine maintenance actions are included in the work, perfect, minimal, and imperfect repairs. Minimal repair recovers the machine from a failure state while retains the machine degradation status. Imperfect repair refers to the better conditions between the perfect repair and the minimal repair (Pham and Wang (1996); Kouedeu *et al.* (2014)). Such consideration increases the flexibility and adaptability for different systems in real practice (Zhang and Jardine (1998), Wu and Zuo (2010) and Liu and Huang (2010)). Furthermore, this dissertation designs an effective PM policy generating algorithm for general serial production systems using a machine aggregation approach. Such a method extends the systematic decision-making process from two machine systems to large systems.

The main contribution of this chapter is in two folds. First, a systematic approach is developed to generate PM policies for production systems with degrading machines

and finite buffers. Specifically, for machine behaviors, minor repair, full repair, and imperfect PM are considered, which covers a wide scope of practical scenarios. Numerical results show that the proposed method outperforms the state-of-the-art. Second, a novel aggregation-based iterative algorithm is designed to generate optimal PM policies for large scale systems. The algorithm results in a decentralized control approach, where the PM policy for each machine only contains local information from itself and the immediate upstream machine and buffer. Such a method can provide significant performance improvement while keeping the policy simple and easy to implement.

3.2 Problem Formulation

In this chapter, a serial production system is considered, as shown in Figure 3.1. The circles represent machines and the rectangles represent buffers. The direction of arrows represents the flow of working parts through the system. Each machine follows a certain condition degradation process and is subject to failure. Their probability of breaking down varies with respect to different operating conditions. The following assumptions address the characteristics of machines, buffers, and their interactions.

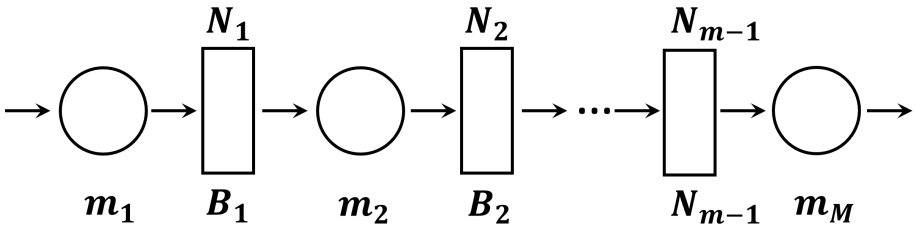


Figure 3.1: Illustration of a Serial Production Line

1. The production system consists of M machines $(m_1, \ldots, m_M)$ and $M-1$ buffers $(B_1, \ldots, B_{M-1})$ separating them.

2. All machines have constant and identical processing time (cycle time). The time frame is thus slotted with cycle time.

3. For machine k, $k = 1, \ldots, M$, there is one failure state F and W_k operating states $1, 2, \ldots, W_k$, which are ordered from the best operating state 1 to the worst one W_k. The machine is subject to state change after finishing each job.

4. All machines could operate normally in any operating state. When the machine is operating at state i, $i = 1, 2, \cdots, W_k$ and $k = 1, \ldots, M$, it can remain at the same state i, transfer to the next worse state $i + 1$, or break down during the next cycle with probabilities $p_{i,i}$, $p_{i,i+1}$, and $p_{i,F}$, respectively. It is assumed that the machine operating at the worse state is more likely to fail, i.e., $p_{i+1,F} \geq p_{i,F}$, $i = 1, 2, \ldots, W_k - 1$ and $k = 1, \ldots, M$.

5. When a machine is under repair, it can transfer back to the best operating state 1 with probability $P_{F,1}$ at the beginning of the next cycle, or remain being repaired with probability $P_{F,F}$. Clearly, $P_{F,F} + P_{F,1} = 1$. It is also assumed that the machine will be as good as new after repair.

6. The degradation processes of all machines are independent.

7. Buffer B_k has finite capacity N_k, $k = 1, 2, \cdots, M - 1$. First-in-first out (FIFO) principle is assumed regarding the buffer outflow process. The buffer occupancy changes at the end of each cycle.

8. Machine m_k is blocked if it is up and the buffer is full at the beginning of a time slot, $k = 1, 2, \cdots, M - 1$. Machine m_M will never be blocked.

9. Machine m_k is starved if it is up and the buffer is empty at the beginning of a time slot, $k = 2, 3, \cdots, M$. Machine m_1 will never be starved.

Note that in practice, the machine condition degradation process can also be characterized by the continuous state space. However, not every single state in the continuous state space necessarily corresponds to different breaking down behaviors (Colledani and Tolio (2012)). Therefore, the continuous state space can be discretized according to the machine's failure behavior and the level of state resolution needed. In addition, assumption 4 shows that machine condition degradation is a non-increasing process and transitions among operating states can only happen between two consecutive ones.

Under assumptions 1-9, the problems to be addressed in this dissertation are as follows:

- Develop analytical methods to evaluate the performance of production systems with machine condition degradation;
- Investigate the system properties. In particular, analyze the effect of single machine condition degradation on the overall system performance;
- Develop bottleneck identification methods to detect the bottleneck machine and operating state.

Solutions to these problems are developed in Sections 3.3-3.4.

3.3 Integrated Model of Machine Condition Degradation and Productivity

The degradation processes for each machine can be generalized into a universal model since it is assumed that they follow similar behaviors, even though the probabilities in different operating conditions might be different. Therefore, the impact of condition degradation process is started first on a single machine and then extended it to analyzing the interactions among multiple machines within the longer lines.

3.3.1 Single Machine Condition Degradation

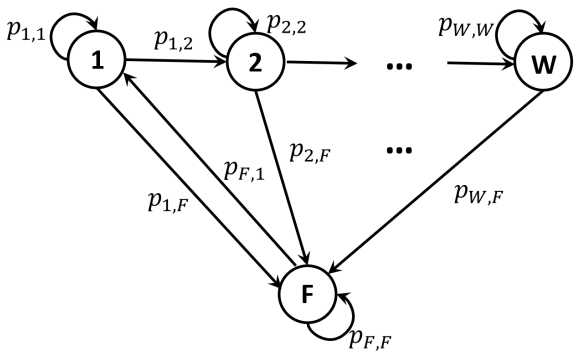


Figure 3.2: Transition Diagram of the Degradation Process for a Single Machine

For the single machine case, the degradation process is described by a discrete time Markov model according to assumptions 3 and 7, as shown in Figure 3.2. The system state represents the machine condition, including W operating states $\{1, 2, \cdots, W\}$ and one failure state F. The transition of states, as shown in Figure 3.2, is described as follows: for any operating state i, it can remain in the same state i, transfer to its next worse state $i+1$, or directly to the failure state F with probability $p_{i,i}$, p_{i+1}, and $p_{i,F}$, respectively, $i \in \{1, 2, \cdots, W-1\}$. The worst operating state W can either remain in the same state or transfer to the failure state. Note that in the degradation process, no reverse transfer is allowed for any operating state i, $i = 1, 2, \ldots, W$. However, a machine can be recovered from the failure state F to state 1 after the repair. Given the above model formulation, a matrix $\mathbf{X}$ can be utilized to express the transition probability:

$$\mathbf{X} = \begin{pmatrix} p_{1,1} & p_{1,2} & 0 & \cdots & 0 & p_{1,F} \\ 0 & p_{2,2} & p_{2,3} & \cdots & 0 & p_{2,F} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & p_{W-1,W-1} & p_{W-1,W} & p_{W-1,F} \\ 0 & 0 & \cdots & 0 & p_{W,W} & p_{W,F} \\ p_{F,1} & 0 & \cdots & 0 & 0 & p_{\mathrm{F,F}} \end{pmatrix}.$$

The above model enables quantifying system performance with machine condition degradation. In this dissertation, long-run performance of a given system is of interest. Specifically, this dissertation focuses on investigating the relationship between steady state system performance and the parameters of the single machine degradation process. To fill the gap between the steady state performance and system parameters, the steady state failure probability π_F is first derived as follows.

Theorem 1 *Given a single machine system defined by assumptions 1-4, the steady state failure probability π_F is obtained as*

$$\pi_F = \frac{1}{1 + \frac{1}{1-p_{1,1}} + \sum_{i=2}^{W} \prod_{j=1}^{i-1} \alpha_j}, \tag{3.1}$$

where

$$\alpha_j = \begin{cases} \frac{p_{j,j+1}}{1-p_{j,j}}, & j = 1, \ldots, W-1, \\ 1, & j = W. \end{cases} \tag{3.2}$$

Proof: Immediately follows by solving balance equations. ∎

It is obvious that the failure probability π_F can be expressed as a function of the degradation model parameters in the matrix $\mathbf{X}$. In addition, a machine's efficiency e can be obtained from Theorem 1, i.e.,

$$e = 1 - \pi_F.$$

3.3.2 State Aggregation

The degradation model depicted above could capture the dynamics of the machine's status change. Then the question arises that how such degradation process for each individual machine would impact the performance of larger systems, specifically with multiple machines and buffers. In the following subsections, the answer to this question is attempted by starting from a simple system with two machines and one buffer in between. First, the operating states for each machine are aggregated while the system uncertainty and degradation dynamics are preserved. Then the aggregated machines are modeled using a Markov chain incorporating their interaction with the buffer.

The idea of state aggregation is that one operating state, denoted as U, is used to represent the aggregated outcome of all the operating states, instead of having multiple operating states in a single machine. Figure 3.3 illustrates the state transition after aggregation, with P representing the aggregated failure probability and R representing the repairing probability. It is worth to mention that R is the same as $p_{F,1}$ in the original transition matrix $\mathbf{X}$. As a result of state aggregation, geometric reliability model is essentially adaptable into the serial production line with machine condition degradation process. The status of each machine can be represented using a set $S = \{U, F\}$, where U is the aggregated operating state and F remains to be the failure state. The transition matrix $\mathbf{Q}$ for the aggregated machine can be expressed as

$$\mathbf{Q} = \begin{pmatrix} 1-P & P \\ R & 1-R \end{pmatrix}.$$

In the aggregated model, the failure state still remains the same as in the original degradation model, with its associated parameters $p_{F,F}$ and $p_{F,1}$ unchanged. However, since all the operating states from the original degradation process are combined

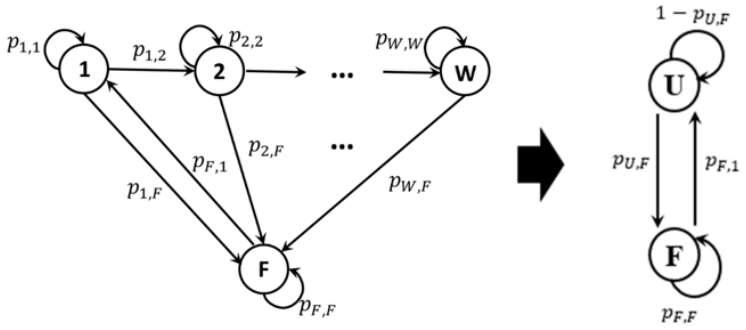


Figure 3.3: State Transition After Aggregation of Condition Degradation Process

together, the aggregated failure probability needs to be reevaluated. Note that in the steady state, π_F derived in Equation (3.1) remains the same in the aggregated model so that it can preserve the original machine degradation characteristics. The resulting failure probability P is shown as follows,

$$P = \frac{p_{F,1}\pi_F}{1-\pi_F} = p_{F,1} + \frac{p_{F,1}}{\sum_{i=1}^{W} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j}, \tag{3.3}$$

where α_j is defined in Equation (3.2). It can be found from Equation (3.3) that the aggregated failure probability is a function of all the parameters of the machine condition degradation process.

3.3.3 Two-machine Case ($M = 2$)

A discrete time Markov chain is utilized to model the two-machine system with machine state aggregation. The state space is defined as $\Omega = \{S_1, n, S_2\}$, where S_1 and S_2 are the status (either up or down) of machine m_1 and m_2, respectively, and n is the number of parts in the buffer. To incorporate the machine condition degradation model into the two-machine system, a geometric model is adopted. To make the model more general, each machine is assumed to follows a unique condition degradation process as defined in the one-machine case. Specifically, machine m_1 has W_1 operating

states with probability matrix $\mathbf{X_1}$, while machine m_2 has W_2 operating states with probability matrix $\mathbf{X_2}$. Using Equation (3.3), the aggregated failure probability for each machine, denoted as P_1 and P_2, respectively, could be obtained.

The system performance measures of interest include production rate PR, machine m_1's blockage probability BL, machine m_2's starvation probability ST, and the work-in-process (WIP) WIP. They could be obtained as follows.

Theorem 2 *Given a two-machine system defined by assumptions 1-4,*

$$\begin{aligned}
\widehat{PR} &= \frac{R_2}{P_2+R_2}\Big[1-\Phi(P_1,R_1,P_2,R_2,N)\Big],\\
\widehat{ST} &= \frac{R_2}{P_2+R_2}\Phi(P_1,R_1,P_2,R_2,N),\\
\widehat{BL} &= \frac{R_1}{P_1+R_1}\Phi(P_2,R_2,P_1,R_1,N),\\
\widehat{WIP} &= \sum_{h=1}^{N} h\mathbf{Pr}[n=h],
\end{aligned} \tag{3.4}$$

where

$$\Phi(P_1,R_1,P_2,R_2,N) = \begin{cases} \frac{P_1\theta_2}{\left(R_1+R_2-R_1R_2\right)\left(P_1+R_1\right)}, & \text{if } N=1,\\ \frac{P_1\eta_1\eta_2\theta_2^2\left(P_2+R_2\right)}{D_1+D_2+D_3+D_4}, & \text{if } N>1, \end{cases} \tag{3.5}$$

and

$$
\begin{aligned}
\eta_1 &= P_1 + P_2 - P_1P_2 - P_2R_1, \\
\eta_2 &= P_1 + P_2 - P_1P_2 - P_1R_2, \\
\theta_1 &= R_1 + R_2 - R_1R_2 - P_1R_2, \\
\theta_2 &= R_1 + R_2 - R_1R_2 - P_2R_1, \\
\rho &= \frac{\eta_2\theta_1}{\eta_1\theta_2}, \\
D_1 &= P_1R_2\eta_1\eta_2\theta_2(P_2 + \theta_2), \\
D_2 &= P_1R_1R_2\eta_2[\theta_2^2 + P(\eta_1 + \theta_1)(\eta_2 + 2\theta_2)], \\
D_3 &= \sum_{j=2}^{N-1} P_1P_2R_1R_2(\eta_2 + \theta_2)^3\rho^{j-1}, \\
D_4 &= P_2R_1\eta_1\theta_2[R_2(\eta_1 + \theta_1) + \eta_2(P_1 + R_1)]\rho^{N-1}.
\end{aligned}
\tag{3.6}
$$

Proof: Immediate follows by combining Equation (3.3) with the results from Li and Meerkov (2008). ■

Using the above theorem, the system performance measures, such as production rate, blockage, and starvation, can be evaluated in closed forms. Specifically, these formulas quantify the impact of machine condition degradation on system performance and enable further investigation on system-theoretic properties.

To investigate the accuracy of performance evaluation obtained from the analytical model, simulation experiments have been conducted. Specifically, the general production setting is considered, where machine m_1 and m_2 can have different numbers of operating states, denoted as W_1 and W_2, respectively. In each simulation run, 3000 time slots are designed as warm-up time and 20,000 following time slots are used to assess the steady state performance. For each set of parameters, 50 replications are conducted. The system parameters are randomly generated using the following criteria:

Procedure 1

(i) *For* $k = 1, 2$, *generate* $W_k \in \{2, 3, \ldots, 10\}$.

(ii) *For* $i = 1, \ldots, W_1$, $j = 1, \ldots, W_2$, *generate* $p^{(1)}_{i,F}, p^{(2)}_{j,F} \in (0, 0.3)$. *Each set of* $p^{(1)}_{i,F}$ *and* $p^{(2)}_{j,F}$ *follows an increasing order, respectively, as described in assumption 3.*

(iii) *For* $i = 1, \ldots, W_1 - 1$, $j = 1, \ldots, W_2 - 1$, *generate* $p^{(1)}_{i,i+1} \in (0, 1 - p^{(1)}_{i,F})$, *and* $p^{(2)}_{j,j+1} \in (0, 1 - p^{(2)}_{j,F})$.

(iv) *For* $i = 1, \ldots, W_1 - 1$, $j = 1, \ldots, W_2 - 1$, *assign* $p^{(1)}_{i,i} = 1 - p^{(1)}_{i,F} - p^{(1)}_{i,i+1}$, *and* $p^{(2)}_{j,j} = 1 - p^{(2)}_{j,F} - p^{(2)}_{j,j+1}$.

(v) *Assign* $p^{(1)}_{W_1,W_1} = 1 - p^{(1)}_{W_1,F}$ *and* $p^{(2)}_{W_2,W_2} = 1 - p^{(2)}_{W_2,F}$.

(vi) *For* $k = 1, 2$, *generate* $p^{(k)}_{F,F} \in (0, 1)$, *and assign* $p^{(k)}_{F,1} = 1 - p^{(k)}_{F,F}$.

(vii) *Generate* $N \in \{1, 2, \ldots, 10\}$.

Let $PR^{sim}, ST^{sim}, BL^{sim}$, denote the mean value of the simulation results. Then accuracy metrics for different performance measurements are defined as follows:

$$\begin{aligned} \delta_{PR} &= \frac{|\widehat{PR} - PR^{sim}|}{PR^{sim}} \times 100\%, \\ \delta_{WIP} &= \frac{|\widehat{WIP} - WIP^{sim}|}{WIP^{sim}} \times 100\%, \\ \delta_{ST} &= |\widehat{ST} - ST^{sim}|, \\ \delta_{BL} &= |\widehat{BL} - BL^{sim}|. \end{aligned} \tag{3.7}$$

The accuracy of PR estimation is shown in Figure 3.4. It can be found that in most cases, the analytical solution falls in the 95% confidence interval of the corresponding simulation trials. Even though in some cases, the analytical solution is on the boundary of the interval, the relative error for PR estimation is small, typically less

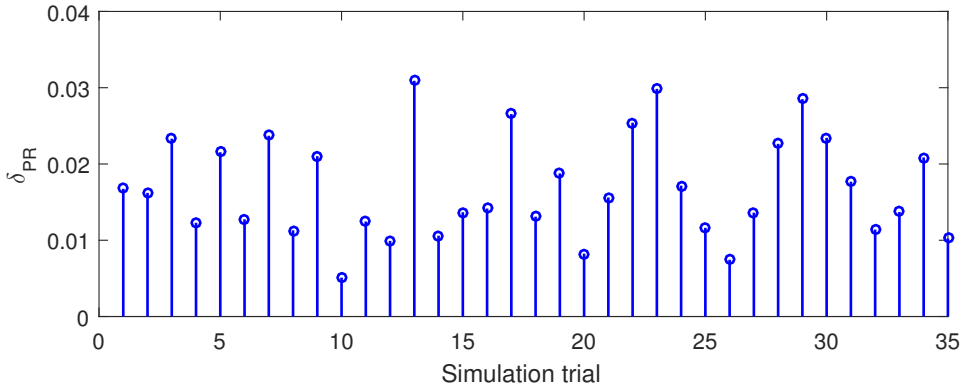


Figure 3.4a: Accuracy for Production Rate Estimation in Serial Lines

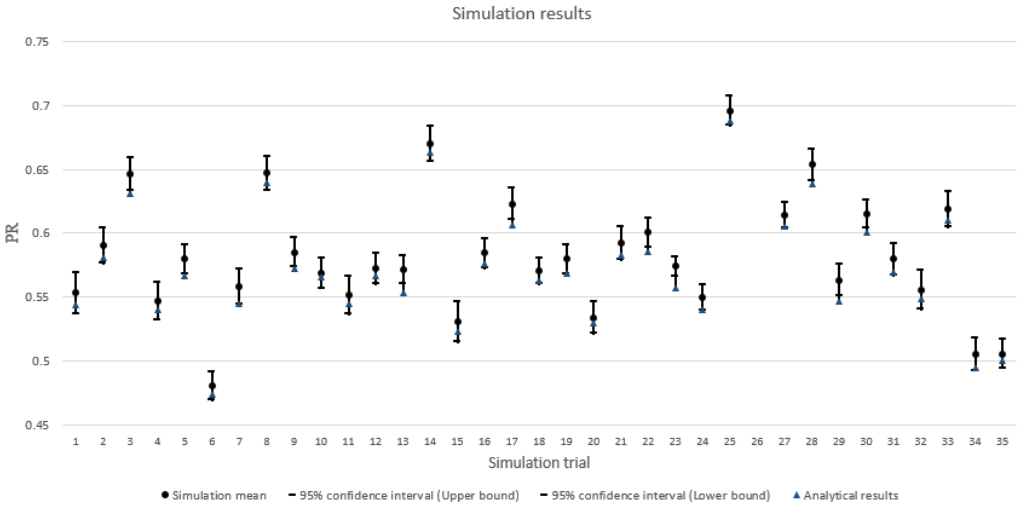


Figure 3.4b: Analytical Results and Simulation Results with Confidence Interval

than 3%. Therefore, it can be concluded that the proposed analytical method could provide enough accuracy for performance evaluation.

3.3.4 General Line ($M > 2$) Case

For a production line with more than two machines, a closed form expression for production rate cannot be found. In order to measure the performance of such systems, a aggregation based recursive procedure can be adopted. To facilitate the

presentation of the procedure, new notations $P_i^b(n)$, $R_i^b(n)$, $P_i^f(n)$, and $R_i^f(n)$, is introduced, representing the failure and repair probabilities in the n^{th} backward and forward aggregation iteration, respectively. The detailed aggregation procedure is shown as follows.

Procedure 2

$$P_i^b(n+1) = P_i + R_i Q(P_{i+1}^b(n+1), R_{i+1}^b(n+1), P_i^f(n), P_i^f(n), N_i), i = 1, \ldots, M-1,$$

$$R_i^b(n+1) = R_i - R_i Q(P_{i+1}^b(n+1), R_{i+1}^b(n+1), P_i^f(n), P_i^f(n), N_i), i = 1, \ldots, M-1,$$

$$P_i^f(n+1) = P_i + R_i Q(P_{i-1}^f(n+1), R_{i-1}^f(n+1), P_i^b(n+1), P_i^b(n+1), N_{i-1}),$$

$$i = 2, \ldots, M,$$

$$R_i^f(n+1) = R_i - R_i Q(P_{i-1}^f(n+1), R_{i-1}^f(n+1), P_i^b(n+1), P_i^b(n+1), N_{i-1}),$$

$$i = 2, \ldots, M. \tag{3.8}$$

The initial conditions for the procedure are

$$P_i^f(0) = P_i, R_i^f(0) = R_i, i = 2, \ldots, M-1, \tag{3.9}$$

with boundary conditions being

$$P_1^f(n) = P_1, R_1^f(n) = R_1, P_M^b(n) = P_1, R_M^b(n) = R_1, n = 0, 1, \ldots \tag{3.10}$$

It has been proved that the procedure will converge to a unique solution (Li and Meerkov (2008)).

Theorem 3

$$P_i^f \triangleq \lim_{n\to\infty} P_i^f(n), P_i^b \triangleq \lim_{n\to\infty} P_i^b(n), R_i^f \triangleq \lim_{n\to\infty} R_i^f(n), R_i^b \triangleq \lim_{n\to\infty} R_i^b(n), i = 1, 2, \ldots, M. \tag{3.11}$$

Therefore, the production rate could be obtained as follows:

$$\widehat{PR} = \frac{R_M^f}{P_M^f + R_M^f} = \frac{R_1^b}{P_1^b + R_1^b}. \tag{3.12}$$

Similar to the two machine case, the expressions of blockage and starvation could also be extended as follows.

$$\begin{aligned} \widehat{BL}_i &= P_i\Phi(P_{i+1}^b, R_{i+1}^b, P_i^f, R_i^f, N_i),\ i = 1, \ldots, M-1, \\ \widehat{ST}_i &= P_i\Phi(P_{i-1}^f, R_{i-1}^f, P_i^b, R_i^b, N_{i-1}),\ i = 2, \ldots, M. \end{aligned} \tag{3.13}$$

To test the accuracy of the aggregation method, simulation experiments have been conducted. The total number of machines are randomly selected between integer 3 to 10, and the remaining parameter selection strategy follows Procedure 1. Similar to the two-machine case, 5,000 time slots are selected as warm-up time, and 20,000 time slots are used to calculate the steady state performance. For each simulation setting, 50 replications are conducted. The relative error δ_{PR} formulated in Equation (3.7) is utilized to illustrate the estimation accuracy. As shown in Figure 3.5, results from 35 cases are reported. The relative errors on production rate estimations are typically below 6%. The WIP difference is within 5%, with the maximum one as 8%. For blockage and estimation, the average difference is within 0.01, and the 95% interval is less than 6% of its mean value. When the system performance is low, typically below 0.45, the production rate estimation error can be higher, i.e., around 5%. However, for most cases, taking the experiments results in Figure 4 as an example, the estimated results fall within the 95% confidence interval of the simulation results. Furthermore, the absolute difference of production rate between the analytical and simulation results is typically within 0.02. The result implies that the proposed method can deliver enough accuracy in system performance estimation.

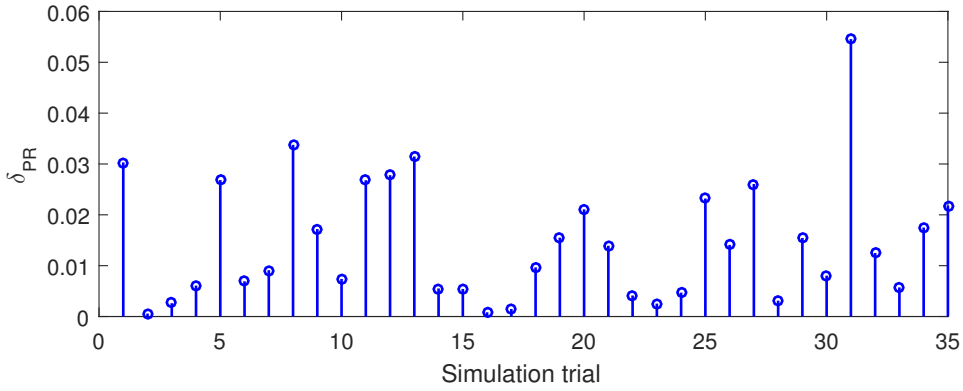


Figure 3.5a: Accuracy for Production Rate Estimation in Serial Lines

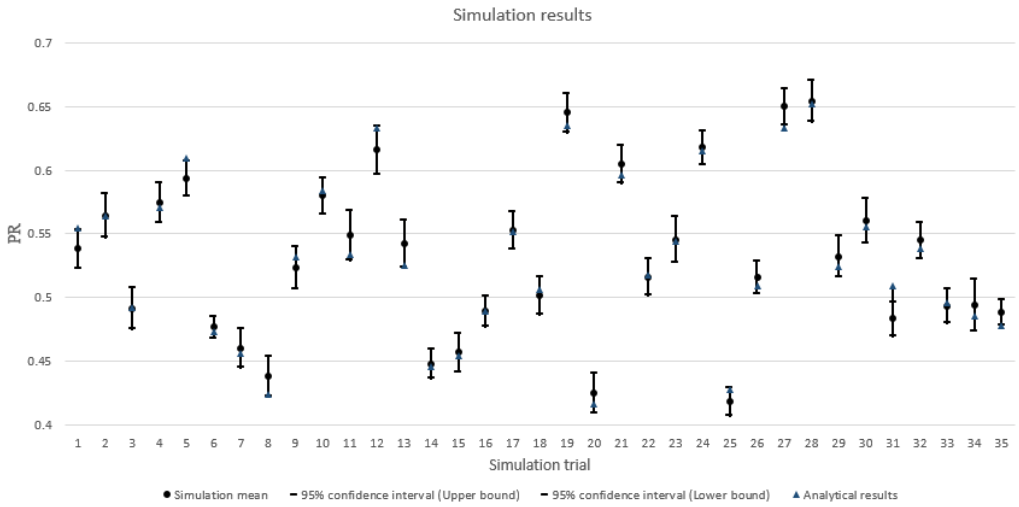


Figure 3.5b: Analytical Results and Simulation Results with Confidence Interval

3.3.5 Structural Properties

In this section, the mathematical model and analytical expressions derived in the previous sections are used to investigate the system-theoretic properties of the production system subject to machine condition degradation. Specifically, this dissertation studies how the parameters of the machine condition degradation process affect the system performance with an emphasis on production rate. To investigate system's structural properties, a two-step approach is undertaken: first study the impact of

machine condition degradation on a single machine, and then extend the result to the two-machine-one-buffer system.

Single Machine System Properties

Corollary 1 *Given a single machine system defined by assumptions 1-4, it follows that*

$$P \geq p_{1,F}, \tag{3.14}$$

where P is the aggregated probability, obtained by Equation (3.3).

Proof: See the Appendix. ■

Corollary 1 provides a relationship between aggregated failure probability and failure probability for the best operating state. Since machine operating at worse state is more likely to fail, the failure probability given that the machine is currently in any one of the operating states P is no smaller than the failure probability given that the machine is always in the best operating state $p_{1,F}$.

Corollary 2 *Given a single machine system defined by assumptions 1-9, P is non-increasing as $p_{k,k}$ increases, $k = 1, 2, \ldots, W$.*

Proof: See the Appendix. ■

Corollary 2 states that if a machine could maintain in one operating state for longer time, then the machine is less likely to fail. This is true because the machine operating at worse state is more likely to fail. An extreme case is that the machine keeps in an operating state k forever, or p_{kk} increases to one, then the machine will never fail, so that P will become zero.

Two-machine System Properties

Corollary 3 *Given a system defined by assumptions 1-4 with buffer size $N = 1$, production rate (PR) is monotonically decreasing w.r.t. P_1 and P_2 and increasing w.r.t. R_1 and R_2.*

Proof: See the Appendix. ■

Corollary 3 reflects the relationship between aggregated probabilities and production rate given a fixed buffer size. This is valid in the real production that if machines break down more frequently (i.e., P_1 or P_2 increases), production rate will be lower. On the other hand, if machines are repaired in a more timely manner (i.e, R_1 or R_2 increases), production rate will also increase.

Corollary 4 *Given a system defined by assumptions 1-9, production rate (PR) is monotonically decreasing w.r.t. $p^{(i)}_{k,k+1}$ and increasing w.r.t. $p^{(i)}_{k,k}$, $k = 1, 2, \cdots, W_i$, $i = 1, 2$.*

Proof: See the Appendix. ■

Corollary 4 combines Corollaries 2 and 3. It implies the impact of machine condition degradation on the system performance. Specifically, when either machine's condition deteriorates faster, it will be more likely to fail, thus leading to lower production rate. Such property provides insights on how degradation parameters affect the overall system performance, and lays the foundation for further bottleneck identification and continuous improvement for production systems subject to machine condition degradation.

3.4 Bottleneck Analysis

3.4.1 Bottleneck Definitions

The bottleneck machine of a system refers to the machine that has the biggest impact on the system performance. In a real production line, people focus heavily on the average amount of time that a machine is operating or a machine is needed from failure to fully repaired. For a geometric machine, the average uptime and downtime for machine i could be expressed respectively as follows:

$$T_{u_i} = \frac{1}{P_i},\ T_{d_i} = \frac{1}{R_i},\ i = 1, 2, \ldots, M. \tag{3.15}$$

In this section, definitions for three types of bottlenecks are provided. The first type of bottleneck is the uptime bottleneck (U-BN). It refers to the machine which has the biggest improvement on production rate when the average uptime of this machine is increased.

Definition 1 *Machine m_i is the uptime bottleneck (U-BN) if for any $j \neq i$,*

$$\left|\frac{\partial PR}{\partial T_{u_i}}\right| > \left|\frac{\partial PR}{\partial T_{u_j}}\right|. \tag{3.16}$$

Another type of bottleneck is the downtime bottleneck (D-BN). It refers to the machine which has the biggest improvement on production rate when the average repair time of this machine is decreased.

Definition 2 *Machine m_i is the downtime bottleneck (D-BN) if for any $j \neq i$,*

$$\left|\frac{\partial PR}{\partial T_{d_i}}\right| > \left|\frac{\partial PR}{\partial T_{d_j}}\right|. \tag{3.17}$$

The third type of bottleneck is the state bottleneck (S-BN). It refers to the most sensitive transition probability in an operating state (for example, p_{kk} for operating state k), which has the biggest improvement on production rate of the production line than any other state in the same machine and any state in other machines.

Definition 3 *Operating state k in Machine m_i is the state bottleneck (S-BN) if for any machine j and any different operating state l,*

$$\left|\frac{\partial PR}{\partial p_{k,k}^{(i)}}\right| > \left|\frac{\partial PR}{\partial p_{l,l}^{(j)}}\right|. \tag{3.18}$$

Even though the definitions above provide rigorous ways of identifying bottleneck machines, these formulas cannot be directly applied in practice since the partial derivatives involved cannot be evaluated analytically and are difficult to obtain from factory floor measurements. Therefore, indirect methods need to be pursued. Specifically, indicators are derived for each type of bottlenecks. Using these indicators, identification rules based on arrow assignment rules are formulated to detect bottlenecks. The details are laid out as follows.

3.4.2 U-BN Identification

Two-machine Case

To derive indicators for the U-BN, this work starts with the simplest case where there are only two machines and one buffer with buffer size $N = 1$. Based on the analytical expression of production rate in this simplest case, a closed form indicator could be formulated as follows.

Theorem 4 *In a two-machine-one-buffer system with buffer size $N = 1$, machine m_2 is the uptime bottleneck if and only if*

$$ST_2 e_1 P_1 < BL_1 e_2 P_2. \tag{3.19}$$

Proof: See the Appendix. ∎

Theorem 4 provides rigorous indicators to compare the severity of uptime of two machines. It implies that uptime bottleneck is influenced by not only machine efficiency, but also by other factors such as blockage and starvation, which capture the

inherent interactions between machines and the buffer. For example, if the second machine is more likely to be starved, then the first machine is more likely to be the uptime bottleneck, because the uptime of the upstream machine might not be long enough to produce enough parts. Therefore, each individual term is not sufficient enough to quantify the severity of each machine, but rather the combined multiplication results need to be considered as a whole to detect the bottleneck.

General Case

However, for the two-machine case with $N > 1$ and longer lines, it is impossible to obtain a closed form expression for production rate and its partial derivative. Instead, the bottleneck indicator derived for two-machine case with $N = 1$ is applied. In addition, an arrow assignment rule is designed to identify the uptime bottleneck for multiple machine systems. The detailed rule is shown as follows.

Arrow Assignment Rule for U-BN: If $BL_i e_{i+1} P_{i+1} > ST_{i+1} e_i P_i$*, assign an arrow from* m_i *to* m_{i+1}*; otherwise, assign an arrow from machine* $i+1$ *to machine* i*,* $i = 1, \ldots, M-1$.

Following the assigned arrows, one can trace upward or downward to find the bottleneck. However, since the arrow assignment rule only compares two consecutive machines, it is likely to end up with multiple "bottlenecks" being identified, namely local bottlenecks. In order to find the real bottleneck of the system (global bottleneck), a heuristic severity index is introduced, which quantified the likelihood of the identified local bottleneck being the global bottleneck. Specifically, bottleneck severity for m_i is denoted as S_i^u and is formulated as below.

$$
\begin{aligned}
S_1^u &= |ST_2 e_1 P_1 - BL_1 e_2 P_2|, \\
S_i^u &= |ST_{i+1} e_i P_i - BL_i e_{i+1} P_{i+1}| + |ST_i e_{i-1} P_{i-1} - BL_{i-1} e_i P_i|, \quad i = 2, \ldots, M-1, \\
S_M^u &= |ST_M e_{M-1} P_{M-1} - BL_{M-1} e_M P_M|.
\end{aligned}
\tag{3.20}
$$

According to the arrow assignment rule for U-BN, if one machine has no emanating arrow from it, then it is the bottleneck machine. If multiple machines have such behavior, then the one with the largest severity value is the bottleneck.

An example is provided in Figure 3.6 to illustrate how the U-BN is identified. All system parameters are randomly picked and shown on the figure. First, arrows are assigned from the larger indicator value to the smaller one. For example, $BL_1 P_2 e_2$ is 0.0339 while $ST_2 P_1 e_1$ is 0.0004. Therefore, an arrow points from machine 1 to machine 2. Once the arrows are assigned for each consecutive machine pair, it is obvious that there is no arrow emanating from machine 3 or machine 5. Thus they form the candidates for machine bottleneck. By calculating their severity values using Equation (3.20), machine 5 has larger severity value 0.0303 compared to 0.0095 for machine 3. Therefore, machine 5 is identified to be the primary uptime bottleneck.

Further, to validate that the identified bottleneck is the real bottleneck, simulation experiments are conducted, with 1000 replications per experiment and 100,000 time-slot simulation length. The partial derivatives are estimated using parameter sensitivity study by changing the corresponding parameters in the derivatives with the rate $\Delta = 0.1$. Then the derivative values can be expressed in the format of $\frac{\widetilde{PR} - PR}{\Delta(\cdot)}$, where $\widetilde{PR}$ is the after the parameter change and PR is the one with the original parameter settings. The real bottleneck is target with the largest derivative value. Moreover, the 95% confidence interval for all the derivative values are provided. These settings are followed throughout the remaining experiments in the dissertation.

In Figure 6, it can be found that the bottleneck identified by the indicators matches the results by performing parameter sensitivity study on the corresponding partial derivatives. For example, the partial derivative obtained numerically for machine 5 is the largest with the mean value 0.0161. Furthermore, the 95% confidence interval of machine 5 is overwhelmingly larger than the values of the other machines, with no overlaps. Therefore, the indicator is an effective approach to identify the real bottleneck.

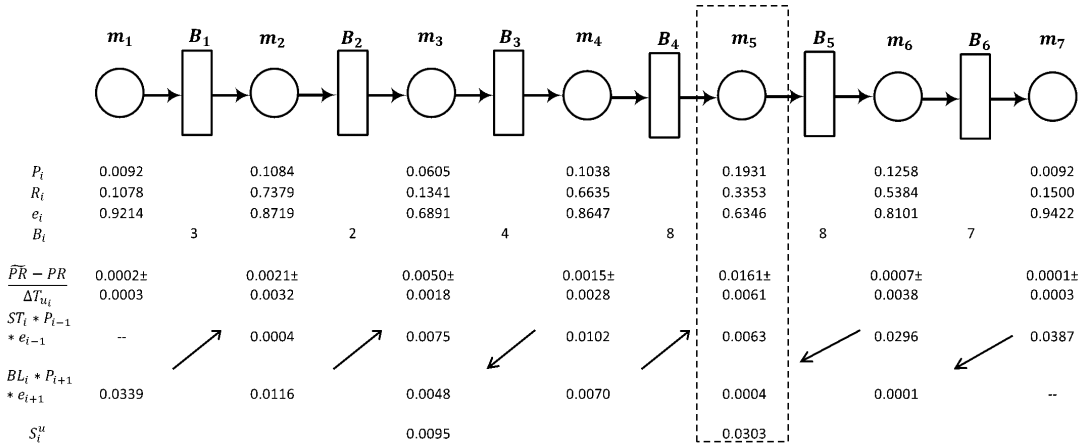


Figure 3.6: An Illustrative Example for Detecting U-BN

3.4.3 D-BN Identification

Two-machine Case

To identify the D-BN, the similar idea is followed as illustrated in the previous section. Indicators for D-BN in the two-machine case when buffer size $N = 1$ is first rigorously derived. Then the obtained bottleneck indicator from the simplest case is extended to more general systems. Based on the D-BN's definition, the indicator for D-BN is formulated as follows.

Theorem 5 *In the two machines and one buffer system with buffer size $N = 1$, machine m_2 is the D-BN bottleneck if and only if*

$$ST_2 e_1 R_1 < BL_1 e_2 R_2. \tag{3.21}$$

Proof: See the Appendix. ■

Similar to the indicators for U-BN, efficiency and blockage/starvation also appear in the indicators for D-BN, inferring that these terms have significant influence on the downtime of machines. For example, if one machine is more likely to be starved, then more time might be needed to repair the failure machine ahead of it. It is also observed that the repairing probabilities are included in the indicators. Intuitively, if a machine is more likely to be repaired once it fails, then its downtime has little influence on the overall production rate. Besides, it should also be noted that the whole indicator has the strongest explanation power to the bottleneck identifications; a single term in the indicator might have straightforward meaning but not necessarily lead to the true bottleneck. Therefore, a systematic approach integrating the conditions of all the machines is necessary.

General Case

For M-machine serial production lines, we can identify the bottleneck using similar analytical techniques. an arrow assignment rule will be established similar to that in the U-BN identifications.

Arrow Assignment Rule for D-BN: If $BL_i e_{i+1} R_{i+1} > ST_{i+1} e_i R_i$, assign an arrow from m_i to m_{i+1}; otherwise, assign an arrow from machine $i+1$ to machine i, $i = 1, \ldots, M-1$.

In addition, similar bottleneck severity for m_i, denoted as S_i^d, is developed as follows:

$$
\begin{aligned}
S_1^d &= |ST_2 e_1 R_1 - BL_1 e_2 R_2|, \\
S_i^d &= |ST_{i+1} e_i R_i - BL_i e_{i+1} R_{i+1}| + |ST_i e_{i-1} R_{i-1} - BL_{i-1} e_i R_i|, \quad i = 2, \ldots, M-1, \\
S_M^d &= |ST_M e_{M-1} R_{M-1} - BL_{M-1} e_M R_M|.
\end{aligned}
\tag{3.22}
$$

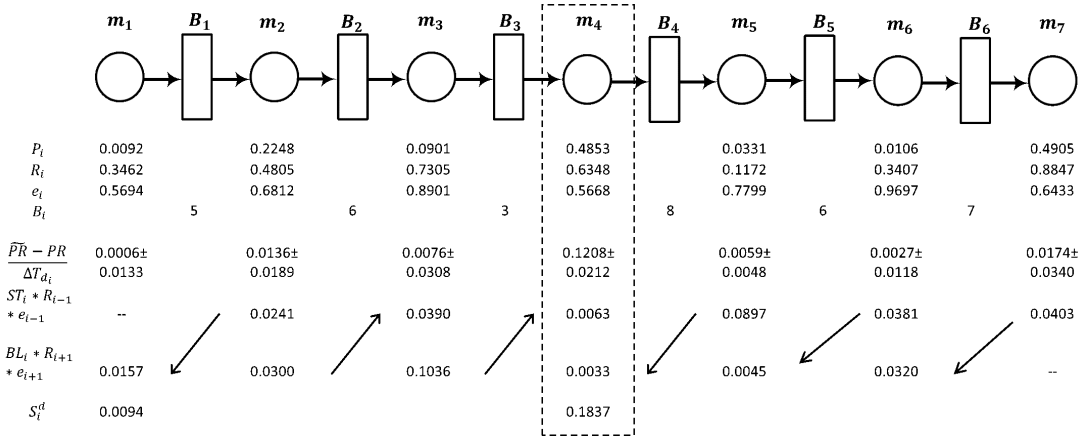


Figure 3.7: An Illustrative Example for Detecting D-BN

An example is provided in Figure 3.7 to illustrate how the D-BN is identified. The production system consists of seven machines and six buffers. System parameters are shown in the figure. The estimation of partial derivatives followed the same numerical method as in the U-BN identification. By applying the arrow assignment rule for D-BN and following the arrow directions, machine 1 and 4 are identified to be the downtime bottleneck. Using the severity index derived in Equation (3.22), machine 4 is determined to be the primary bottleneck, which delivers the same result as what the partial derivative implies.

3.4.4 S-BN Identification

The state bottleneck is different from the previous two types of bottlenecks in the sense that it not only specifies the bottleneck machine but also targets on a specific bottleneck operating state of that machine. According to the definition, S-BN is quantified using the following partial derivative,

$$\left|\frac{\partial PR}{\partial p_{k,k}^{(i)}}\right| = \left|\frac{\partial PR}{\partial P_i}\right| \cdot \left|\frac{\partial P_i}{\partial p_{k,k}^{(i)}}\right|. \tag{3.23}$$

Using the chain rule, the derivative on the left hand-side can be divided into two components with one for the inter-machine level and one for the intra-machine level. Therefore, to identify the state bottleneck for a system, a two-step approach is conducted. First, the state bottleneck for each individual machine is determined. Then the severity of these state bottlenecks will be compared relative to all machines within the system.

One-machine Case

For a single machine line, the S-BN could be obtained according to the definition, and Theorem 6 shows the findings.

Theorem 6 *Operating state l in a machine is the state bottleneck if for any other operating state h,*

$$\left|\frac{\partial P}{\partial p_{l,l}}\right| > \left|\frac{\partial P}{\partial p_{h,h}}\right|, \tag{3.24}$$

where

$$\frac{\partial P}{\partial p_{k,k}} = -\frac{P^2}{(1-p_{k,k})^2}\prod_{j=1}^{k-1}\alpha_j\Big(1-\sum_{i=k+1}^{W}\frac{p_{k,F}}{1-p_{i,i}}\prod_{j=k+1}^{i-1}\alpha_j\Big), k=1,2,\ldots,W. \tag{3.25}$$

Proof: See the Appendix. ∎

Using this theorem, the severity of each operating state for an arbitrary machine can be easily determined, thus solving the problem of identifying intra-machine level state bottlenecks. Since all machines have independent degradation process, such expression can be readily extended to longer lines with multiple machines.

General Case

In large systems with multiple machines, the impact of operating states in both intra- and inter-machine levels needs to be considered. Specifically, the bottleneck indicators for U-BN are combined with the single machine state bottleneck indicators to incorporate the influence of operating states on both levels. Therefore, a two-step approach is formulated as follows.

Procedure 3 *Step 1: Find the state bottleneck O_i for each machine i using the following severity index w_i.*

$$\begin{aligned} w_i &= max\Big\{\frac{\partial P_i}{\partial p_{k,k}^{(i)}}|k=1,2,\ldots,W_i\Big\}, i=1,\ldots,M, \\ O_i &= arg_k\ max\Big\{\frac{\partial P_i}{\partial p_{k,k}^{(i)}}|k=1,2,\ldots,W_i\Big\}, i=1,\ldots,M, \end{aligned} \tag{3.26}$$

where $\frac{\partial P_i}{\partial p_{k,k}^{(i)}}$ is defined in Equation (3.25).

Step 2: Multiple single machine severity index w_i with U-BN indicators for each machine. Follow the arrow assignment rule below to detect the overall state bottleneck.

Arrow Assignment Rule for S-BN: If $w_i BL_i e_{i+1} P_{i+1} > w_{i+1} ST_{i+1} e_i P_i$, assign an arrow from m_i to m_{i+1}; otherwise, assign an arrow from machine $i+1$ to machine i, $i = 1,\ldots,M-1$.

Using the above rule, if only one machine has no emanating arrow, then it is the machine that contains the overall state bottleneck. If multiple "bottleneck" machines

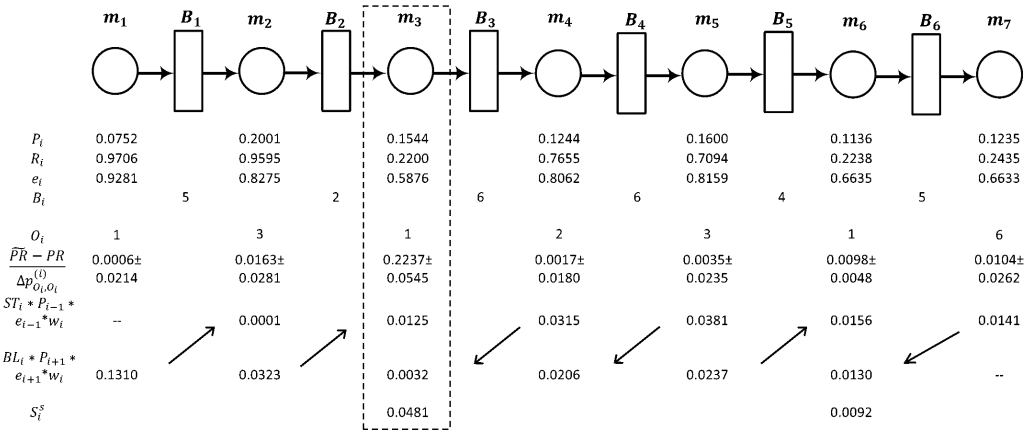


Figure 3.8: An Example for Detecting S-BN

are identified, the following severity index S_i^s for machine i is utilized to determine the machine which contains the primary bottleneck state (true bottleneck state).

$$\begin{aligned} S_1^s &= |w_2 ST_2 e_1 P_1 - w_1 BL_1 e_2 P_2|, \\ S_i^s &= |w_{i+1} ST_{i+1} e_i P_i - w_i BL_i e_{i+1} P_{i+1}| + |w_i ST_i e_{i-1} P_{i-1} - w_{i-1} BL_{i-1} e_i P_i|, \\ &\qquad\qquad i = 2, \ldots, M-1, \\ S_M^s &= |w_M ST_M e_{M-1} P_{M-1} - w_{i-1} BL_{M-1} e_M P_M|. \end{aligned} \tag{3.27}$$

In the indicator of the arrow assignment rule, w_i reflects the level of influences that the increase of probability $p_{k,k}^{(i)}$ could influence the aggregated failure probability of a certain machine, namely intra-machine level influence. Indicators for U-BN are used to evaluate how aggregated probability could impact the overall system performance, which characterizes the inter-machine level influence.

Figure 3.8 illustrates how the S-BN is identified using the two-step procedure. Following step 1, the bottleneck state for each machine can be found and shown in row O_i. For example, state 1 is the bottleneck state for machine 3 in this case. Then, following the arrow assignment rule, machine 4 is the machine containing the bottleneck state since it has no emanating arrow and also possesses larger severity

value compared with machine 6. Therefore, state 3 of machine 4 is the bottleneck state for the whole production line. It meets with the result using the numerical estimation on partial derivatives, of which state 3 of machine 4 has the largest value, 0.0481, identified as the bottleneck.

3.4.5 Accuracy of Bottleneck Identification Rules

To evaluate the accuracy of these bottleneck identification rules, 1000 sets of experiments have been conducted. The system parameters are randomly generated using the following procedure:

Procedure 4

1. *Generate machine number* $M \in \{2, \ldots, 8\}$.
2. *Generate buffer size* $B_i \in \{1, \ldots, 10\}$, $i = 1, \ldots, M-1$.
3. *Use Procedure 1 step* (i) *to* (v) *to generate the operating states transferring probabilities in matrix* $\mathbf{X_i}$, $i = 1, \ldots, M$.
4. *Generate machine efficiency* $e_i \in (0.5, 1)$, $i = 1, \ldots, M$. *Then* R_i *could be calculated using formula* $e_i = \frac{R_i}{P_i + R_i}$, *where* P_i *is obtained by aggregating the probabilities in matrix* $\mathbf{X_i}$, $i = 1, \ldots, M$.
5. *For matrix* $\mathbf{X_i}$, *assign* $p_{F,1}^{(i)} = R_i$, *and* $p_{F,F}^{(i)} = 1 - R_i$, $i = 1, \ldots, M$.

For each experimental case, the partial derivatives are estimated using parameter sensitivity study by changing the corresponding parameters in the derivatives with the rate of 0.1. The real bottleneck is target with the largest derivative value and then compared with the one obtained using the corresponding indicator. To provide more clarification on numerical experiments, an additional table with several representative

cases is provided. In Figure 9, the bottlenecks identified using indicators and the true bottlenecks are reported, together with their associated derivative values. For example, in case one, the indicator finds the true uptime bottleneck (machine 2), with the derivative value of 0.0212. It implies that if a slightly improvement (1%) can be achieved on the uptime of machine 2, the production rate for the overall system will increase roughly by 0.212%. For the S-BN cases, in case 1, the expression [3,2] means that operating state 2 in machine 3 is the bottleneck. Other cases follow the same expression.

As shown in Figure 3.9, the accuracy for three bottleneck identification methods are typically above 80%, suggesting that the derived indicators for U-BN, D-BN, and S-BN could efficiently identify the real bottlenecks in practice. For those cases in which the bottleneck indicators could not identify the correct bottleneck, the derivatives of the real bottleneck and the misidentified bottleneck by indicators are typically close to each other. For example, in Table 1, the misidentified cases, marked in bold letters, imply that the differences of the derivatives between the true bottleneck and the identified one are below 0.005, or 0.5%. Furthermore, it can be found out that over 65% of the mis-identification cases, the indicators detect the machine which has the second largest derivative values (named as "second bottleneck"). Considering the correct cases and the cases with the second bottleneck together, the total accuracy for all the three types of bottlenecks could be over 90%. This implies that even though the bottleneck indicators cannot guarantee to identify the true bottleneck in some cases, they still can provide useful guidance on effective improvement strategies, which is near optimal and practically useful.

In practice, the bottleneck identification method provides guidelines on system maintenance and improvements, especially in complex manufacturing systems. Using such a method, practitioners can devote their limited resources (like maintenance

Case	System parameters			BN-type	Indicator results		True results	
#	M	N	W		BN	$\partial PR/\partial(\cdot)$	BN	$\partial PR/\partial(\cdot)$
1	3	[2,3]	[4,4,3]	U-BN	2	0.0212	2	0.0212
				D-BN	2	-0.0347	2	-0.0347
				S-BN	[3,2]	0.0385	[3,2]	0.0385
2	4	[5,2,5]	[4,3,3,4]	U-BN	3	0.0323	3	0.0323
				D-BN	**3**	**-0.0440**	**2**	**-0.0478**
				S-BN	[3,2]	0.0646	[3,2]	0.0646
3	5	[8,3,4,9]	[8,5,6,7,8]	**U-BN**	**5**	**0.0088**	**4**	**0.0099**
				D-BN	4	-0.0264	4	-0.0264
				S-BN	[4,1]	0.0204	[4,1]	0.0204
4	5	[4,5,10,5]	[5,4,5,5,4]	U-BN	1	0.0319	1	0.0319
				D-BN	1	-0.0572	1	-0.0572
				S-BN	[1,4]	0.0617	[1,4]	0.0617
5	6	[5,5,2,9,3]	[6,6,8,6,3,8]	U-BN	6	0.0071	6	0.0071
				D-BN	2	-0.0272	2	-0.0272
				S-BN	**[5,2]**	**0.0214**	**[4,1]**	**0.0236**

Table 3.1: Illustrative Numerical Examples of Bottleneck Identifications

staffs and tools) to the bottleneck machines so as to improve the system performance more effectively. Specifically, the state bottleneck is a critical indication for preventative maintenance. For example, the case 4 in Table 1 shows that operating state 4 in machine 1 is the S-BN. It implies that the system production rate has the largest improvement if the machine can stay in this state longer by preventative maintenance. Therefore, practitioners can consider improving machine 1s condition when it reaches to state 4. In other words, the bottleneck operating state can be treated as a

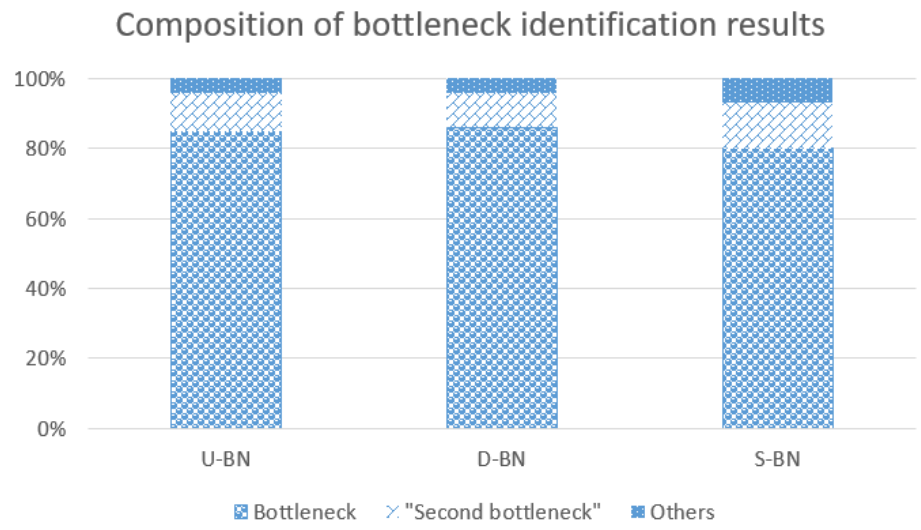


Figure 3.9: Accuracy of Bottleneck Identifications

threshold policy for preventative maintenance. The practitioners can further consider the maintenance plans for the best system improvements.

3.5 Conclusions and Future Work

In this chapter, a serial production system with machine condition degradation process is introduced and analyzed. Analytical formulas and approximation algorithms are developed to quantify system's overall performance considering machine condition degradation. As validated by simulation experiments, the developed methods provide an robust way to evaluate system performance. In addition, structural properties are provided to investigate the effect of machine degradation parameters on system performance. Finally, three types of bottlenecks are presented and their associated indicators and arrow assignment rules are developed to find the machine and the state that have the most influence on overall system performance. Such methods can provide practitioners comprehensive and quantitative tools to uncover the dynamics of serial production systems with machine condition degradation.

To extend the research, the following topics can be addressed in future work.

- Extend the model to more complicated manufacturing systems, such as assembly systems and parallel machine networks;
- Generalize the machine degradation model, such as Wiener degradation process and the hidden Markov model;
- Investigate quality issues associated with the degradation process to evaluate the performance of a production system;
- Develop real-time decision making and control strategies to improve the system performance;
- Incorporate the on-line data thread into the model to study the transient performance of production systems with machine condition degradation.

Chapter 4

FLEXIBLE PREVENTATIVE MAINTENANCE FOR SERIAL PRODUCTION LINES WITH MULTI-STAGE DEGRADING MACHINES AND FINITE BUFFERS

In this chapter, a serial production system is considered and illustrated in Figure 5.1. The circles and the rectangles represent the machines and buffers respectively. The flow of the working parts within the manufacturing line is directed using arrows. For each machine, a specific degradation model is followed, with transition probabilities depending on the operating states. The following assumptions describe the characteristics of the machines, buffers, and their interactions.

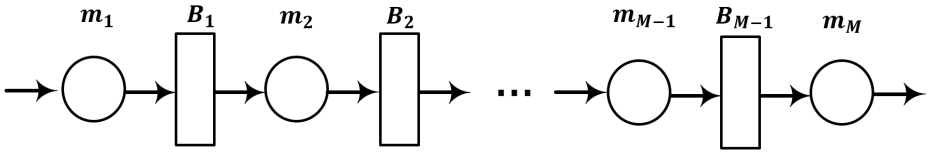


Figure 4.1: Illustration of a Serial Production Line

1. There are in total M machines in the line. The processing time, or the cycle time for each machine is identical (sychrounous production lines). So the time could be slotted by the processing time.

2. The capacity of the buffer B_k is finite, denoted as N_k.

3. Machine m_k is blocked if it is up and the buffer B_k is full at the beginning of a time slot, $k = 1, 2, \cdots, M-1$. The last machine m_M will never be blocked.

4. Machine m_k is starved if it is up and the buffer B_k is empty at the beginning of a time slot, $k = 2, 3, \cdots, M$. The first machine m_1 will never be starved.

5. There are W_k operating states for machine k, $k = 1, \ldots, M$. The states are ordered from the best operating state 1 to the worst one W_k. Th machine state changes at the end of each cycle. The degradation processes among different machines are independent.

6. For any operating state, a machine can produce one part in each cycle. The transition rules among the machine states are as follows (shown in Figure 4.2). If no maintenance is performed (as the solid arrow shows), for a machine working at state i, $i = 1, 2, \cdots, W_k$ and $k = 1, \ldots, M$, it can remain at the same state i, transfer to the next worse state $i + 1$, or break down to state i^F during the next cycle with probabilities $p_{i,i}$, $p_{i,i+1}$, and p_{i,i^F}, respectively. The failure probability is assmued to be higher for a machine working in worse states, i.e., $p_{i+1,(i+1)^F} \geq p_{i,i^F}$, $i = 1, 2, \ldots, W_k - 1$ and $k = 1, \ldots, M$.

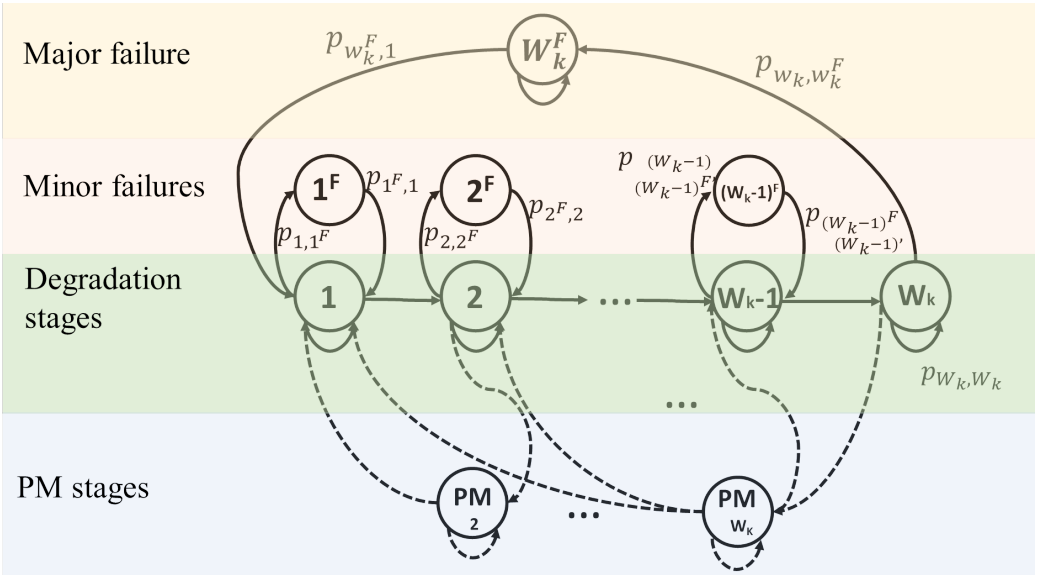


Figure 4.2: Machine State Transition Diagram with PM Actions

7. Given that PM is not performed and machine m_k is in state $\{1, 2, \ldots, W_k - 1\}$, $k = 1, \ldots, M$, a random failure may occur. Such failure is treated as minor failure, without influencing the machine degradation stage. At this time, a minimal repair is considered to handle the failure, restoring the machine state right to the state it fails from. Such minimal repair is independent of machine state and one unit time is required for any minimal repair.

   Remark: Minimal repair is commonly observed in multi-component machines, such as milling machines, drilling machines, and grinding machines (Ja *et al.* (2001)). The random failures in the assumption refers to some minor failures that are independent of the machine degradation stages. For example, in a drilling machine, the burn-out of a fuse or the falling of the drill can cause the stop of a machine. However, these failures may not necessarily lead to full maintenance service (i.e., replace the drill with a brand-new one). Instead, in practice, engineers can replace the fuse or install the drill back to the machine in short time and restart the machine. Such an operation has minimal impact on the degradation level of the machine.

8. When machine m_k fails from state W_k, $k = 1, \ldots, M$, the failure is considered critical and full maintenance (i.e., repair) should be performed. At this time, the machine will be fully repaired and restored to the best operating state (state 1, or the "brand new" state). The repairing probability for machine m_k is $p_{w_k^F,1}$.

9. When machine m_k is in any imperfect operating state $(2, \ldots, W_k)$ at the beginning of a time slot, the decision maker can choose whether to perform PM or not (as the dashed lines suggest in Figure 4.2).

10. If PM is performed, then imperfect PM is allowed to recovery the machine state to any better operating states. The transition probabilities for PM are state

dependent, with notation $p_{PM_i,j}$ representing that a machine is recovered to state j when it is under PM state at state PM_i. The PM recovery probabilities is assumed to be lower for better operating states. For example, performing PM from state 4, the recovering probability to state 3 is higher than that to state 1.

Under the above assumptions, the problem to be studied in this chapter is:

- Provide an analytical approach for performance evaluation and PM decision making for the two machine and one buffer systems;
- Investigate an effective and computationally efficient algorithm for developing the optimal PM policies for general serial production systems.

Solutions to these problems are developed in the following sections.

4.1 A Markov Decision Process Approach for a 2M1B System

The research starts first with the simplest system with two machines and one buffer (2M1B). Based on the system descriptions illustrated above, the optimal maintenance scheduling problem can be formulated as a Markov decision process for a 2M1B system. The system states, action spaces, transition probabilities, and rewards are defined and modeled as follows.

4.1.1 State and Action Space

A 2M1B system can be modeled as a discrete time Markov chain based on the assumptions. The state space of the system composes contains the buffer level and the status of both individual machines. The states of the system at time t can be expressed as $\mathbf{S(t)} = \{h(t), S_1(t), S_2(t)\}$, where $S_1(t)$ is the state of the first machine, $S_2(t)$ is

the state of the second machine, and $h(t)$ denotes the buffer level. Through this way, the state variation of both machines and the buffer are tracked and considered for the PM actions.

The structure of the single machine state $S_k(t)$ for machine $m_k, k = 1, 2$ can be further categorized into three types: operating states, failure states, and PM states. For operating states, there are two components: $K_1 = \{1, 2, \ldots, (W_k-1)\}$ and W_k. K_1 is a set of intermediate operating states characterizing a machine's degradation stage, with further degradation to worse operating states. State W_k is the boundary state for the degradation process and no further degradation is allowed beyond this state. Similarly, for the failure states, there are two corresponding components matching with the operating states, $K_2 = \{1^F, 2^F, \ldots, (W_k - 1)^F\}$ and W_k^F. They represent the two types of failure modes and the corresponding repairing approaches. State j^F in K_2 is the state that a machine fails from state j and minimal repair will be performed. State W_k^F means that the machine fails and major repair has to be performed at that state. For the preventative maintenance, set Γ_k denotes all possible combinations of the maintenance actions for machine m_k. When the machine approaches to state i, $i \in \{2, \ldots, W_k\}$, the maintenance can recovers the machine to state j, $j \in \{1, \ldots, i-1\}$. Since PM could recover a machine to any better operating states, the total number of PM states is $\frac{W_k(W_k-1)}{2}$. Therefore, the set of machine states for machine k can be expressed as $S_k(t) = \{K_1, K_2, W_k, \Gamma_k, W_k^F\}$.

The action space $\mathcal{A}$ is a collection of all possible PM actions given the system state $S(t)$. PM actions refer to all the possible actions to recover the machine conditions. In this work, each action will match a state in PM set of $S(t)$ because PM set denotes the transition results of PM actions. Moreover, for both machines, since they act independently (following the assumptions), there are two independent action sets to control the behavior of each individual machine respectively. Define $\omega_k(S(t))$ as the

decision variable for machine m_k under the system state $S(t)$, then the PM actions can be expressed in the equation as follows:

$$\omega_k(S(t)) = \begin{cases} j, & \text{if PM recovers state } S_k(t) \text{ to } j, \\ 0, & \text{do nothing.} \end{cases} \tag{4.1}$$

Notice that since the system state collects the behaviors of both machines and the buffer level, the actions of a single machine is determined by not only the state of the machine itself, but also the buffer level and the state of the other machine.

Let $(\omega_{i1}, \omega_{i2})$ denote action i for the entire system, which is the combination of actions on two individual machines. The collection of all the actions on the system is the set:

$$\mathcal{A} = \{(\omega_{11}, \omega_{12}), (\omega_{21}, \omega_{22}), \ldots\}.$$

Notice that the number of all the possible actions is finite, since it is the permutation of all the possible actions on both machines, which is also finite. Besides, the number of all the possible actions is also state dependent.

4.1.2 Transition Probabilities Matrix and rewards

The state transition probability matrix, denoted as $\mathbf{B}$, contains all the system state and the corresponding transition probabilities. In order to construct the matrix, a function, $\eta(h, S_1, S_2)$, representing the state number of the system containing two machines and one buffer, is firstly assigned to match the location of system state $\{h, S_1, S_2\}$. To determine the transition of the system state, the transition dynamics for both the machine states and buffer levels should be considered. Note that given the current system state $\mathbf{S(t)}$ at time t, the transition of the buffer level is deterministic. Therefore, the transition rules for buffer level can be determined based on machine states and summarized in Table 4.1. "U" represents that the machine is operating,

$h(t)$	0	h	N	$S_1(t)$,$S_2(t)$	$\phi(h(t), h(t+1))$
$h(t+1)$	1	$h+1$	N	U, F/PM	1
	1	h	N	U, U	
	0	$h-1$	N-1	F/PM, U	
	0	h	N	F/PM, F/PM	
	Other values				0

Table 4.1: Buffer Level Transition under Different Machine State Combinations

"F" means the machine is in failure, and "PM" means that the machine is under maintenance. The first row lists all possible values of the buffer level at time t: 0, N, or a value h in between. Such classification considers possible blockage and starvation when the buffer level is at the boundary. Then, at time $(t+1)$, the transition of buffer level is determined by machine states $S_1(t)$ and $S_2(t)$. For example, in row 2, machine 1 is in operating states and machine 2 is not. Then the buffer level at time $(t+1)$ will be $(h+1)$ if $h(t) < N$ because machine 1 will add a part while machine 2 will not take a part from the buffer. However, if $h(t) = N$, machine 1 will be blocked, thus the buffer level remains N in the next period. Following the similar idea, all the scenarios in Table 2 can be determined. Further, in the last column, a indicating ϕ function is assigned. If the transition from $h(t)$ to $h(t+1)$ exceeds the situation beyond the four situations above the last row, then the ϕ value will be zero.

In terms of the machine states transitions, the two machines can be considered separately according to the independence assumption. Since PM could influence machine state transitions, an indicator function is introduces first for machine m_k to check if PM is performed at time t:

$$I_k(S(t)) = \begin{cases} 1, & \text{if } \omega_k(S(t)) > 0, \text{ i.e., PM is performed,} \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

Generally, when maintenance is performed under the system, the indicating function value will be one, forcing the system state to transfer to the corresponding maintenance state. If no maintenance is needed, then the value is zero and the system degrades in normal path. For a single machine m_k, the transition probabilities can be calculated as follows:

$$\mathbf{Pr}[S_k(t+1) = v | S_k(t) = u] = \begin{cases} p_{u,v}(1 - I_k(S(t))) & u \in K_1 \text{ and } v \in \{u, u+1, u^F\}, \\ p_{u,v} I_k(S(t)), & u \in K_1 \text{ and } v = \Gamma_k, \\ p_{u,v} I_k(S(t)) & u \in \Gamma_k, v = \omega_k(S(t)), \\ 1 & u \in K_2, v^F = u, \\ p_{u,v} & u \in \{W, W^F\}, v \in \{1, W^F\}, \\ 0, & \text{Otherwise.} \end{cases} \tag{4.3}$$

For example, when the machine state u at time t is in the operating state set K_1 ($u \in K_1$), then the transition probability that the machine will be in same state u is $p_{u,u}(1 - I_k(S(t)))$. The indicator function ensures that if PM is performed, then the machine will be sent for PM and will not be retained in the same state. This example explains the first line on the right-hand side of Equation (4). The rest of the equation is the mathematical expression obtained using the rules described in assumptions 6 - 8 in Section 3.

Therefore, the transition matrix $\mathbf{B}$ can be built and calculated as follows:

$$B(\eta(h, S_1, S_2), \eta(h', S_1', S_2')) = \mathbf{Pr}[S_1(t+1) = S_1' | S_1(t) = S_1] \times \mathbf{Pr}[S_2(t+1) = S_2' | S_2(t) = S_2] \times \phi(h, h'). \tag{4.4}$$

For the reward function, the purpose is to find the output of the system under each state. Based on the assumptions given in Section 2, the system produces a part at the end of slot t if and only if the system state $S(t)$ meets with three requirements: 1) m_2 is in any operating state; 2) the buffer is not empty ($h(t) > 0$) to ensure no starvation; 3) No PM is performed in time slot t under action i ($I_{i2}(S(t)) = 0$) to ensure the machine can work normally during the period. Therefore, let $C(S(t))$ denote the reward function, then it can be expressed as follows:

$$C(S(t)) = \begin{cases} 1, & \text{if } S_2(t) \in K_1 \cup W_2, h(t) > 0, \omega_{ik}(S(t)) = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{4.5}$$

which represents that the reward will be 1 for each part produced. If no part is produced, the reward for the state is 0.

4.1.3 Solution Method

The action space includes all the decisions to determine whether a machine will perform PM when it is in an operating state state given the state of the other machine and the current buffer level. Let $0 < \lambda < 1$ be a discount factor, and let $V_t(S_1, S_2, h)$ denote the expected total achievable reward for the period t to the infinite horizon when the system in state (S_1, S_2, h) at time t, then

$$V_t(S_1, S_2, h) = C_1(S_1, S_2, h) + \lambda \sum_{(S_1', S_2', h') \in \mathbf{S}, \mathbf{I}} \mathbf{Pr}(S_1', S_2', h' | S_1, S_2, h, \mathbf{I}) E[V_{t+1}(S_1', S_2', h')]. \tag{4.6}$$

On the right hand side of Equation (4.6), it explicitly describes the two components of the reward: the reward for the current state at time t, and the discounted reward of reaching all the possible system states starting from the next time period, weighted by the corresponding transition probabilities. Then the optimal policies starting from time 0 can be calculated by using the value iteration approach (Puterman (2005)).

Following the discussion in Sections 4.1 and 4.2, a generator $\mathcal{TP}_1$ can be defined to express the processes of PM policy generation for the two machine systems as follows:

$$[I_1, I_2] = \mathcal{TP}_1(\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}, N), \tag{4.7}$$

where I_1, I_2 are obtained through Equations $(1) - (7)$. The input of the generators are the machine state transition matrices $\mathbf{Q_1}, \mathbf{Q_2}$, PM recovery matrices $\mathbf{PM_1}, \mathbf{PM_2}$, and the buffer level N. The output is the PM policies for both machines.

4.1.4 An Illustrative Example

The example considered is a 2M1B system, with the buffer capacity $N = 3$. The state transition matrices and preventative maintenance recovery matrices for both machines are shown in Table 4.2. For the state transition matrix, the rows represent the operating states and the major failure state (W^F). The columns represents the operating states and the last column represents that the system state moves from the state labeled in the row to the corresponding failure states. For example, $\mathbf{Q_1}(\mathbf{1}, \mathbf{4}) = 0.05$ represents that the probability that machine 1 fails from state 1 to the minor failure state 1^F is 0.05. $\mathbf{Q_1}(\mathbf{4}, \mathbf{4}) = 0.9$ represents the probability that the system remains in the major failure state 3^F (transit from 3^F to 3^F) is 0.9.

By solving the MDP problem introduced above, one can obtain the optimal policy. The structure of the optimal PM policy is shown in Table 4.3 for both machines

State Transition Matrix	PM Recovery Matrix

$$\mathbf{Q_1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 3^F \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & FS \end{array} \\ \begin{bmatrix} 0.8 & 0.15 & 0 & 0.05 \\ 0 & 0.7 & 0.2 & 0.1 \\ 0 & 0 & 0.6 & 0.4 \\ 0.1 & 0 & 0 & 0.9 \end{bmatrix} \end{array} \qquad \mathbf{PM_1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0.9 & 0 & 0 \\ 0.75 & 0.8 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{Q_2} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 3^F \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & FS \end{array} \\ \begin{bmatrix} 0.9 & 0.05 & 0 & 0.05 \\ 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0 & 0.7 & 0.3 \\ 0.1 & 0 & 0 & 0.9 \end{bmatrix} \end{array} \qquad \mathbf{PM_2} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0.6 & 0 & 0 \\ 0.6 & 0.75 & 0 \end{bmatrix} \end{array}$$

Table 4.2: Machine State Transition Matrix and Preventative Maintenance Recovery Matrix

under different buffer levels. The rows and columns represent the state of m_1 and m_2 respectively. The cells with bracketed numbers indicates that PM policies are performed under the associated states. For example, $(1, 1)$ means that both machines will perform PM to recover the machine states to 1. If a number is only shown on the first of the paired numbers, with a star (*) on the second, then PM is performed on machine m_1 and no PM is performed on the second machine. Similarly, $(*, 1)$ means that PM is performed only on machine m_2. The states with blank mean that no PM is performed on either machine.

The results indicate that the optimal policy varies under different buffer levels. When the buffer level is low, for example, $h = 0$ or $h = 1$, m_2 tends to perform PM

h=0		M2								
		1	2	3	1F	2F	3F	2,1	3,1	3,2
M1	1		(*,1)	(*,1)						
	2		(*,1)	(*,1)			(1,*)			
	3	(1,*)	(1,*)	(1,1)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)
	1F		(*,1)	(*,1)						
	2F		(*,1)	(*,1)						
	3F		(*,1)	(*,1)						
	2,1		(*,1)	(*,1)						
	3,1		(*,1)	(*,1)						
	3,2		(*,1)	(*,1)						

h=1		M2								
		1	2	3	1F	2F	3F	2,1	3,1	3,2
M1	1		(*,1)	(*,1)						
	2		(*,1)	(*,1)			(1,*)			
	3	(1,*)	(1,1)	(1,1)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)
	1F		(*,1)	(*,1)						
	2F		(*,1)	(*,1)						
	3F		(*,1)	(*,1)						
	2,1		(*,1)	(*,1)						
	3,1		(*,1)	(*,1)						
	3,2		(*,1)	(*,1)						

h=2		M2								
		1	2	3	1F	2F	3F	2,1	3,1	3,2
M1	1			(*,1)						
	2		(*,1)	(1,1)			(1,*)			
	3	(1,*)	(1,1)	(1,1)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)
	1F		(*,1)	(*,1)						
	2F		(*,1)	(*,1)						
	3F		(*,1)	(*,1)						
	2,1		(*,1)	(*,1)						
	3,1		(*,1)	(*,1)						
	3,2		(*,1)	(*,1)						

h=3		M2								
		1	2	3	1F	2F	3F	2,1	3,1	3,2
M1	1			(*,1)						
	2			(*,1)			(1,*)			
	3	(1,*)	(1,*)	(1,1)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)	(1,*)
	1F			(*,1)						
	2F			(*,1)						
	3F			(*,1)						
	2,1			(*,1)						
	3,1			(*,1)						
	3,2			(*,1)						

Table 4.3: The Optimal Policy for the Illustrative Example

when it is in the imperfect operating states. The reason is that at that time, m_2 is more likely to be starved and few parts are produced. Therefore, the policy suggests to perform PM without having heavy influence on the system performance. On the other hand, when the buffer level is high, i.e., $h = 2$ or $h = 3$, m_2 tends to perform less PM when it is in the imperfect operating states. Especially comparing with the case when $h = 2$, m_2 will not perform PM when it is state 2. The higher buffer level "forces" m_2 to work longer to produce more parts.

In addition, the state to which the PM recovers the machine is determined by the PM probabilities. In this illustrative case, specifically, PM is performed under state 3 for both machines. The reason is that the reward of performing PM, based on the MDP algorithm, is more than the reward of possible failure and recovery from state 3. Furthermore, the algorithm compares the recovery path from state 3 to state 2

and the one from state 3 to state 1 and decides to recovery the system operating state to 2. It can be shown that the condition based maintenance policy proposed in this work further considers the benefit of imperfect PM to all the possible operating states and provides the best action.

Notice that when both machines are under unhealthy operating states, i.e., operating state 3, the optimal policy suggests both machines should perform PM at the same time, regardless of the buffer level. The reason is that the influence of inferior machine operating states is more severe than the variation of buffer level such that PM is always needed. From Table 4.2, it can be observed that the recovering probability is much smaller for system failure than adopting PM. For example, for m_2, the repairing probability is 0.1 while performing PM is at least 0.6. Therefore, the average repairing time, using Geometric model, is 10, while the average PM time is 1.67. Given the probability that the machine will fail in the next period is 0.3, the expected non-operating time without PM is 3, which is higher than the average time that PM will take. Similar situations also apply to m_1. Therefore, the policy suggests that both machines should perform PM, even though with no parts produced for a short time, to benefit the system performance from the long run.

4.2 An Aggregation-based Recursive Approach for Longer Lines

4.2.1 Machine and State Aggregation for 2M1B Systems

In this section, using the 2M1B system as the building block and the result can be extended to the longer lines. Specifically, the machine states and buffer information within the first 2M1B system are considered to find the optimal PM policies for the two machine cases, by applying the approach developed in Section 4.1. Then, all the components within the 2M1B system are aggregated into one single virtual

machine. It will further form a 2M1B system with the immediate downstream buffer and machine, and the similar decision model and aggregation approach will be applied. The same method can be repeated until all the machines and buffers are visited. Using this approach can significantly reduce the total number of system states and the computation overhead as well.

To facilitate such an aggregation approach, state aggregation will be applied. The idea of the system state aggregation is that one operating state and one repairing state, denoted as U and F respectively, are used to represent the aggregated performance of the 2M1B block, instead of using the original complicated model. Such an approach is applicable because once the PM policies are obtained, all the state transition paths are fixed and the probabilities are completely known. As a result of state aggregation, a geometric reliability model is essentially applicable to model the 2M1B system with machine condition degradation processes. The status of each machine can be represented using a set $S = \{U, F\}$, where U is the aggregated system operating state and F is the aggregated repairing state. A probability p thereafter will represent the aggregated failure probability and r represent the repairing probability.

Using such aggregation, the 2M1B system can be modeled using geometric reliability models with transition matrix $\mathbf{Q^a}$ expressed as follows:

$$\mathbf{Q^a} = \begin{pmatrix} 1-p & p \\ r & 1-r \end{pmatrix}.$$

An illustration for the procedure can be found in Figure 4.3.

Following the notation in Section 4.1, transition probabilities p, r and the blockage probability for the first machine Bl_1 can be calculated as follows:

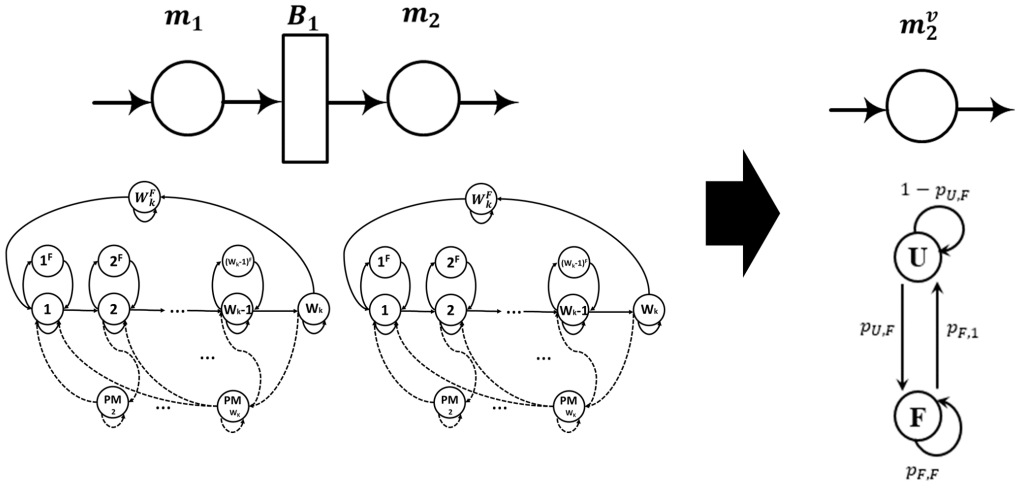


Figure 4.3: Aggregate a 2M1B Model into a Single Geometric Model

$$
\begin{aligned}
p &= \frac{\sum_{u_1}\sum_{v_1} x(v_1) * \mathbf{B}(v_1, u_1)}{\sum_{v_1} x(v_1)}, \\
r &= \frac{\sum_{u_2}\sum_{v_2} x(v_2) * \mathbf{B}(v_2, u_2)}{\sum_{v_2} x(v_2)}, \\
Bl_1 &= \sum_{u_3}\sum_{v_3} x(v_3) * \mathbf{B}(v_3, u_3),
\end{aligned}
\tag{4.8}
$$

$$
\begin{aligned}
&u_1 = \eta(\cdot, \cdot, S_2), S_2 \in K_2 \cup PM \cup W^F; v_1 = \eta(\cdot, \cdot, S_2), S_2 \in K_1 \cup W_k; \\
&u_2 = \eta(\cdot, \cdot, S_2), S_2 \in K_1 \cup W_k; v_2 = \eta(\cdot, \cdot, S_2), S_2 \in K_2 \cup PM \cup W^F; \\
&u_3 = \eta(\cdot, \cdot, S_2), S_2 \in K_2 \cup PM \cup W^F; v_3 = \eta(h, \cdot, S_2), S_2 \in K_1 \cup W_k; h = N,
\end{aligned}
\tag{4.9}
$$

where $\eta(h, S_1, S_2)$ is the state number of the systems when machine m_1 in state S_1, m_2 in state S_2, the buffer level is h and $x(\cdot)$ represents the steady state probability when the two machines system is in the given state, the value of which can be obtained immediately by solving balance equations. The matrix $\mathbf{B}$, defined in Equation (4.4), is the system state transition matrix. The transition probability p in matrix $\mathbf{Q^a}$ is obtained by adding up all the conditional probabilities that the system will fail to

produce a part in the next cycle given that the system is producing a part in the current cycle. Similarly, the transition probability r can be found out by summing up all the probabilities that the system fails in the current cycle and will be repaired in the next cycle. The blockage probability is expressed as the summation of the probabilities that machine 1 will be operating, machine 2 will fail and the buffer is full in the next cycle.

Aggregating a 2M1B system into a single virtual unit enables us to continue finding the PM policies for the adjacent machine in the serial production line. Through moving the block forward, more machines are aggregated into the single virtual units and the transition matrices should be determined. Initially, a block is established on machine m_1, machine m_2 and buffer B_1. Given the policies I_1, I_2, together with the machine state transition matrices $\mathbf{Q_1}, \mathbf{Q_2}$, PM recovery matrices $\mathbf{PM_1}, \mathbf{PM_2}$, and buffer size N_1, the transition matrix $\mathbf{Q_2^a}$ for the virtual unit can be obtained, representing the first block, labeled as m_2^v. The performance of such virtual unit equals to the performance of the 2M1B subsystem operating under the optimal PM policy. Then, this virtual unit is used together with the adjacent buffer and machine m_3 to form a new 2M1B system and use the similar MDP method to find the PM policies for machine m_3. Therefore, for a machine i, $i \geq 3$, given the PM policy denoted as I_i, the transition matrix for the virtual unit m_i^v can be obtained by using parameters $\mathbf{Q_{i-1}^a}$ from the virtual machine m_{i-1}^v; $\mathbf{Q_i}, \mathbf{PM_i}$ from machine m_i; and buffer information N_{i-1}. Besides, the blockage probability Bl_{i-1} for the virtual unit m_{i-1}^v can also be calculated.

Using the procedure defined above, generators $\mathcal{AG}_1$ and $\mathcal{AG}_2$ can be defined for the virtual machine generating process, where the $\mathbf{Q_i^a}$ matrix and blockage probability Bl_{i-1} for the virtual block m_i^v are generated using the results from Equation (4.8):

$$\begin{cases} \mathbf{Q_2^a} = \mathcal{AG}_1(\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}, I_1, I_2, N_1), \\ [\mathbf{Q_i^a}, Bl_{i-1}] = \mathcal{AG}_2(\mathbf{Q_{i-1}^a}, \mathbf{Q_i}, PM_i, I_i, N_{i-1}), i \geq 3. \end{cases} \tag{4.10}$$

For the virtual unit m_i^v, there is only one operating and one failure state. In practice, these states refer to the output of the aggregated subsystem containing $m_1, \ldots, m_{i-1}$. For example, the event that a part is produced and comes out of machine m_{i-1} indicates that the virtual unit m_i^v is in the operating state. Such information can be readily obtained through the sensors monitoring the production flows.

4.2.2 Generating Initial Policies Based on Aggregation

The machine and state aggregation allows us to simplify the system and reduce the computation overhead to generate the PM policies. By applying the approach for the 2M1B system as discussed in Section 4.1, PM policies for all the machines therefore can be determined through repeating the aggregation procedure along the serial line. For the first subsystem with machines m_1, m_2 and buffer B_1, the approach in Section 4.1 is followed to obtain PM policies I_1 and I_2 for machine m_1 and m_2 using the predefined transition matrices and PM recovery matrices $\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}$, and N_1. Furthermore, for subsystems containing a virtual unit, no PM action is considered on the virtual unit since it already incorporates the optimal PM decisions from the aggregation procedure. The optimal PM policy is thus developed only for the original machines within the subsystem. However, since the subsystem with aggregated virtual unit has different states and actions, the approach in Section 4.1 needs to be modified for the 2M1B system with the first machine being a virtual one and the second machine kept the same. Specifically, given the transition matrix $\mathbf{Q_{i-1}^a}$ for the virtual unit m_i^v, $\mathbf{Q_i}, \mathbf{PM_i}$ from machine m_i, and buffer capacity N_{i-1}, the PM

policy I_i for machine i, $i \geq 3$, can be obtained using the similar solution approach as illustrated in Section 4.1.

Following these procedures, together with generator $\mathcal{TP}_1$, another generator $\mathcal{TP}_2$ can be defined for the PM policy generating process using machines and the aggregated virtual unit, where the PM policy I_i for machine m_i are generated using the results from Equation (4.8):

$$\begin{cases} [I_1, I_2] = \mathcal{TP}_1(\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}, N_1) & , \\ I_i = \mathcal{TP}_2(\mathbf{Q^a_{i-1}}, \mathbf{Q_i}, \mathbf{PM_i}, N_{i-1}), i \geq 3 & . \end{cases} \tag{4.11}$$

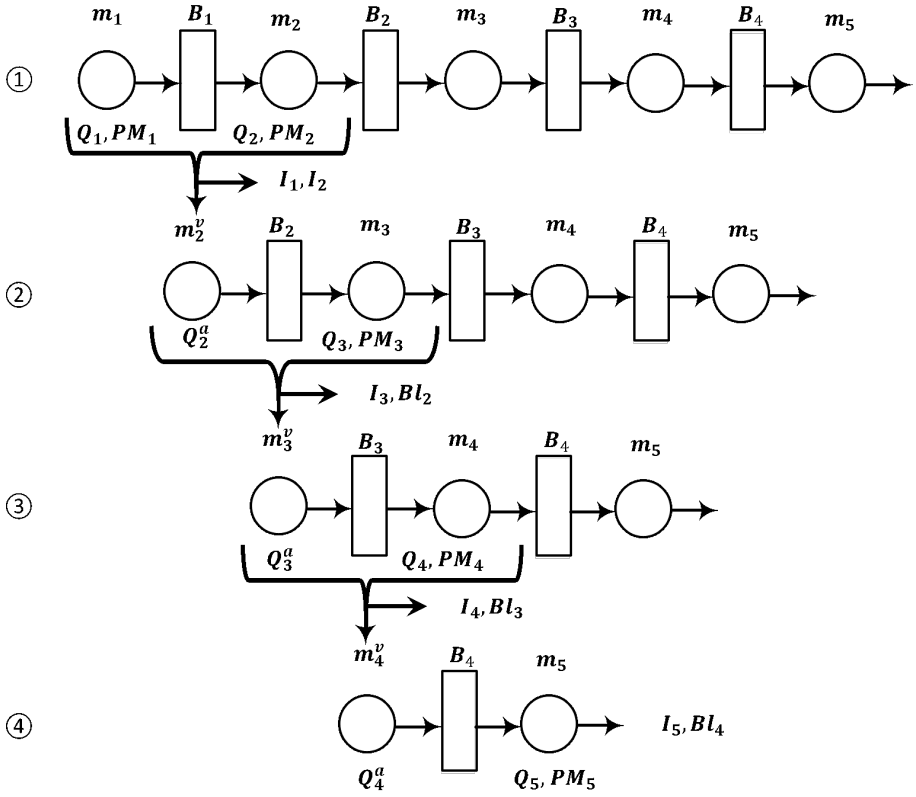


Figure 4.4: Demonstration on Updating PM Policies Using Blockage Probability

An illustrative example explaining how the algorithm generates initial policies for all the machines can be found in Figure 4.4, with the following steps:

- In step ①, policies I_1 and I_2 are generated for machine m_1 and m_2 considering only the two machine system, with $\mathcal{TP}_1(\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}, N_1)$. Then the first building block is merged to obtain the virtual machine m_2^v, with transition matrix $\mathbf{Q_2^a}$, using the generator $\mathcal{AG}_1(\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{PM_1}, \mathbf{PM_2}, I_1, I_2, N_1)$.

- In step ②, using the two machine approach on m_2^v, B_2, and m_3, the PM policy for m_3 is obtained, denoted as $I_3(j)$ with generator $\mathcal{TP}_2(\mathbf{Q_2^a}, \mathbf{Q_3}, \mathbf{PM_3}, N_2)$.

- Then such procedures are repeated, but using the generator $\mathcal{AG}_2$, to obtain the virtual machine m_3^v with transition matrix $\mathbf{Q_3^a}$ and blockage probability Bl_2. The PM policy for m_4 is obtained as denoted in step ③.

- For the last machine, the policy I_5 is found out for machine m_5 in step ④ and the blockage probability for m_4 is calculated as well.

For machines m_1 and m_2, since no state aggregation is applied, the PM policy enumerates all the combinations of machine operating states for both machines. Therefore, for machine m_2, the PM policy is related to the state of machine m_1. For the machines afterwards, the PM policy contains all the state information for the machine in the block alone, together with the state of the virtual unit in the upstream. Thus, for machine m_i, $i \geq 3$, its PM policy is related to the output of the subsystems from machine $m_1, \ldots, m_{i-1}$, not the state of machine m_{i-1} alone. For example, a sample policy for m_5 therefore can be read as "machine m_5 will perform PM back to state 1 if it is under operating state 4, buffer 2 has no inventory, and there is no part coming out from subsystem $m_1, \ldots, m_4$ at the beginning of a time slot". In this way, the approach considers the influence of all the machines ahead, including but not limited to the immediate upstream machine.

4.2.3 Updating Transition Probabilities Considering Buffer Blockage

By aggregating states and machines and applying the results from 2M1B systems, a set of initial PM policies for all the machines in the production line can be obtained. However, since only the forward aggregation is applied, PM policies are generated based on the assumptions that there is no blockage from the downstream system, which is practically not true. As a result, the generated PM policies may not be optimal. Therefore, an updating procedure is developed to refine the initial policy by incorporating the influence from blockage.

The idea is to modify a machine's state transition probabilities using its blockage probability. Note that when a machine is blocked, no parts will be produced; and when the blockage is eliminated, the machine will get back to the same operating state when the blockage occurs. Therefore, the failure probability is revised for state k on machine m_i, $i = 2, \cdots, M-1$ with transition matrix $\mathbf{Q_i}$ using the equations as follows:

$$\begin{aligned} p'_{k,k} &= p_{k,k}(1 - Bl_i), \\ p'_{k,k+1} &= p_{k,k+1}, \\ p'_{k,k^F} &= 1 - p'_{k,k} - p'_{k,k+1}. \end{aligned} \tag{4.12}$$

Referring to Figure 4.2, the probability that a machine is in state k, $p_{k,k}$ is revised into $p'_{k,k}$, through multiplying by the probability that the machine is not blocked. The probability that the machine will be transferred to the next operating state in the next cycle, or $p_{k,k+1}$, remains the same. The adjusted machine failure probability, p'_{k,k^F} is then calculated using $p'_{k,k^F} = 1 - p'_{k,k} - p_{k,k+1}$ to ensure that the probability matrix is valid (the summation of all the transition probabilities out of state k equals to one).

Notice that no adjustment is needed for machine m_1 and machine m_M. The reason is that for machine m_1, the blockage effect is automatically considered when generating PM policies together with machine m_2 in the first building block. For machine m_M, it is never blocked according to the assumptions. In other words, $\mathbf{Q_1^v} = \mathbf{Q_1}$ and $\mathbf{Q_M^v} = \mathbf{Q_M}$. Using such a procedure, a function $\mathcal{BK}$ is defined for the original $\mathbf{Q_i}$ matrix for machine m_i, given the blockage probability Bl_i, where the adjusted transition matrix $\mathbf{Q_i^v}$ is generated using the results from Equation (4.12):

$$\mathbf{Q_i^v} = \mathcal{BK}(\mathbf{Q_i}, Bl_i), i \geq 2. \tag{4.13}$$

4.2.4 A Recursive Approach

Following the above approach, the influence of blockage could be effectively incorporated into the PM decision making in isolated machines. However, these blockage probabilities are initially calculated based on the PM policy with the assumption of no blockage on the second machine. The PM policy may be further changed when the blockage is incorporated, resulting new values of the blockage probability. Therefore, an iterative approach should be pursued to update the PM policy and blockage probability. Eventually the values of the blockage probability and optimal PM policy will converge to a unique solution. The illustration of such an iteration algorithm is shown in Figure 4.5.

Initially, there is no blockage information available and thus the value of the initial blockage is set to be zero, as shown in line 2. The iteration is numbered using j and for all the notations defined in Section $5.1 - 5.3$, the meanings are the same with one additional bracket script added to the right to indicate the iteration number. For example, $Bl_2(j)$ has the same meaning with the notation Bl_2 defined in Section 5.1, but it represents the results for iteration j. For machine m_2, the state transition

Input : $\mathbf{Q_i}, \mathbf{PM_i}$ for all machine m_i, buffer level $\mathbf{N}$, threshold ϵ

Output: PM policies matrix $\mathbf{I}$

% Initial conditions

$\mathbf{Bl(0)} = zeros(1, M)$; $j = 1$; Check = FALSE;

while *Check = FALSE* **do**

  $\mathbf{Q_2^v(j)} = \mathcal{BK}(\mathbf{Q_2}, Bl_2(j-1))$;

  $[I_1(j), I_2(j)] = \mathcal{TP}_1(\mathbf{Q_1}, \mathbf{Q_2^v(j)}, \mathbf{PM_1}, \mathbf{PM_2}, N_1)$;

  $\mathbf{Q_2^a(j)} = \mathcal{AG}_1(\mathbf{Q_1}, \mathbf{Q_2^v(j)}, \mathbf{PM_1}, \mathbf{PM_2}, I_1(j), I_2(j), N_1)$;

  $Bl_M(j) = 0$;

  for $i = 3$ **to** M **do**

    $\mathbf{Q_i^v(j)} = \mathcal{BK}(\mathbf{Q_i}, Bl_i(j-1))$;

    $I_i(j) = \mathcal{TP}_2(\mathbf{Q_{i-1}^a(j)}, \mathbf{Q_i^v(j)}, \mathbf{PM_i}, N_{i-1})$;

    $[\mathbf{Q_i^a(j)}, Bl_{i-1}(j)] = \mathcal{AG}_2(\mathbf{Q_{i-1}^a(j)}, \mathbf{Q_i^v(j)}, \mathbf{PM_i}, \mathbf{I_i(j)}, N_{i-1})$;

  end

  % Check if stopping criteria is met

  Check = TRUE;

  for $i = 2$ **to** M **do**

    if $Bl_i(j) - Bl_i(j-1) > \epsilon$ **then**

      Check = FALSE;

      $j = j + 1$;

      break;

    end

  end

end

Figure 4.5: Algorithms to Generate PM Policies for Serial Production Systems

matrix is updated first using operator $\mathcal{BK}$ in line 4. Then PM policies will be found in line 5, with the adjusted matrix obtained in line 4. Then the building block is aggregated into a single virtual unit in line 6. Then the block moves forward to machine m_3 and other machines onward, following the similar idea as used in the initial block. In the algorithm, line $9-11$ indicate the procedure to generate policies for machine $m_3, \ldots, m_M$. Besides, since the last machine will never be blocked, the blockage probability is manually set to zero for the last machine m_M in line 7. Line $15-21$ discuss the stopping criteria for the algorithm. If for all the machines, the change of the blockage probability between two iterations is smaller than a predetermined threshold, then the algorithm can stop; otherwise, new iterations are needed.

4.2.5 Numerical Experiments

In this subsection, two types of experiments are conducted. One illustrative example is provided first, followed by the general numerical experiments.

A case with six machines is studied and the structure of the optimal policies are obtained using the proposed algorithm. An illustrative picture can be found in Figure 4.6. The sensors, labeled as κ's, are installed to detect if a part is coming out of a machine in real time. All the parameters describing the machine degradation and PM behaviors can be found in Table 4.4 and 4.5 . Using the proposed approach, the PM policy for each machine can be effectively obtained, as illustrated in Table 4.6. The production rate for the system without PM policies is 0.52, comparing to the PM-implemented system, with the production rate of 0.62. Thus the suggested PM policy contributes to an 19% improvement on system production performance.

In terms of the structure of the PM policies, first, for machine m_1 and m_2, the format and the structure is the same as the two machines and one buffer system. Since they are the first two machines, there is no state aggregation and thus the

State Transition Matrix **PM Recovery Matrix**

$$\mathbf{Q_1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 3^F \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & FS \end{array} \\ \begin{bmatrix} 0.32 & 0.66 & 0 & 0.02 \\ 0 & 0.14 & 0.55 & 0.31 \\ 0 & 0 & 0.62 & 0.38 \\ 0.23 & 0 & 0 & 0.77 \end{bmatrix} \end{array} \quad \mathbf{PM_1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0.77 & 0 & 0 \\ 0.58 & 0.69 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{Q_2} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 3^F \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & FS \end{array} \\ \begin{bmatrix} 0.58 & 0.33 & 0 & 0.09 \\ 0 & 0.42 & 0.44 & 0.14 \\ 0 & 0 & 0.81 & 0.19 \\ 0.34 & 0 & 0 & 0.66 \end{bmatrix} \end{array} \quad \mathbf{PM_2} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0.7 & 0 & 0 \\ 0.63 & 0.92 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{Q_3} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 3^F \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & FS \end{array} \\ \begin{bmatrix} 0.27 & 0.53 & 0 & 0.2 \\ 0 & 0.41 & 0.31 & 0.28 \\ 0 & 0 & 0.61 & 0.39 \\ 0.82 & 0 & 0 & 0.18 \end{bmatrix} \end{array} \quad \mathbf{PM_3} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 \\ 0.66 & 0 & 0 \\ 0.66 & 0.92 & 0 \end{bmatrix} \end{array}$$

Table 4.4: Machine State Transition Matrix and Preventative Maintenance Recovery Matrix (Machine $1-3$)

policies are specified to each operating state. In this case, the structure of the PM policies for m_1 and m_2 is similar to the one introduced for the two-machine case in Table 4.3 and the entire policy is not listed out part of due to space limit. For the machines afterwards, the PM policies can only separate all the operating states for the certain machine and the sensor information. For example, the first line PM

of machine m_3 is $(h_2 = 1, S_{2'} = 0, S_3 = 2, I_3(S) = 1)$. This can be interpreted as "Perform PM on machine m_3 and recover machine state to 1 when there is one spare part in buffer h_2, no parting coming from machine m_2, and machine m_3 is in state 2". For the PM policies of machine m_3 to m_6, it can be observed that PM are performed when the machine is inferior operating states, and the buffer level is low. For example, for machine m_3, PM are performed when it is in operating state 2 or 3 and the buffer level is $h_2 = 1$. Furthermore, the algorithm compares the path to conduct the PM, whether recover the machine state from state 3 to 2 or from 3 to state 1, and decides to recover the system operating state to 2 considering the parameters of the aggregated subsystems ahead of the machine. It can be shown that the condition based maintenance policy proposed in this work further considers the benefit of imperfect PM to all the possible operating states and provide the best action in the blocked two machines and one buffer system.

Besides, immediately following each machine, there is a sensor collecting the information whether there is a part produced at the end of a cycle time. The information provided by the sensors can facilitate actualizing the proposed policies in the real production practice. For example, when there is a part coming out of machine 3, sensor κ will notice and send the information to decision makers instantly.

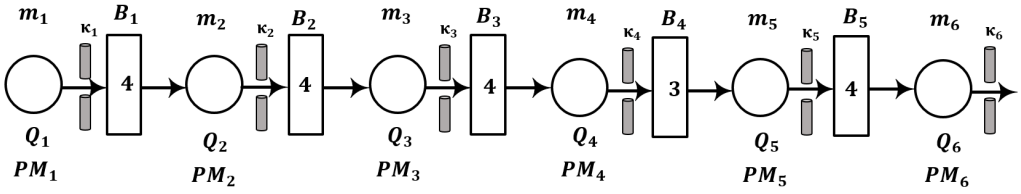


Figure 4.6: An Illustrative Production Lines with Six Machines

Furthermore, the blockage updates for machine 2 to 5 are shown in Figure 4.7. In the first iteration, no blockage is considered and thus the blockage probabilities

for all the machines are zero. For iteration 2, the blockage probability values for all the machines are obtained and used to revise the model and update the initial policy. For example, the blockage probability for machine 2 during iteration 2 is 0.0629. It is further updated based on the new model and PM policy during iteration 3 and remains unchanged in the following iterations. From Figure 4.7, it can be observed that the iterative procedure converges typically within five iterations.

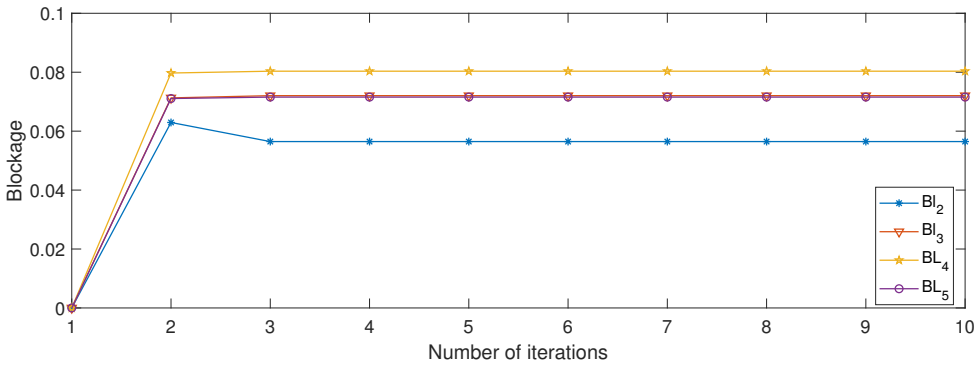


Figure 4.7: Machine Blockage Updates under Each Iteration

To validate the analytical performance results of the optimal policy generated by the MDP, 100 sets of experiments have been conducted. The system parameters are randomly generated using Procedure 5:

Procedure 5

1. *Set up machine numbers in the system* $M \in \{3, \ldots, 10\}$.
2. *Determine buffer level* $B_i \in \{1, \ldots, 6\}$, $i = 1, \ldots, M-1$.
3. *Randomly select operating state numbers for each machine* $W_i \in \{2, \ldots, 5\}$, *for* $i = 1, \ldots, M$.
4. *Generate the state transition matrix* $\mathbf{Q_i}$ *for all machines,* $i = 1, \ldots, M$.

5. *Generate the PM recovery matrix* $\mathbf{PM_i}$ *for all machines,* $i = 1, \ldots, M$.

6. *For each simulation case, using the proposed algorithm to find the optimal PM policies, and apply the policies into the simulation. The simulation is conducted with replication number of 50 and length of 20,000 cycles per replication.*

The performance measurement adopted, denoted as δ_{gen}, is the relative error on production performance of the proposed method comparing with the simulation results:

$$\delta_{gen} = \frac{PR_{sim} - \widehat{PR}}{\widehat{PR}} \times 100\%,$$

where PR_{sim} is the simulation result and $\widehat{PR}$ is the production performance provided by the proposed method. In Table 4.7, the numerical results of 30 cases are provided. It can be found that on average, the deviation of the production performance from the model is within 15% of that from the simulation results. For the cases with large percentage deviation, for example in Case 10, the absolute deviation of the two results is typically small, i.e., within 0.1. These results show that the proposed method is valid.

State Transition Matrix	PM Recovery Matrix

$$\mathbf{Q_4} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 4^F \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & FS \end{array} \\ \begin{bmatrix} 0.03 & 0.94 & 0 & 0 & 0.03 \\ 0 & 0.21 & 0.64 & 0 & 0.15 \\ 0 & 0 & 0.05 & 0.73 & 0.22 \\ 0 & 0 & 0 & 0.66 & 0.34 \\ 0.48 & 0 & 0 & 0 & 0.52 \end{bmatrix} \end{array} \quad \mathbf{PM_4} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.95 & 0 & 0 & 0 \\ 0.98 & 0.99 & 0 & 0 \\ 0.88 & 0.9 & 0.97 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{Q_5} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 4^F \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & FS \end{array} \\ \begin{bmatrix} 0.32 & 0.66 & 0 & 0 & 0.02 \\ 0 & 0.22 & 0.63 & 0 & 0.14 \\ 0 & 0 & 0.32 & 0.5 & 0.18 \\ 0 & 0 & 0 & 0.61 & 0.39 \\ 0.55 & 0 & 0 & 0 & 0.45 \end{bmatrix} \end{array} \quad \mathbf{PM_5} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.95 & 0 & 0 & 0 \\ 0.84 & 0.84 & 0 & 0 \\ 0.63 & 0.73 & 0.92 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{Q_6} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 4^F \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & FS \end{array} \\ \begin{bmatrix} 0.45 & 0.46 & 0 & 0 & 0.09 \\ 0 & 0.74 & 0.13 & 0 & 0.13 \\ 0 & 0 & 0.05 & 0.59 & 0.36 \\ 0 & 0 & 0 & 0.45 & 0.55 \\ 0.36 & 0 & 0 & 0 & 0.64 \end{bmatrix} \end{array} \quad \mathbf{PM_6} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.95 & 0 & 0 & 0 \\ 0.84 & 0.84 & 0 & 0 \\ 0.63 & 0.73 & 0.92 & 0 \end{bmatrix} \end{array}$$

Table 4.5: Machine State Transition Matrix and Preventative Maintenance Recovery Matrix (Machine $4-6$)

	S			$\omega_1(S)$
	h_1	S_1	S_2	
	3	3	1	1
	3	3^F	2	1
m_1	4	3	2	1
Policy	4	3	3	1
	4	3	3^F	1
		...	...	

	S			$\omega_2(S)$
	h_1	S_1	S_2	
	0	3	3	2
	0	2^F	2	1
m_2	0	2^F	3	2
Policy	0	3^F	2	1
	0	3^F	3	2
		...	...	

	S			$\omega_3(S)$
	h_2	$S_{2'}$	S_3	
m_3	1	0	2	1
Policy	1	0	3	2

	S			$\omega_4(S)$
	h_3	$S_{3'}$	S_4	
	0	0	4	1
m_4	1	0	2	1
Policy	1	0	3	1
	1	0	4	1

	S			$\omega_5(S)$
	h_4	$S_{4'}$	S_4	
m_5	1	0	2	1
Policy	1	0	3	1
	1	0	4	1

	S			$\omega_6(S)$
	h_5	$S_{5'}$	S_6	
	0	1	3	1
m_6	0	1	4	2
Policy	1	0	2	1
	1	0	3	1
	1	0	4	2

Table 4.6: Structural Policy for the Illustrative Example

Case	$\widehat{PR}$	PR_{sim}	δ_{gen}	Case	$\widehat{PR}$	PR_{sim}	δ_{gen}	Case	$\widehat{PR}$	PR_{sim}	δ_{gen}
1	0.5356	0.4847	10.51%	11	0.5977	0.5405	10.59%	21	0.6285	0.6833	-8.02%
2	0.5343	0.6206	-13.90%	12	0.5697	0.5560	2.46%	22	0.6488	0.6350	2.17%
3	0.4624	0.5262	-12.13%	13	0.5201	0.4697	10.74%	23	0.6184	0.6642	-6.89%
4	0.5486	0.5520	-0.61%	14	0.6236	0.6340	-1.64%	24	0.5826	0.6018	-3.18%
5	0.4921	0.5687	-13.46%	15	0.5666	0.6160	-8.02%	25	0.6601	0.6361	3.78%
6	0.5924	0.5395	9.81%	16	0.5605	0.6209	-9.73%	26	0.6036	0.5866	2.91%
7	0.6471	0.6032	7.28%	17	0.5128	0.4989	2.79%	27	0.5532	0.6052	-8.59%
8	0.5310	0.6334	-5.00%	18	0.5997	0.6417	-6.55%	28	0.5977	0.5405	10.59%
9	0.4835	0.5423	-10.84%	19	0.5306	0.5467	-2.95%	29	0.5056	0.5800	-12.83%
10	0.3529	0.4368	-19.21%	20	0.5563	0.5517	0.83%	30	0.5412	0.5339	1.38%

Table 4.7: Model Validation for General Serial Production System Cases

4.2.6 Comparison with State-of-the-art

To further illustrate the effectiveness of the proposed method, two methods are compared, which are relevant to the system under consideration . Approach one is to maximize the long term machine efficiency (Lu *et al.* (2015); Fitouhi *et al.* (2017)). Approach two is to find the best maintenance opportunity window, which is determined by the buffer levels (Xia *et al.* (2015); Gu (2016); Lee *et al.* (2013a)).

Approach 1: The idea is to maximize the long term steady state performance through adopting appropriate PM policies. The goal of this approach is to ensure that at any time the machine is working, it is of the highest efficiency. To achieve such a goal with the implementation of PM policies, methods are developed to find the threshold operating state in this approach. Therefore, the machine can operate only on states ahead of the threshold state and will perform PM immediately when reaching the threshold state. In this dissertation, the method by Fitouhi *et al.* (2017) is used for comparison purpose.

Approach 2: The goal is to find the best maintenance opportunity time window by setting up two buffer thresholds, h^{up} and h^{down}. All PM will be performed if and only if the buffer level is above h^{up} or below h^{down}. For example, in a 2M1B system, if the buffer level is larger than h^{up}, then m_2 is unlikely to be starved for the next few time slots, while the probability that m_1 will be blocked will be higher. In this scenario, PM could be applied on machine m_1 if it is currently operating on an imperfect state. Similarly, if the buffer level is below h^{down}, PM can be performed on m_2 and the buffer level will be higher. The goal of such an approach is to maintain a medium size of buffer level and make use of the time to improve the machine status: when m_1 is highly likely to be blocked due to high buffer level and m_2 to be starved due to low buffer level. An analytical approach to determine the optimal buffer threshold

for PM in the serial production systems is developed in Gu (2016), and is used for comparison purpose in this chapter.

Procedure 1 is used to conduct the simulation experiments and obtain the production performance of systems with no PM policy (denotes as PR_{nl}), PM policy with our proposed method (PR_p),PM policy with benchmark one (PR_{b1}), and PM policy with benchmark two (PR_{b2}). The performance measurement adopted, denoted as $\delta_.$, is the relative difference of the production rate between the proposed approach and the existing approach:

$$\delta_. = \frac{PR_p - PR_.}{PR_p} \times 100\% \tag{4.14}$$

where PR_p represents the production rate performance using our proposed method, and $PR_.$ is the production rate with no PM policy or using either of the benchmark approach. Therefore, the measurements is obtained to find the relative difference for the cases with no policy (denotes as δ_{nl}), using benchmark 1 (denoted as δ_{b1}), and using benchmark 2 (denoted as δ_{b2}). Furthermore, 15 cases are listed out for comparison, shown in Table 4.8.

From Table 4.8, it can be found out that the performance of our proposed method is better than approach 1 in most of cases. The improvement typically ranges from 10% to 30%. There are also cases where the improvement margin is close to 0. That is because both the two methods generated similar policies and the difference mainly comes from the simulation error. Further, for the results of benchmark 2, or δ_{b2}, it suggests that the performance of our proposed method outperforms approach 2 in most of the cases as well, with improvement mostly between 2% to 20%. There are a few cases with negative improvement. However, the values are typically below 10%, which are acceptable considering the simulation error.

The main reason our method outperforms the two comparing approaches is that our proposed method systematically considers all the combinations of machine status

Case	Production Rate				Performance Measurement		
	PR_p	PR_{nl}	PR_{b1}	PR_{b2}	δ_{nl}	δ_{b1}	δ_{b2}
1	0.3970	0.3247	0.3762	0.3755	22.27%	5.53%	5.73%
2	0.6767	0.6121	0.6805	0.6511	10.55%	-0.56%	3.93%
3	0.4774	0.3426	0.4053	0.4020	39.35%	17.79%	18.76%
4	0.5882	0.5575	0.5163	0.5756	5.51%	13.93%	2.19%
5	0.4585	0.3859	0.4584	0.4080	18.81%	0.02%	12.38%
6	0.6216	0.5538	0.5487	0.6155	12.24%	13.29%	0.99%
7	0.5409	0.4771	0.5197	0.5163	13.37%	4.08%	4.76%
8	0.5943	0.5398	0.5394	0.5713	10.10%	10.18%	4.03%
9	0.5933	0.5816	0.5306	0.6056	2.01%	11.82%	-2.03%
10	0.4774	0.3426	0.4053	0.4020	39.35%	17.79%	18.76%
11	0.5940	0.5743	0.5642	0.5908	12.50%	28.15%	3.44%
12	0.5075	0.4617	0.4520	0.4855	3.43%	5.28%	0.54%
13	0.4411	0.3447	0.4563	0.4293	27.97%	-3.33%	2.75%
14	0.4876	0.4744	0.4046	0.4894	2.78%	20.51%	-0.37%
15	0.5126	0.4363	0.5114	0.4380	17.49%	0.23%	17.03%

Table 4.8: Comparison Results With State-of-the-art

and buffer level. For approach one, the policy focuses on improving the efficiency of each individual machine, while considering neither the state of other machines nor the buffer level. For approach two, the machines are forced to perform PM when the threshold buffer level is reached, regardless of the machine conditions. These actions could potentially lead to production loss over the long term. The proposed method, on the other hand, further specifies different PM actions by considering the machine states and buffer levels in an integrated way.

4.3 Conclusions

In this work, preventative maintenance policies are developed for the general serial production systems, with finite buffers and machines subject to multi-stage degradation. The solution approach begins with the two machines and one buffer system. Condition-based maintenance decisions are made for such systems, considering machine degradation stages and the buffer level. Moreover, imperfect maintenance is considered in this work such that a machine can be recovered to any better operating state. The optimal maintenance policy in infinite production runs can be obtained using Markov decision models. Furthermore, for general production lines, an approximation method is developed. The two-machine case is used as a building block and machine aggregation approaches are used to generate the PM policies for all the machines. An iterative approach is further designed to incorporate the interactions among all the machines and buffers within the system. The comparisons with the state-of-the-art and numerical experiments demonstrate the effectiveness of the proposed method.

In the future, the research will be extended to more general assembly systems, with complicated structures. Different common types of degradation processes will be utilized to make the method more generic, such as Wiener process. In addition, the real-time signal information will be incorporated to more accurately characterize the machine degradation process, and computationally efficient algorithms will be developed to generate the PM decisions in real time.

Chapter 5

PERFORMANCE EVALUATION OF PRODUCTION SYSTEMS USING REAL-TIME MACHINE DEGRADATION SIGNALS

An illustration of a two-machine-and-one-buffer system is shown in Figure 5.1. The circles and the rectangle are used to represent the machines and the buffer, respectively. The arrows in the graph indicates the flow of working parts within the line. The degradation path for each machine is independent from the other machines. Based on the characteristics of machines, buffers, and their interactions, the assumptions are addressed as follows.

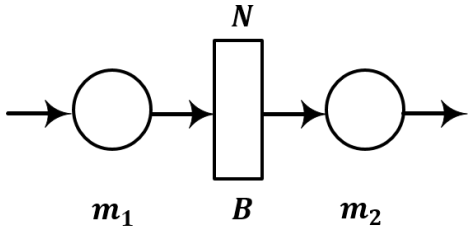


Figure 5.1: A Two-machine-and-one-buffer System

1. The two machines in the system are denoted as m_1 and m_2. The buffer B has finite capacity N.

2. The two machines operate independently. The processing time (cycle time) for one part at machine k is τ_k, $k = 1, 2$. Similarly, the processing speed (or capacity) for machine m_k is c_k, where $c_k = 1/\tau_k$, $k = 1, 2$. It is assumed that the two machines operate in different processing speeds.

3. Machine m_2 is starved if it is up and the buffer is empty at the beginning of a time slot. Machine m_1 is never starved.

4. Machine m_1 is blocked if it is up and the buffer is full at the beginning of a time slot. Machine m_2 is never blocked.

5. For each machine m_k, the real-time degradation signal $z_k(t)$ is observable or can be estimated in real-time to quantify the machine health condition. Furthermore, a parametric form is assumed on the degradation signal $z_k(t) = \eta(\beta_k, \epsilon(t))$. Here, $\eta(\cdot, \cdot)$ is the parameric model of the degradation signal; β_k is the degradation rate; and $\epsilon(t)$ is the noise of the degradation signal.

6. The failure of machine m_k is assumed to occur when the corresponding degradation signal z_k first passes a predefined failure threshold D_k. At each time, the remaining life R_k is defined as the time from now until machine m_k fails.

7. The repairing time for machine m_k follows an exponential distribution with parameter μ_k, $k = 1, 2$.

To evaluate the production performance, system throughput, the number of parts produced per unit of time, is considered. The problem to be studied in this chapter is: Given the real-time machine degradation signals, develop a approach to continuously evaluate and predict the long-term production performance of two-machine-and-one-buffer systems.

5.1 Remaining Life Distribution for Single Machine

5.1.1 Derivation of the Remaining Life Distribution

To model the evolution of the general degradation signal, the degradation signals is built following the Brownian motion model by Hao *et al.* (2017) for each machine k

in the system using the arithmetic Brownian motion at time t as shown in Equation (5.1):

$$dz_k(t) = \beta_k dt + dW_k(t), k = 1, 2, \tag{5.1}$$

where $W_k(t)$ is a Brownian motion with variance $\sigma^2 t$.

Given the real-time degradation signal for each machine k, the remaining life distribution (RLD) of machine k can be estimated based on such model. Using Equation (5.1), the cumulative distribution function (CDF) of the estimated remaining life R_k can be obtained by the Inverse Gaussian (IG) distribution Doksum and Hbyland (1992) as follow:

$$R_k | z_k(t), \beta_k, t \sim IG(t; \mu_k(t), \lambda_k(t)), \tag{5.2}$$

where $IG(t; \cdot, \cdot)$ represents the CDF of an IG distribution, with $\mu_k(t) = (D_k - z_k(t))/\beta_k$ and $\lambda_k(t) = (D_k - z_k(t))^2/\sigma_k^2$ are the mean parameter and the shape parameter, respectively. Here, the RLD only depends on the the most recent degradation signal $z_k(t)$ and the failure coefficient β_k due to the Markov property of the Brownian motion.

It is worth noting that although the conditional distribution of R_k has been derived, there is no explicit expression of the unconditional RLD, since the integral of β_k doesn't yield a close-form solution. For simplicity, the same techniques are adoped in Hao *et al.* (2017), where the Maximum-A-Posteriori point estimator of β_k at time t, denoted by $\hat{\beta}_k(t)$, is used in estimating the mean parameter as $\mu_k(t) = (D_k - z_k(t))/\hat{\beta}_k(t)$. Besides, the mean and variance of the IG distribution can be computed analytically, as $E(R_k) = \mu_k(t), Var(R_k) = \frac{\mu_k(t)^3}{\lambda_k(t)}$. Furthermore, the other moments can also be calculated using numerical methods. For other types of degradation signals, the RLD can be similarly derived.

5.1.2 Approximation of RLD Using Phase-type Distributions

Integrating the real-time degradation signal to the system-level modeling is very challenging due to the non-Markovian property of the machine remaining life distribution. To address this, a new method is proposed in this section by estimating the machine remaining life distributions using phase-type (PH) distributions. PH distributions are defined as the time from any of the transient states to the one absorbing state, which connects with the Markovian models Okamura *et al.* (2011). With such transformation, the system performance modeling based on the Markovian model can be thereafter utilized to estimate the system performance. To do this, a two-phase PH distribution is adapted by introducing two virtual operating states, denoted as state 1 and state 2, and one failure state, denoted as F, for each machine, as shown in Figure 5.2.

When machine m_k fails, it can be recovered either to state 1 or state 2, with probability $p_k\mu_k$ and $(1-p_k)\mu_k$ correspondingly. When the machine is in state 1, it can transfer to state 2 with rate λ_{k1}. When the machine is in state 2, it can transfer to state 2 with rate λ_{k2}.

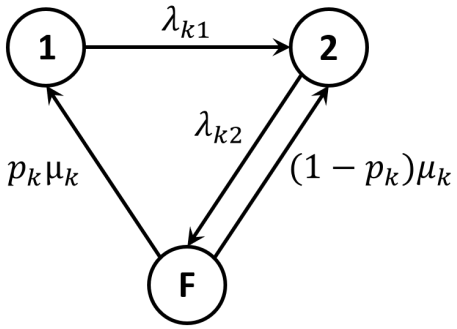


Figure 5.2: Modeling Machine RLD Using PH Distributions

Furthermore, in order to find a PH distribution which has the similar behavior as the IG distribution estimated from the real-time degradation signal, a moment

matching approach is introduced. The idea is to match the three moments of the IG distribution and the PH distribution, thus determining the three unknown parameters in the PH distribution. The symbolic expression is shown in the following proposition.

Proposition 1 *Given an IG distribution $\mathcal{IG}(\alpha, \beta)$, a corresponding PH distribution, which has the exact format as shown in Figure 5.2, can be generated, with parameters as follows:*

- *When* $\frac{\alpha}{\beta} > 1$

$$\begin{aligned} p &= \frac{6\alpha\kappa_1 + \sqrt{\kappa_4} - \kappa_2}{\kappa_2 + \sqrt{\kappa_4}}, \lambda_1 = \frac{\kappa_2 + \sqrt{\kappa_4}}{\kappa_3}, \\ \lambda_2 &= \frac{\kappa_2 - \sqrt{\kappa_4}}{\kappa_3}, \end{aligned} \tag{5.3}$$

  where

$$\begin{aligned} \kappa_1 &= \frac{\alpha^2(\alpha - \beta)}{\beta}, \kappa_2 = \frac{\alpha^3(3\alpha^2 - 2\beta^2)}{\beta^2}, \\ \kappa_3 &= \frac{\alpha^4(3\alpha^2 - \beta^2)}{\beta^2}, \\ \kappa_4 &= \frac{\alpha^6(9\alpha^4 - 18\alpha^3\beta + 6\alpha^2\beta^2 + 6\alpha\beta^3 - 2\beta^4)}{\beta^4} \end{aligned} \tag{5.4}$$

  .

- *When* $\frac{\alpha}{\beta} = 1$

$$p = 0, \lambda_1 = 0, \lambda_2 = \alpha. \tag{5.5}$$

- *When* $\frac{\alpha}{\beta} < 1$

$$p = min(1, \frac{\kappa_5 + \sqrt{\kappa_5^2 + \kappa_5}}{\kappa_6}), \lambda_1 = \lambda_2 = \frac{1 + p}{\alpha}. \tag{5.6}$$

  where

$$\kappa_5 = \frac{\alpha^2(\beta - \alpha)}{\beta}, \kappa_6 = \frac{\alpha^2(\beta + \alpha)}{\beta} \tag{5.7}$$

  .

Proof: See Appendix. ■

Therefore, the equations generated can be solved by the moment matching, thus constructing the PH distribution for a machine's remaining life. It is worth noting that the Proposition 1 can be generalized to other RLDs derived from other types of degradation signals as well.

5.2 Modeling for Two-machine-one-buffer System

5.2.1 Two-machine System Perform Analysis

Using the derived PH distributions for machines' remaining life, a continuous-time and mixed-state Markovian model is developed to estimate the system performance. The state space for the system is defined as the combination of machine states (s_1, s_2), $s_1, s_2 \in \{1, 2, F\}$ and the buffer level h, $h \in [0, N]$, at time t.

To facilitate the derivation, the following notations are introduced:

- $X_{s_1 s_2}(h, t)$: the probability density that machine m_1 in state s_1 and machine m_2 in state s_2, when the buffer occupancy is at h. The buffer level is $h \in (0, N)$, at time t.

- $Y_{s_1 s_2}(N, t)$: the probability density that machine m_1 in state s_1 and machine m_2 in state s_2, when the buffer is full at time t.

- $Y_{s_1 s_2}(0, t)$: the probability density that machine m_1 in state s_1 and machine m_2 in state s_2, when the buffer is zero at time t.

The dynamic transition equations of the probability density functions $X_{s_1 s_2}(h, t)$ can be obtained through the integral equations. Taking $X_{11}(\cdot)$ as an example, the probability density function at time $t + \Delta t$ has a summation format with four components as follows:

- The system stays in the same state from time t to $t + \Delta t$, with probability $X_{11}(h + (c_2 - c_1)\Delta t, t)e^{-(\lambda_{11}+\lambda_{21})\Delta t}$;

- Machine m_2 remains in the operating state 1 and machine m_1 turns up from down state at time $t+\tau$. The probability can be expressed as: $e^{-\lambda_{21}\Delta t}\int_0^{\Delta t} X_{d1}(h+ c_2\Delta t - c_1(\Delta t - \tau), t)p_1\mu_1 e^{-\mu_1\tau}d\tau$.

- Machine m_1 remains in the operating state 1 and machine m_2 turns up from down state at time $t+\tau$. The probability can be expressed as: $e^{-\lambda_{12}\Delta t}\int_0^{\Delta t} X_{1d}(h+ c_2(\Delta t - \tau) - c_1\Delta t, t)p_2\mu_2 e^{-\mu_2\tau}d\tau$.

- A miscellaneous term representing the deviation of estimation with order $O(\Delta t^2)$.

Furthermore, at most one transition could happen within Δt when it is small enough. Therefore,

$$\begin{aligned} &X_{11}(h, t + \Delta t) = X_{11}(h + (c_2 - c_1)\Delta t, t)e^{-(\lambda_{11}+\lambda_{21})\Delta t} \\ &+ e^{-\lambda_{21}\Delta t}\int_0^{\Delta t} X_{d1}(h + c_2\Delta t - c_1(\Delta t - \tau), t)p_1\mu_1 \\ &e^{-\mu_1\tau}d\tau + e^{-\lambda_{12}\Delta t}\int_0^{\Delta t} X_{1d}(h + c_2(\Delta t - \tau) - c_1\Delta t, t) \\ &p_2\mu_2 e^{-\mu_2\tau}d\tau + O(\Delta t^2). \end{aligned} \tag{5.8}$$

Following the similar idea, all the other $X_{s_1s_2}(h, t)$'s can be expressed, which are shown in the appendix.

By simplifying the right hand side of Equation (5.8) using Taylor expansion with an accuracy of $O(\Delta t)$, the simplified expression can be obtained as:

$$\begin{aligned} &\frac{X_{11}(h, t + \Delta t) - X_{11}(h, t)}{\Delta t} = -(\lambda_{11} + \lambda_{21})X_{11}(h, t) \\ &+ (c_2 - c_1)\frac{\partial X_{11}(h, t)}{\partial h} + p_1\mu_1 X_{d1} + p_2\mu_2 X_{1d} + O(\Delta t^2). \end{aligned} \tag{5.9}$$

When t approaches to zero ($\Delta t \to 0$), an differential equation can be obtained by taking the limit of t as:

$$\frac{\partial X_{11}(h,t)}{\partial t} + (c_1 - c_2)\frac{\partial X_{11}(h,t)}{\partial h} = -(\lambda_{11} + \lambda_{21})X_{11}(h,t) + p_1\mu_1 X_{d1} + p_2\mu_2 X_{1d}. \tag{5.10}$$

Furthermore, since the Markov process is irreducible, it can be found out that the limiting distributions with regards to t exist. Introduce the following notation:

$$X_{s_1 s_2}(h) = \lim_{t\to\infty} X_{s_1 s_2}(h,t). \tag{5.11}$$

Then Equation (5.10) can be transformed into the steady state equation as shown below:

$$(c_1 - c_2)\frac{\partial X_{11}(h,t)}{\partial h} = -(\lambda_{11} + \lambda_{21})X_{11}(h,t) + p_1\mu_1 X_{d1} + p_2\mu_2 X_{1d}. \tag{5.12}$$

Following the similar idea, all the rest steady state equations can be obtained for all the remaining system states, as illustrated in the Appendix. Then, a matrix form can be generated to express all the steady state differential equations, as shown in matrix $\mathbf{A_1}$.

$\mathbf{A_1} =$

$$\begin{bmatrix}
-(\lambda_{11}+\lambda_{21}) & 0 & p_2\mu_2 & 0 & 0 & 0 & p_1\mu_1 & 0 & 0 \\
\lambda_{21} & -(\lambda_{11}+\lambda_{22}) & (1-p_2)\mu_2 & 0 & 0 & 0 & 0 & p_1\mu_1 & 0 \\
0 & \lambda_{22} & -(\lambda_{11}+\mu_2) & 0 & 0 & 0 & 0 & 0 & p_1\mu_1 \\
\lambda_{11} & 0 & 0 & -(\lambda_{12}+\lambda_{21}) & 0 & p_2\mu_2 & (1-p_1)\mu_1 & 0 & 0 \\
0 & \lambda_{11} & 0 & \lambda_{21} & -(\lambda_{12}+\lambda_{22}) & (1-p_2)\mu_2 & 0 & (1-p_1)\mu_1 & 0 \\
0 & 0 & \lambda_{11} & 0 & \lambda_{22} & -(\lambda_{12}+\mu_2) & 0 & 0 & (1-p_1)\mu_1 \\
0 & 0 & 0 & \lambda_{12} & 0 & 0 & -(\mu_1+\lambda_{21}) & 0 & p_2\mu_2 \\
0 & 0 & 0 & 0 & \lambda_{12} & 0 & \lambda_{21} & -(\mu_1+\lambda_{22}) & (1-p_2)\mu_2 \\
0 & 0 & 0 & 0 & 0 & \lambda_{12} & 0 & \lambda_{22} & -(\mu_1+\mu_2)
\end{bmatrix}$$

In order to find the steady state distribution $\mathbf{X(h)}$, the following equation has to be solved:

$$\mathbf{c} * \mathbf{X(h)}' = \mathbf{A_1 X(h)}, \tag{5.13}$$

where

$$\mathbf{c} = \begin{bmatrix} c_1 - c_2 & c_1 - c_2 & c_1 & c_1 - c_2 & c_1 - c_2 & c_1 & -c_2 & -c_2 & 0 \end{bmatrix}^T,$$
$$\mathbf{X(h)} = \begin{bmatrix} X_{11}(h) & X_{12}(h) & X_{1d}(h) & X_{21}(h) & X_{22}(h) & X_{2d}(h) & X_{d1}(h) & X_{d2}(h) & X_{dd}(h) \end{bmatrix}^T. \tag{5.14}$$

The operator $(*)$ in Equation (5.13) represents element-wise multiplication of two vectors. The general solution of the differential equation systems has the following format:

$$\mathbf{X(h)} = \sum_{i=1}^{9} k_i \nu_i e^{\gamma_i h}, \tag{5.15}$$

where γ_i's are the eigenvalues, ν_i's are the eigenvectors and k_i's are constants yet to be determined.

The boundary state transition probabilities can be determined following the similarly idea. For boundary probability $Y_{11}(0, t + \Delta t)$, it can be determined as follows:

$$\begin{aligned} Y_{11}(0, t + \Delta t) =& Y_{11}(0, t) e^{-(\lambda_{11} + \lambda_{21})\Delta t} + \int_0^{(c_2 - c_1)\Delta t} X_{11}(h, t) e^{-(\lambda_{11} + \lambda_{21})\Delta t} dh \\ &+ \int_0^{\Delta t} Y_{d1}(0, t) p_1 \mu_1 e^{-\mu_1 \tau} d\tau + O(\Delta t^2) \end{aligned} \tag{5.16}$$

Similar equations for other states are shown in the Appendix. Therefore,these equations can be simplified and put into matrix $\mathbf{A_2}$. The values of all entries in $\mathbf{Y(0)}$ can be obtained by solving the matrix equation:

$$\mathbf{A_2 Y(0)} = \mathbf{c_2} * \mathbf{X(0)}, \tag{5.17}$$

where

$$\begin{aligned} \mathbf{c_2} &= \begin{bmatrix} c_1 - c_2 & c_1 - c_2 & c_1 - c_2 & c_1 - c_2 & -c_2 & -c_2 & 0 \end{bmatrix}^T, \\ \mathbf{Y(0)} &= \begin{bmatrix} Y_{11}(0) & Y_{12}(0) & Y_{21}(0) & Y_{22}(0) & Y_{d1}(0) & Y_{d2}(0) & Y_{dd}(0) \end{bmatrix}^T \\ \mathbf{X(0)} &= \begin{bmatrix} X_{11}(0) & X_{12}(0) & X_{21}(0) & X_{22}(0) & X_{d1}(0) & X_{d2}(0) & X_{dd}(0) \end{bmatrix}^T. \end{aligned} \tag{5.18}$$

$\mathbf{A_2} =$

$$\begin{bmatrix} -(\lambda_{11}+\lambda_{21}) & 0 & 0 & 0 & p_1\mu_1 & 0 & 0 \\ \lambda_{21} & -(\lambda_{11}+\lambda_{22}) & 0 & 0 & 0 & p_1\mu_1 & 0 \\ \lambda_{11} & 0 & -(\lambda_{12}+\lambda_{21}) & 0 & (1-p_1)\mu_1 & 0 & 0 \\ 0 & \lambda_{11} & \lambda_{21} & -(\lambda_{12}+\lambda_{22}) & 0 & (1-p_1)\mu_1 & 0 \\ 0 & 0 & \lambda_{12} & 0 & -(\mu_1+\lambda_{21}) & 0 & p_2\mu_2 \\ 0 & 0 & 0 & \lambda_{12} & \lambda_{21} & -(\mu_1+\lambda_{22}) & (1-p_2)\mu_2 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{22} & -(\mu_1+\mu_2) \end{bmatrix}$$

Similarly, the boundary condition is obtained when the buffer is full.

$$\begin{aligned} Y_{11}(N, t+\Delta t) =& Y_{11}(N,t)e^{-(\lambda_{11}+\lambda_{21})\Delta t} + \int_0^{(c_1-c_2)\Delta t} X_{11}(N-h,t)e^{-(\lambda_{11}+\lambda_{21})\Delta t}dh \\ &+ \int_0^{\Delta t} Y_{1d}(N,t)p_2\mu_2 e^{-\mu_2\tau}d\tau + O(\Delta t^2) \end{aligned} \tag{5.19}$$

Simplifications for other states when the buffer is full are also shown in the Appendix. Similarly, matrix $\mathbf{A_3}$ can be used to represent the parameters in the equations containing $\mathbf{Y(N)}$.

$\mathbf{A_3} =$

$$\begin{bmatrix} -(\lambda_{11}+\lambda_{21}) & 0 & p_2\mu_2 & 0 & 0 & 0 & 0 \\ \lambda_{21} & -(\lambda_{11}+\lambda_{22}) & (1-p_2)\mu_2 & 0 & 0 & 0 & 0 \\ 0 & \lambda_{22} & -(\lambda_{11}+\mu_2) & 0 & 0 & 0 & p_1\mu_1 \\ \lambda_{11} & 0 & 0 & -(\lambda_{12}+\lambda_{21}) & 0 & p_2\mu_2 & 0 \\ 0 & \lambda_{11} & 0 & \lambda_{21} & -(\lambda_{12}+\lambda_{22}) & (1-p_2)\mu_2 & 0 \\ 0 & 0 & \lambda_{11} & 0 & \lambda_{22} & -(\lambda_{12}+\mu_2) & (1-p_1)\mu_1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_{12} & -(\mu_1+\mu_2) \end{bmatrix} \tag{5.20}$$

Therefore, the values of all entries in $\mathbf{Y(N)}$ can be obtained by solving the matrix equation:

$$\mathbf{A_3 Y(N)} = -\mathbf{c_3} * \mathbf{X(N)}, \tag{5.21}$$

where

$$\mathbf{c_3} = [c_1 - c_2 \quad c_1 - c_2 \quad c_1 c_1 - c_2 \quad c_1 - c_2 \quad c_1 \quad 0]^T,$$
$$\mathbf{Y(N)} = \begin{bmatrix} Y_{11}(N) & Y_{12}(N) & Y_{1d}(N) & Y_{21}(N) & Y_{22}(N) & Y_{d2}(N) & Y_{dd}(N) \end{bmatrix}^T,$$
$$\mathbf{X(N)} = \begin{bmatrix} X_{11}(N) & X_{12}(N) & X_{1d}(N) & X_{21}(N) & X_{22}(N) & X_{d2}(N) & X_{dd}(N) \end{bmatrix}^T. \tag{5.22}$$

Notice that the values in $\mathbf{X(0)}$ and $\mathbf{X(N)}$ are the special solutions in Equation (5.15) when $h = 0$ and $h = N$ respectively.

In order to find the values of all k's in Equation (5.15), the particular solution of the Equations in (5.17) and (5.21) will be found. First of all, notice that the summation of all the probabilities should be equal to one.

$$\int_0^N X_{\cdot}(h)dh + Y_{\cdot}(0) + Y_{\cdot}(N) = 1. \tag{5.23}$$

Furthermore, in the long run, when the first machine is faster, and both machines are operating, the buffer level cannot always be zero. On the other hand, if the second machine is faster, the buffer level cannot always be full. Therefore, the following conditions are obtained:

- If $c_1 > c_2$

$$Y_{11}(0) = 0, Y_{12}(0) = 0, Y_{21}(0) = 0, Y_{22}(0) = 0. \tag{5.24}$$

- If $c_1 < c_2$

$$Y_{11}(N) = 0, Y_{12}(N) = 0, Y_{21}(N) = 0, Y_{22}(N) = 0. \tag{5.25}$$

Following the procedures mentioned above, the steady state distribution for all the states can be determined, and the system throughput can be determined as follows:

$$TP = c_2 \sum_{s_1 \in \{1,2\}} \sum_{s_2 \in \{1,2\}} (\int_0^N (X_{s_1 s_2}(h) + X_{d s_2}(h))dh + c_1 Y_{s_1 s_2}(0) + c_2 Y_{s_1 s_2}(N)). \tag{5.26}$$

The system throughput is calculated with the summation of the throughput of given the system state times the probability that the system in that state. In Equation (5.26), the throughput is expressed with three situations: when the buffer is between 0 and N, when the buffer is empty and machine 2 is faster, and when the buffer is full and machine 1 is faster.

5.2.2 Bayesian Updating

After the RLD and the system throughput is estimated, the real-time update is needed on the machine remaining life $R_k, k = 1, 2$ and the system throughput TP with real-time data.

Before the online Bayesian updating, the offline analysis are conducted. To do so, the degradation signals are collected over the different cycles $i = 1, \cdots, n$ for each machine k using the historical information. Then the mean and covariance of the degradation coefficient β_k can be estimated, based on the n time points as sample mean $\kappa_{k,0} = \frac{1}{n}\sum_{i=1}^{n}\beta_{k,i}$ and sample variance $\tau_{k,0}^2 = \frac{1}{n}\sum_{i=1}^{n}(\beta_{k,i} - \kappa_{k,0})^2$. When collecting the real-time sensing data for machine condition $z_k(t)$, the increment of the degradation signal is known to follow the Gaussian distribution as $\delta z_k(t)|\beta_k \sim N(\beta_k \delta t, \sigma_k^2 \delta t)$ by the property of Brownian motion. By the Markov property, it is known that all the non-overlapping incremental of the degradation signals $\delta z_k(1), \cdots, \delta z_k(t)$ are statistical independent. Therefore the likelihood can be derived as $p(\delta z_k(1), \cdots, \delta z_k(t)|\beta_k) = \prod_{i=1}^{t} p(\delta z_k(i)|\beta_k)$. Finally, the posterior distribution can be derived as $\beta_k|\delta z_k(t) \sim N(\kappa_k(t), \tau_k^2(t))$, with the posterior mean $\kappa_k(t) = \frac{\tau_k^2 \sum_{i=1}^{t} \delta z_k(i) + \kappa_{k,0}\sigma_k^2}{\tau_k^2 t + \sigma_k^2}$, and posterior variance $\tau_k^2(t) = \frac{\sigma_k^2 \tau_{k,0}}{\tau_{k,0}^2 t + \sigma_k^2}$, where the $\kappa_{k,0}$ and $\tau_{k,0}^2$ are the prior mean and variance.

Case	m_1					m_2					N	$\widehat{PR}$	PR_{sim}	Δ
	β_1	σ_1	z_1	r_1	c_1	β_2	σ_2	z_2	r_2	c_2				
1	0.47	0.64	4.58	1.08	1.50	0.61	0.24	6.11	1.42	0.98	5	0.4831	0.4881	-1.02%
2	0.42	0.20	4.90	1.33	1.96	0.63	0.84	4.04	1.35	0.88	2	0.5857	0.5850	-0.11%
3	0.29	0.30	6.29	0.76	1.19	1.12	0.60	2.63	1.12	1.86	3	0.9581	0.9520	0.64%
4	0.97	0.60	4.31	1.18	1.60	0.39	0.57	5.01	1.44	0.83	8	0.7556	0.7556	0.00%
5	0.21	0.02	6.54	0.95	1.36	0.28	0.04	6.54	0.66	1.75	3	0.9600	0.9358	-2.52%
6	0.92	0.44	4.12	0.64	1.46	0.77	0.59	3.72	0.32	1.16	8	0.7003	0.6831	2.52%
7	0.73	0.20	3.59	0.36	1.83	0.65	0.39	3.69	0.30	1.53	4	0.8455	0.8346	1.30%
8	0.61	0.78	5.34	1.37	1.39	0.37	0.30	5.98	1.41	0.93	7	0.5950	0.5892	-0.98%
9	0.79	0.39	3.33	1.38	1.63	0.72	0.10	5.04	0.45	1.45	6	0.9378	0.9296	0.88%
10	0.47	0.17	5.88	1.45	0.85	0.58	0.78	5.15	1.28	1.58	6	0.6201	0.6052	-2.40%
11	0.64	0.43	5.04	1.34	1.44	0.44	0.07	6.67	0.27	1.16	7	0.5155	0.5168	-0.26%
12	0.20	0.07	6.71	1.24	1.50	0.42	0.53	6.01	0.85	1.22	3	0.9711	0.9630	0.84%
13	0.60	0.20	5.07	0.80	1.72	0.64	0.98	3.18	0.62	1.64	8	1.2128	1.1935	1.62%
14	0.59	0.27	4.53	0.75	1.48	0.58	0.34	4.43	0.39	1.69	2	1.0055	0.9897	1.59%
15	0.25	0.06	6.31	1.34	1.45	0.62	0.32	4.29	1.31	1.02	3	0.8973	0.8968	0.05%
16	0.30	0.27	6.11	1.37	1.94	0.53	0.03	5.86	0.64	1.67	5	0.9329	0.9225	-1.11%
17	0.21	0.06	7.03	0.25	1.66	0.44	0.12	5.89	0.68	1.42	6	0.7305	0.7173	1.83%
18	0.58	0.05	4.28	0.22	1.56	0.76	0.89	4.05	1.13	1.37	6	0.8328	0.8130	2.44%
19	0.33	0.39	6.93	0.53	1.13	0.51	0.10	5.47	0.30	1.14	8	0.6539	0.6689	-2.25%
20	0.52	0.40	5.00	0.54	1.46	0.54	0.49	4.39	0.76	1.61	7	1.0442	1.0358	0.81%

Table 5.1: Numerical Results

5.2.3 Validation

In order to evaluate the system performance of the model, simulation experiments have been conducted. The system parameters are randomly generated using the following procedure:

Procedure 6

1. *Set up machine capacity* $c_k \in (0.5, 2), k = 1, 2$, *and buffer level* $N \in \{2, \ldots, 8\}$.
2. *Set up machine repair rate* $r_k \in (0.2, 1.8), k = 1, 2$.
3. *Set up the pre-defined degradation threshold* $D_k = 8, k = 1, 2$.
4. *Set up machine degradation level* $z_k \in (0, 5), k = 1, 2$.
5. *Set up degradation rate* $\beta_k \in (0, 1)$ *and noise standard deviation* $\sigma_k \in (0, 1)$, $k = 1, 2$.

All the simulation cases are conducted with 50 replications and 20,000 simulation length. The performance measurement adopted is the relative production performance difference between the analytical model and the simulation experiments:

$$\Delta = \frac{\widehat{PR} - PR_{sim}}{PR_{sim}} \times 100\%,$$

where PR_{sim} is the simulation results, and $\widehat{PR}$ is the analytical solution to the model. Numerical examples are shown in Table 5.1.

Furthermore, 500 simulation experiments are conducted to test the performance of the analytical comparing to the simulation results. For most of the cases, the Δ-measure is below 1%. For the cases with production rate under 0.7, the average relative difference is below 5%. There are extreme rare cases (under 1% of the total experiments) that the Δ-measure is over 10%. They occur when the degradation

level of a machine is very close to the pre-defined threshold. Under these scenarios, the throughput is very low, i.e., less than 0.2. However, for these cases, the absolute difference between the simulation and the analytical results is within 0.03. These results can show that our method can effectively evaluate the system performance.

5.3 Case Study

To demonstrate the effectiveness of the method, a case study is provided to evaluate a two-machine-one-buffer system's performance with real machine degradation signals. To ensure the confidentiality of the data, all the data and parameters introduced below have been modified and are used for illustration purpose only.

The dataset used for the real-time degradation signals includes 21 bearing samples. These degradation signals are calculated from the original vibration signals, captured by an accelerometer, to track the evolution of the vibration level with respect to time Gebraeel *et al.* (2005). A failure threshold of the degradation signals is set according to the industrial standard Blake and Mitchell (1972). Even though the experiments are conducted on a set of identical thrust ball bearings in an accelerated testing, the time-to-failure for all samples are quite different, ranging from 100 cycles to 300 cycles, with sampling interval of 2 minutes per cycle. The differences are mainly due to the sample-to-sample variation and the environmental uncertainty.

The remaining life prediction can be estimated and updated online using the real-time degradation signal. Figure 5.3 is an illustration of the predicted remaining life distribution, with the range of the estimated lige distribution with different level of observed degradation signals. From Figure 5.3, it can be observed that initially, the remaining life distribution has large variance. After observing more degradation signals, the updated remaining life presents much smaller variance, which shows the power of observing more real-time degradation signals.

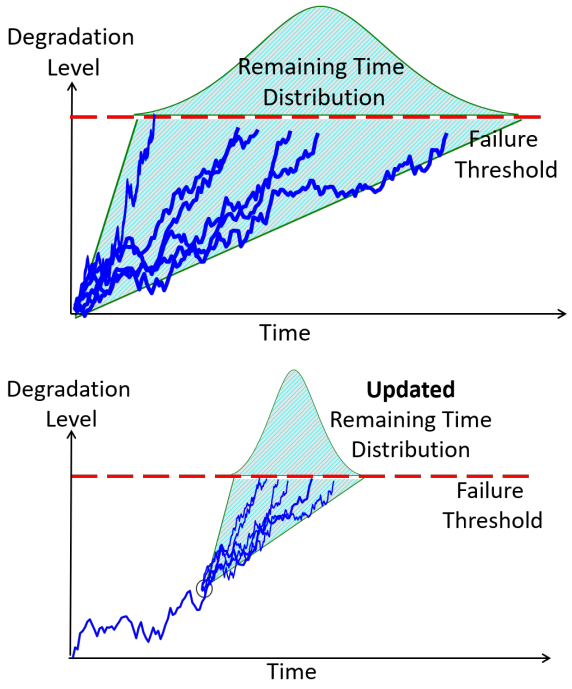


Figure 5.3: Degradation Signals and Predicted Remaining Life Prediction

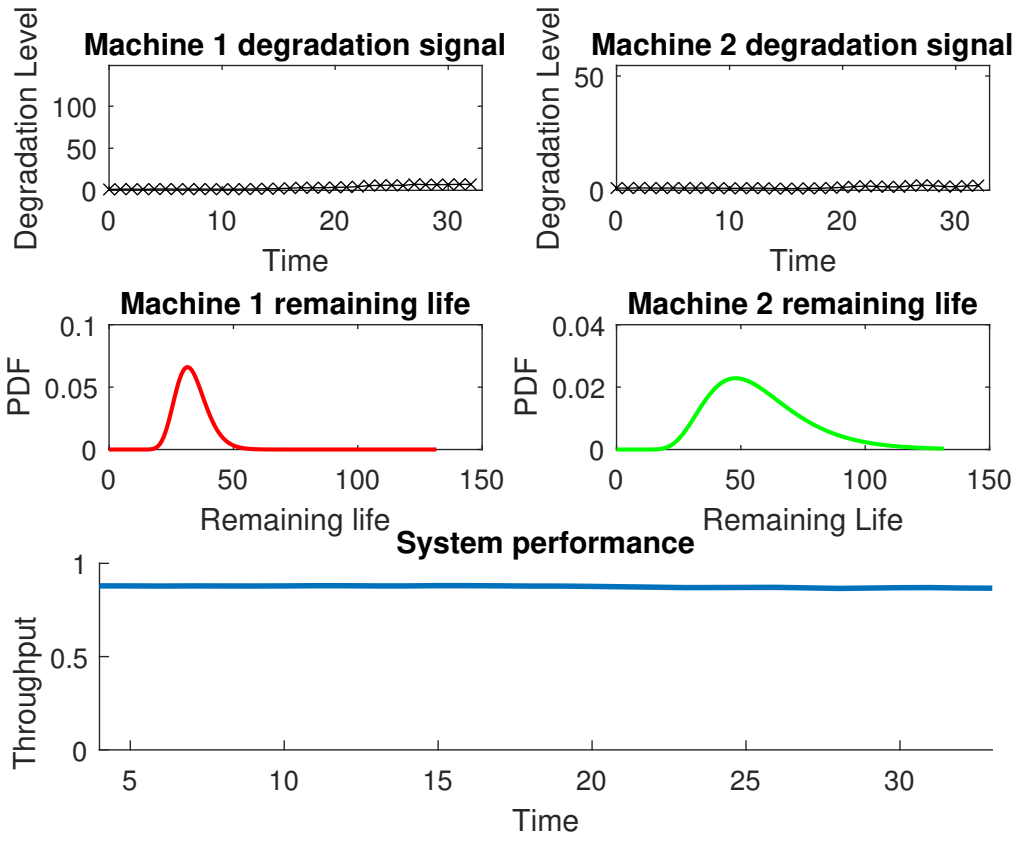
Finally, the system throughput at different stages of machine health status is discussed. Figure 5.4 shows the system status at initial stages (e.g. the system operates less than 50 cycles), where both machines are in good operating states, with low degradation level. The average of remaining time for both machines are large, according to the RLD. Therefore, the system throughput is high. Theoretically, over the long run, the throughput cannot be larger than the minimum capacity in the system, which is known as the bottleneck. In this case, the minimum capacity is 0.9 for machine m_2. Since the machine is less likely to fail in short time, the estimated system throughput is very close to the theoretical upper bound 0.9. When machine 2 approaches to the threshold (e.g. shown in Figure 5.5 at $t = 95$), the estimated system performance drops significantly. Since the single machine degradation process is not monotone, it can be observed that throughput ramps up when the status of a machine recovers in a short period of time. For example, between time 90 and 100 as

shown in Figure 5.5, there is a decrease on the degradation level of machine 1. The system throughput, similarly, shows an increase between time 90 and 95 due to the recovery of machine status. However, when either the machine fails, over the long run, the estimated production rate is zero, even though for short time, there might be output due to the remaining parts in the buffer. The throughput will be zero when the failed machine is under repair and then ramps up when both machines are under operations.

5.4 Conclusions

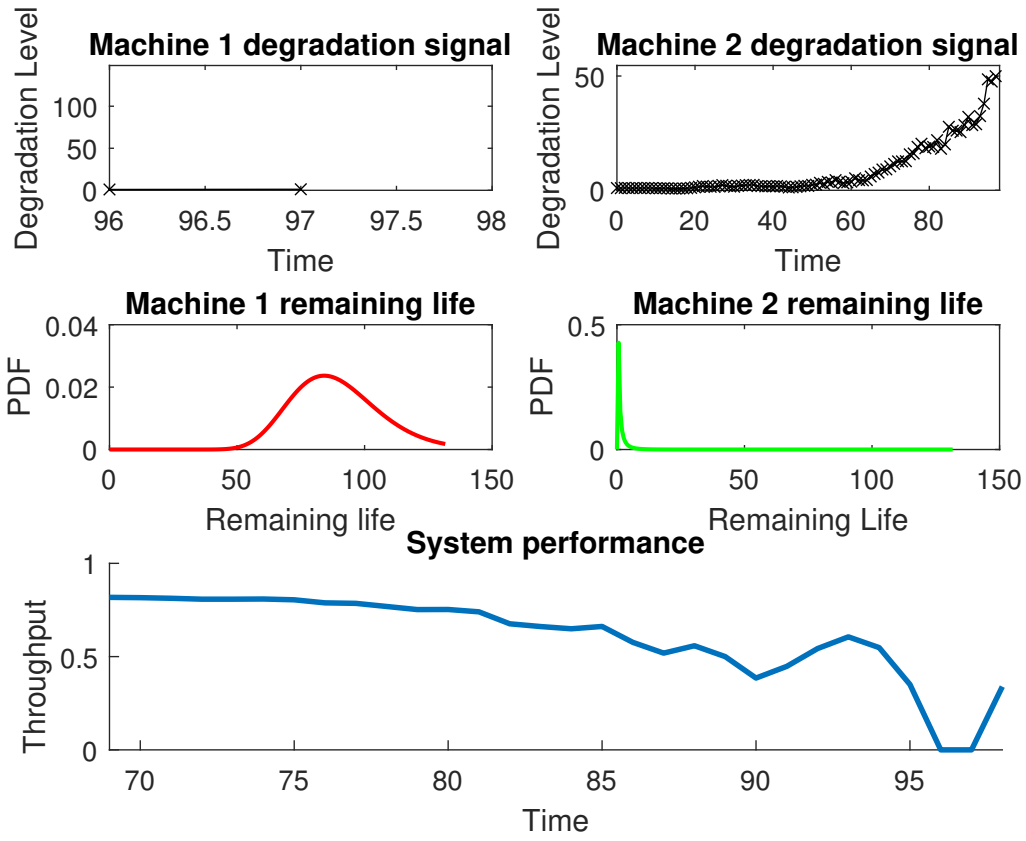
In this chapter, a novel analytical model is developed to evaluate the system performance of a two-machine-and-one-buffer line given real-time machine degradation signals. Specifically, PH distributions are generated to mimic the RLD of each machine and a continuous time Markovian model is formulated to estimate the system throughput. A case study is included to demonstrate the effectiveness of the proposed method. Future work can be dedicated to extending such a model to more complicated manufacturing systems, such as longer lines or assembly systems. Furthermore, the developed approach can be easily generalized to other types of degradation signals, modeled by stochastic processes other than Brownian motion, to enhance the model's efficacy in a variety of practical systems.

In practice, the method can be applied for system performance monitor, projection, and control. Practitioners can obtain the real-time evaluations and predictions of the system production performance. They can have better understanding on individual machines, such as degradation level and the remaining useful life, as further explore the possible impact on overall system performances. These analytics can provide guidelines and contribute to the other operational activities, such as production scheduling and maintenance.



The degradation signal plots record the degradation level read from the signals. Machine remaining life plots depict probability density function of the distribution given the current degradation level estimated using Equation (2). The system performance chart plots the throughput obtained using the method in Section V.

Figure 5.4: System Performance Evaluation When Two Machines in Good Operating State



The degradation signal plots record the degradation level read from the signals. Machine remaining life plots depict probability density function of the distribution given the current degradation level estimated using Equation (2). The system performance chart plots the throughput obtained using the method in Section V.

Figure 5.5: System Performance Evaluation When Machine m_2 in Inferior Operating State

In the future, the work can be extended in the following possible directions. First, in terms of the real-time degradation signals, the data from multiple sensors can be implemented into the model, which can provide additional information and therefore possibly improve the quality of machine condition monitoring. Second, for the system structure, the work can be extended to more complicated while common system structures, such as the serial production lines, assembly lines and production networks. Besides, the current research problem can be further extended to obtained system improvement and control policies. Indeed, to achieve these goals, a comprehensive understanding on system structures, especially in complex manufacturing systems, together with utilizing the data from multiple source and in multiple attributes is needed.

Chapter 6

OPTIMIZING MULTIPLE CONDITION POLICY BASED ON REAL-TIME DEGRADATION SIGNALS VIA MODEL-BASED REINFORCEMENT LEARNING

6.1 Introduction

The degradation process of a unit is directly influenced by different operating conditions. Controllable factors such as the level of usage, speed, force, load, operating time, etc., have been examined to tie closely to the degradation process a machine. For example, in a battery pack, lighter workload usually results in slower increase in the temperature in the battery cells, resulting in longer time in good using conditions and less frequent (preventative) maintenance. On the other hand, the lighter workload can also result in less capacity, which triggers the long time production performance, and can make existing replacement policies cost-ineffective. The wind turbine gearboxes and generators deteriorate faster at higher speeds, conveyor belts fail more often when used at higher rotational speeds and cutting tools wear faster at higher speeds. Furthermore, the optimal conditions should also be determined by the current machine states, which can be inferred from the real-time degradation signals.

In many engineering systems, proper predictive maintenance and operational control should be jointly considered to increase a unit's effectiveness and reduce the time and cost of repairing. Typically for such systems, multiple sensors are normally used to monitor performance, which create difficulties for system state identification. In this chapter, a systematic decision making framework is proposed to improve the system performance in manufacturing practice considering the real-time degradation

signals generated by multiple sensors. Specifically, we propose a partially observed Markov decision process (POMDP) model to generate the optimal capacity and PM policies, given the fact that the observation on the system state is imperfect. To the best of the knowledge, this is the first work to provide a systematic approach that focuses on jointly controlling the operating conditions and preventative maintenance utilizing the real-time machine deterioration signals by incorporating the degradation constraint and non-observable states.

6.2 Methodology Development

6.2.1 Problem Formulation

In this chapter, we would like to first introduce our problem definition and assumptions in Section 6.2.2. We would like to discuss then how to use POMDP to decide the optimal operation condition in Section 6.2.3, by assuming the system dynamics (i.e., the degradation process is given). Finally, we would like to discuss how to estimate unknown system dynamics under multiple operation conditions.

6.2.2 Problem Definition and Assumptions

In this chapter, we build up a discrete model to depict the operation and degradation process of the machine. The time epochs are denoted as $n = 1, 2, \ldots, T$. These time intervals within two time epochs are set as the machine cycle time, which is assumed to be the time that is required to finish processing the assigned units of product. Meanwhile, all the decision, i.e., the capacity adjustment and maintenance options, are made and execute only at the beginning of each time epoch. Here is the definition and assumptions of our studied system.

- Machine degradation process: we assume the state $\mathbf{X} = \{1, 2, \ldots, K, F\}$ with K operating states and one failure states (denoted by F). The transition matrix X between the states are assumed to be static and unknown. For the operating states, they fall into the increasing order that state 1 represents best operating state ("brand new") and state K is the worst operating state before the system fails. Furthermore, state F denotes the failure state, which is the last and non-operating state in the degradation process. The transitions of machine states in the degradation process is assumed to occur at the beginning of a time epoch. We assume that the condition of the machine is retaining or decreasing from time to time. For the machine at an operating state k at time t, it can retain at state k or transfer to any inferior state at time $t + 1$. In other words, no condition reverse is allowed in the machine degradation process. The machine degradation process is expressed with the solid lines as shown in Figure 6.1. It is worth noting that we use discrete state representation of the degradation process compared to the continuous state representation. One main reason is that the discrete state is good to model the non-continuous degradation process with discrete physical state, which appears in the study of many systems (Gu, 2016).

- Control actions 1: Maintenance decision. When the machine is failed, the operation must be stopped, and the maintenance must be performed, replacing the failed component and recovering the machine conditions thoroughly (usually known as "corrective maintenance"). We assume that such replacement and

repair can be finished within one production cycle, and the machine condition will be recovered to the best state (state 1). Apart from the passive corrective maintenance, the decision maker has the option choose to conduct preventative maintenance (PM), when the machine is in any of the operating states $(1, \ldots, K)$ at the beginning of a time slot. If PM is selected, then the machine is stopped immediately and the the degrading component in the machine will be replaced. The selection of PM will lead to the early retirement and discard of the component. We also assume that the replacement is conducted within one cycle. The newly installed component is "brand new", which recovers the machine condition to the best, or state 1. The transitions of machine states with PM conducted is expressed with the dashed lines as shown in Figure 6.1. We further assume that preventative maintenance will incur a small cost and machine break down will incur a large cost. More definition will be discussed in Section 6.2.3.

- Control actions 2: Capacity adjustment. The machine is also assumed able to operate under different conditions throughout the lifetime, i.e., with different operating capacities, and the operator can switch the capacity when needed at the beginning of a production cycle. Let $\mathbf{C} = \{c_1, \ldots, c_m\}$ denote the set of all speed options for the machine. We assume that the speed adjustment options are finite, i.e., the different levels on switchers or gears. Correspondingly, the degradation process under each condition varies. However, we assume that the degradation process can be represented using the same set of states $\mathbf{S}$ and may follow different state transition probabilities for different capacities. We denote $\mathbf{A^c}$ as the state transition probability matrix when the machine is under condition c.

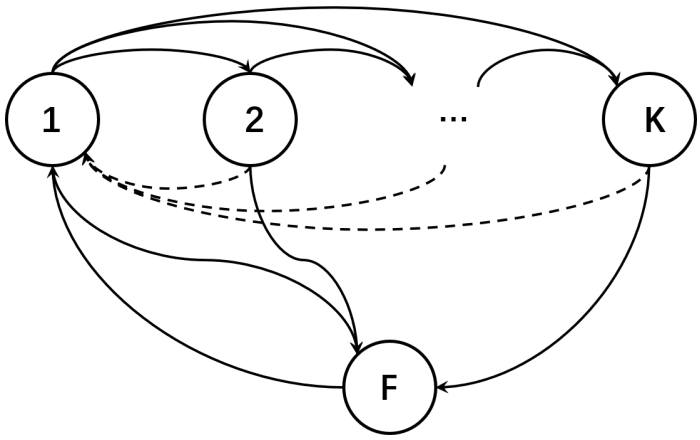


Figure 6.1: The Original System State Transitions

The overall solution scheme is summarized as shown in Figure 6.2. For the offline training, as shown on the left side, historical data sets are collected to abstract the features and thereafter used to obtain the parameters for the HMM model and the POMDP model, which will ultimately return the vectors for decision making given different observations. When the model is conducted for real-time control, the signal is abstracted into the features, which will be classified as the belief vectors on the discrete observations. The belief vector, together with the alpha vectors generated from the training stage, will be used to determine the ultimate control action at the exact time.

6.2.3 A Partially Observed Markov Decision Process Approach to Optimize Multiple Operating Conditions

In practice, the states of the system are non-observable and thus cannot be evaluated directly. Instead, it can be estimated from the signals which monitor the conditions of the machines in real-time. To model and solve the problem, we develop a Partial Observable Markov Decision Process (POMDP) model to solve the optimize the operating condition problem. We will first start with the definition and how to

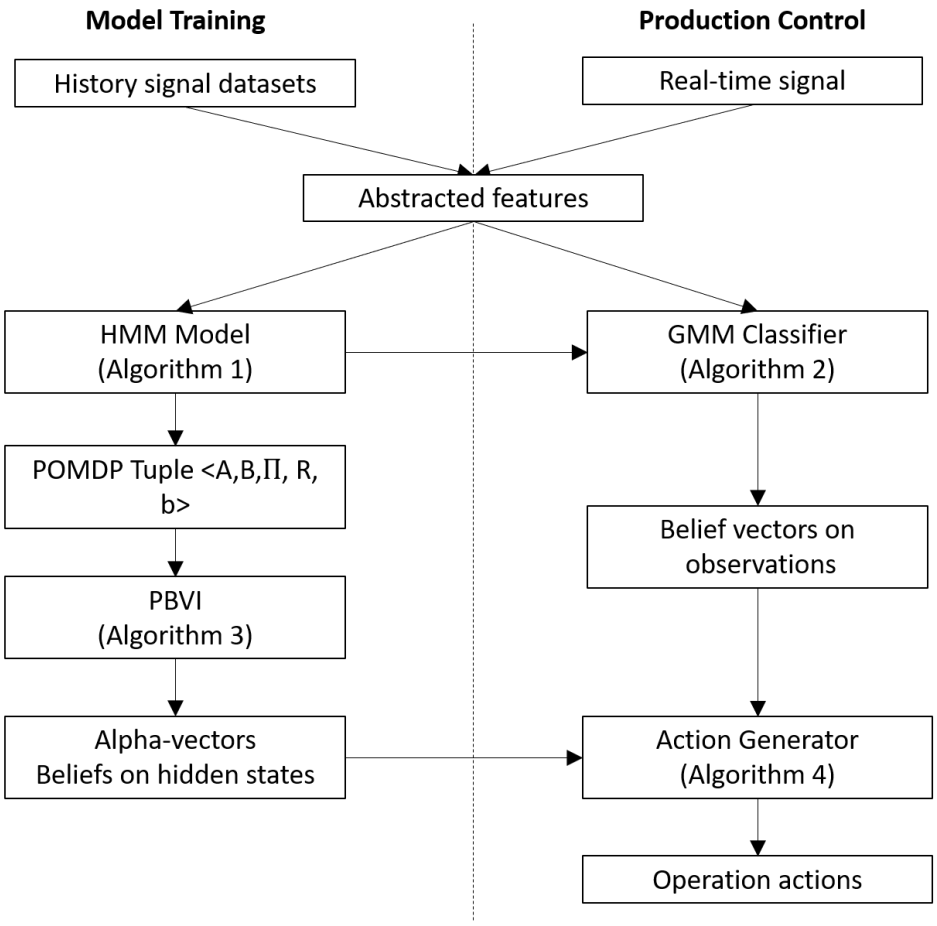


Figure 6.2: The Structure of the Algorithms

map our decision optimization problem into the POMDP utilizing the observations from the degradation signals.

Model the multiple operating conditions control problem using POMDP

A POMDP is a generalization of a Markov decision process (MDP), which models an agent decision process in which the system dynamic is modeled by an MDP, but the agent cannot directly observe the underlying state space. A discrete time Markov model can be built based on the problem descriptions. The state space of the system contains all the states in the machine. In literature POMDP is a 7-tuple

$(S, A, X, R, \Omega, O, \gamma)$, where

- S is a set of states, the non-observable health state of the machine.
- A is a set of actions, which refers to the change of operating conditions or maintenance decision.
- X is a set of conditional transition probabilities between states, which models the dynamics of the degradation process.
- $R : S \times A \to \mathbb{R}$ is the reward function, which refers to the maintenance cost.
- Ω is a set of observations, which is corresponding to the degradation signals observed by single or multiple sensors.
- O is a set of conditional observation probabilities, which models the data generative process given the observation and state.
- $\gamma \in [0, 1]$ is the discount factor, which models how the reward function degrades over time.

The performance of the machine denoted as $r(S, a)$, is condition-base cost metric determined by the machine condition S and the action a. When the machine is under operating conditions ($S \in \mathbf{S}$), the production reward evaluated by the capacity that the machine posits. When the machine fails, the corrective maintenance cost is occurred, which is denoted as c_d. For the preventative maintenance, the cost is uniform for all the states and is expressed as c_p. Then the system reward can be expressed using the equation as follows:

$$r(S,a) = \begin{cases} c_a, & \text{if } S \in \{1,\ldots,K\}, \\ c_d, & \text{if } S \in F, \\ c_p, & \text{if } S \in PM, \\ 0, & \text{otherwise,} \end{cases} \tag{6.1}$$

Furthermore, we have the notations for the observations and transition matrices, which are specified under different operating conditions:

- Let z_t represent the signal observations collected in the period $[t-1,t]$. Introduce T_i, the life time of unit i. Furthermore, let $\mathbf{O} = [o_1(z_t), o_2(z_t), \ldots, o_{|O|}(z_t)]$ denote the vector of features, which are transformed from the observing segments of the signals during time period $t-1$ to t.

- We denote $S_{i,t}^c \in \mathbf{X}$ as the system state at time t for unit $i \in \{1,\cdots,n\}$ under condition $c \in \{1,\cdots,M\}$.

- A transition probability matrix is defined as $\mathbf{X^c} = \{p_{kk'}^c\}$, in which $p_{kk'}^c$ is the transition probability that the system state is changed from state k to state k' under condition c, $k,k' \in \{1,\ldots,K\}$ at time t,

$$p_{kk'}^c = P(S_{i,t+1}^c = k | S_{i,t}^c = k'), 1 \le k,k' \le K. \tag{6.2}$$

- An emission probability is defined on multi-variate response $\mathbf{O}_{i,t}^c$ by two sets of matrix $\mathbf{B} = \{\boldsymbol{\mu}_k^c, \boldsymbol{\Sigma}_k^c\}$, in which $\boldsymbol{\mu}_k^c$ is the mean value of multiple sensors in state k under condition c, and $\boldsymbol{\Sigma}_k^c$ denote the covariance matrix of multiple sensing measurement. In another word,

$$\begin{gathered} (\mathbf{O}_{i,t}^c | S_{i,t}^c = k) \sim N(\boldsymbol{\mu}_k^c, \boldsymbol{\Sigma}_k^c), \\ where\ i = 1,\cdots,n,\ k = 1,\cdots,K,\ c = 1,\cdots,M. \end{gathered} \tag{6.3}$$

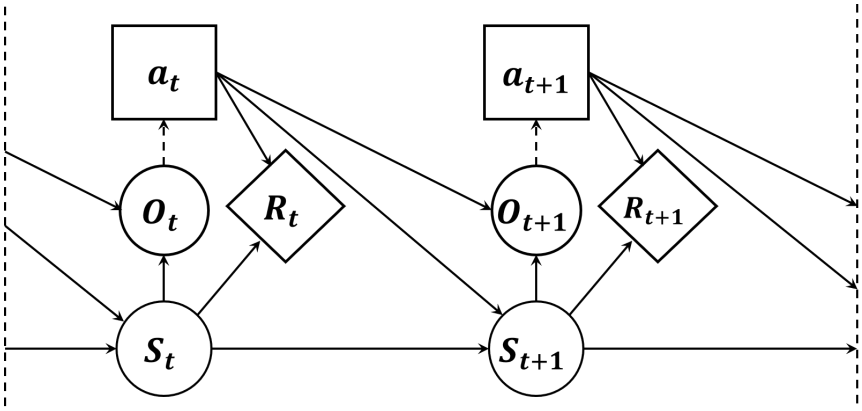


Figure 6.3: The State Transition Diagram of POMDP

- An initial state vector $\mathbf{\Pi_i^c} = \{\pi_{i,k}^c\}$ is defined to express the distribution of the system states for unit i at state k at $t = 1$ under condition c.

6.2.4 Parameter Estimation with Hidden Markov Model

POMDP can be treated as HMM with controllable action. In the training phase, we assume that all control actions are given (i.e., the system have recorded which action to use). However, the dynamics under each condition is unknown. In this subsection, we will study how to estimate the parameters in transition matrix T. It is worth noting that this problem is similar to estimate the transition matrix and emission matrix in the hidden Markov Model (HMM). In Section 6.2.4, we will discuss how to extend HMM into the semi-supervised settings with multi-channel signals and multiple operation conditions with the left-to-right degradation dynamic constraint in Section 6.2.4. Then we will discuss the parameter estimation of the proposed model in Section 6.2.4.

Semi-supervised Left-to-right Constrained Multiple Condition HMM for Prognostics Modeling

HMM is a probabilistic model which describes the transitions of a finite number of states over time. These states characterize two stochastic processes: the hidden state transition in discrete time and an observed process. In addition, three sets of probability distributions are utilized to characterize the system dynamics: the initial probabilities for all the hidden states; the transition probabilities between two hidden states and the emission probabilities of an observation from a hidden state. The elements of an HMM are defined as follows: With the descriptions shown above, for the simplicity and clarification, the full HMM model is given by $\Lambda = \{\mathbf{Q}, \mathbf{X}, \mathbf{B}, \mathbf{\Pi}\}$.

To model the system degradation, we assume that the system can only degrade to worse states over time, which means that the transition is only one-directional. Therefore, we assume that the transition matrix is left-to-right only. Finally, we assume that the failure state is the only absorbing state in the system.

Commonly, EM algorithm is used for finding the maximum likelihood estimate of the parameters of a hidden Markov model given a set of observed feature vectors. However, the traditional EM algorithm is not able to provide the transition matrix with left-to-right constraint. Furthermore, the naive implementation of constraint such as mask out the transition matrix does not lead to robust estimation of the transition matrix and emission parameters. Here we propose to modify the EM algorithm of the HMM learning based on the left-to-right constraint. Due to the fact that the signal is mostly monotone, in each iteration, we rank the posterior mean $\boldsymbol{\mu_1} < \cdots < \boldsymbol{\mu}_k$, where k is the number of state.

Estimation of the Constrained Hidden Markov Model

In this section, we will introduce the notation and framework of the HMM. The random variable $\mathbf{O}_{i,t}^c$ denotes the ith signal observation at time t under condition c and $S_{i,t}^c$ is the hidden state which denotes the health condition for the systems under consideration. The following is the joint likelihood of the problem.

$$\begin{aligned} P(\{\mathbf{O}_{i,t}^c\}_{i,t}^c) &= \sum_{c=1}^{M}\sum_{i=1}^{N}\sum_{s_1,\cdots,s_T} P(\{\mathbf{O}_{i,t}^c\}_{t=1}^{T_i}, \{S_{i,t}^c\}_{t=1}^{T_i}) \\ &= \sum_{c=1}^{M}\sum_{i=1}^{N}\sum_{s_1,\cdots,s_{T_i}} P(S_{i,1}^c)\prod_{t=2}^{T_i} P(S_{i,t}^c|S_{i,t-1}^c)\prod_{t=1}^{T} P(\mathbf{O}_{i,t}^c|S_{i,t}^c) \end{aligned} \tag{6.4}$$

However, the joint likelihood is not tractable. The inference can be broken down into the forward-backward algorithm and the Baum-Welch Algorithm.

The goal of the forward-backward algorithm is to decode the system state given the degradation signals in (6.5). The step of Forward-backward algorithm of the proposed method is the same with ordinary HMM. For more details, please see Rabiner (1989).

$$\gamma_t^k = P(S_t = k|\{\mathbf{O}_t\}_{t=1}^{T_i}) \tag{6.5}$$

However, the original Baum-Welch algorithm cannot be used for the semi-supervised setting with the monotonicity constraint. Therefore, we proposed to modify the Baum-Welch algorithm to address the semi-supervised challenge, multiple sensors challenge, and the monotonicity state constraint as follows:

1) Addressing the challenges of *semi-supervised state* and *multiple sensors* The starting state and failure state are typically observable. For these states, we can directly estimate the mean and covariance matrix from the multiple sensor measurements. The mean $\boldsymbol{\mu}_{c,k}$ and covariance matrix $\boldsymbol{\Sigma}_{c,k}$ of observed state k under condition

c can be estimated as:

$$\begin{aligned}
\boldsymbol{\mu}_k^c &= \frac{1}{N_k^c}\sum_{i=1}^{n}\sum_{t=1}^{T_i} 1(S_{i,t}^c = k)\mathbf{O}_{i,t}^c \\
\boldsymbol{\Sigma}_k^c &= \frac{1}{N_k^c}\sum_{i=1}^{n}\sum_{t=1}^{T_i} 1(S_{i,t}^c = k)(\mathbf{O}_{i,t}^c - \boldsymbol{\mu}_k^c)(\mathbf{O}_{i,t}^c - \boldsymbol{\mu}_k^c)^{T_i} \\
\text{where, } N_k^c &= \sum_{i=1}^{n}\sum_{t=1}^{T_i} 1(S_{i,t}^c = k).
\end{aligned} \tag{6.6}$$

For the unobserved system states, we first use the forward-backward algorithm to estimate the posterior probability of unit i under condition c in state k and time t as $\gamma_{i,c,k}(t) = P(S_{i,t}^c = k|\{\mathbf{O}_{i,t}^c\}_{t=1}^{T_i})$. The mean and covariance matrix for each state k under condition c can be estimated as follows:

$$\begin{aligned}
\boldsymbol{\mu}_k^c &= \frac{1}{N_k^c}\sum_{i=1}^{n}\sum_{t=1}^{T_i} \gamma_{i,k}^c(t)\mathbf{O}_{i,t}^c \\
\boldsymbol{\Sigma}_k^c &= \frac{1}{N_k^c}\sum_{i=1}^{n}\sum_{t=1}^{T_i} \gamma_{i,k}^c(t)(\mathbf{O}_{i,t}^c - \boldsymbol{\mu}_k)(\mathbf{O}_{i,t}^c - \boldsymbol{\mu}_k^c)^{T_i} \\
\text{where, } N_k^c &= \sum_{i=1}^{n}\sum_{t=1}^{T_i} \gamma_{i,k}^c(t).
\end{aligned} \tag{6.7}$$

2) Addressing the challenges of *left-to-right state transition* and *monotone degradation signals* We have the prior knowledge that $\mu_{k'j} > \mu_{kj}$ should be larger for system state $k' > k$ for some sensor j, and $\mu_{k'j} < \mu_{kj}$ should be smaller for system state $k' > k$ for the other sensor j. We propose to sort the $(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k) = \text{sort}(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k)$ according to the majority voting procedures.

Furthermore, the initial probability is $\pi_{ik}^c = \gamma_{ik}^c(1)$ and the state transition matrix can be estimated in the following:

$$\begin{aligned}
\tilde{p}_{kk'}^c &= \frac{\sum_{t=1}^{T-1}\xi_{kk'}^c(t)}{\sum_{t=1}^{T_i-1}\gamma_k^c(t)} \\
\text{where, } \xi_{kk'}^c(t) &= P(S_{t-1} = k, S_t = k'|\mathbf{O}_c; \theta_c)
\end{aligned} \tag{6.8}$$

However, the estimated $p^c_{kk'}$ are without any constraint. To impose the left-to-right constraint, we propose to perform the following update steps.

$$p^c_{kk'} = \begin{cases} 0 & k < k' \\ \sum_{k=k'}^{K} \tilde{p}^c_{kk'} & k \geq k' \end{cases} \tag{6.9}$$

$$(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k) = \text{sort}(\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_k) \tag{6.10}$$

Through our simulation study, the combined sorting (6.10) and projection (6.9) to lower triangular matrix could dramatically increase the robust of the training procedure than using the projection (6.9) alone.

3) Estimating the parameters under multiple condition assumption. The estimation process is based on the assumption that different conditions will share same emission matrix but different transition matrix. The same emission matrix means that all machines will degrade through similar hidden states and the measurements under same hidden states but different conditions have same distributions. Different transition matrix means that the degradation speed under different conditions will be different. Under those two assumptions, we can compute the common emission matrix by first estimating the posterior probability of unit i under condition c in state k and time t as $\gamma_{i,c,k}(t) = P(S^k_{i,c,t} = 1 | \{\mathbf{O}_{i,c,t}\}_{t=1}^{T})$. The mean and covariance matrix for each state k can be estimated as follows:

$$\begin{aligned} \boldsymbol{\mu}_k &= \frac{1}{N_k} \sum_{c=1}^{M} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \gamma^c_{i,k}(t) \mathbf{O}^c_{i,t} \\ \boldsymbol{\Sigma}_k &= \frac{1}{N_k} \sum_{c=1}^{M} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \gamma^c_{i,k}(t) (\mathbf{O}^c_{i,t} - \boldsymbol{\mu}^c_k)(\mathbf{O}^c_{i,t} - \boldsymbol{\mu}^c_k)^{T_i} \\ \text{where, } N_k &= \sum_{c=1}^{M} \sum_{i=1}^{n} \sum_{t=1}^{T_i} \gamma^c_{i,k}(t). \end{aligned} \tag{6.11}$$

A remark is that discrete observation already provide enough resolution to derive the maintenance policy and to further refine the resolution to continuous is not nec-

Input: Number of state K, Multi-sensor observation $\mathbf{O}_{i,c,t}$

Output: Transition P_{ij}^{c}, emission $\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$

Function `Constained HMM()`:

  $\mu_k^c, \Sigma_k^c \longleftarrow S_{i,t}^c, \mathbf{O}_{i,t}^c$; //Estimate observed state emission using (6.6)

  while *Not converged* **do**

    $\mu_k^c, \Sigma_k^c \longleftarrow \gamma_{i,k}^c, \mathbf{O}_{i,t}^c$; //Estimate unobserved state emission using (6.7)

    $\tilde{p}_{kk'}^c \longleftarrow \mu_{c,k}, \Sigma_k^c, \mathbf{O}_{i,t}^c$; //Estimate transition (6.8)

    $P_{ij}^c \longleftarrow \tilde{p}_{kk'}^c$; //Projection and sorting using (6.10)

    $\mu_k, \Sigma_k \longleftarrow$

     $\mu_k^c, \Sigma_k^c, \mathbf{O}_{i,t}^c$; //Computing shared emission matrix using (6.11)

  end

end

Algorithm 1: Constrained HMM Fitting

essary but can also lead to the large computational burden in the POMDP model. An algorithm therefore should transform the continuous features $\mathbf{O}$ to the discrete observations $\mathbf{o}$. We utilize Gaussian Mixture Model to discretize the observations as $\arg\max_i p(\mathbf{o_i}|\mathbf{O})$. The probability can be calculated as following where ϕ, μ, σ represent weight, mean and variance. The model parameters can be estimated through expectation maximization:

$$p(\mathbf{o_i}|\mathbf{O}) = \frac{p(\mathbf{o_i}, \mathbf{O})}{p(\mathbf{O})} = \frac{p(\mathbf{o_i})p(\mathbf{O}|\mathbf{o_i})}{\sum_{j=1}^{k} p(\mathbf{o_j})p(\mathbf{O}|\mathbf{o_j})} = \frac{\phi_i \mathcal{N}(\mathbf{O}|\mu_i, \sigma_i)}{\sum_{j=1}^{k} \phi_j \mathcal{N}(\mathbf{O}|\mu_j, \sigma_j)} \tag{6.12}$$

Solve POMDP

In order to generate the optimal control policy, we use Point-Based Value Iteration (PBVI) algorithm (Pineau *et al.*, 2005). Specifically, we would like to estimate the actions a_t at time t of the POMDP problem defined in Section 6.2.3 assuming that

Input: Observing signal z, ϕ, μ, σ

Output: Belief space $\mathbf{b}$

Function `GMM()`**:**

  $O \longleftarrow$ $\{X_{rms}(z), \ldots, X_e(z)\}$; //Abstract features from signals using expressions in Table 2

  $\mathbf{b}(\mathbf{o}|\mathbf{z}) \longleftarrow \frac{\phi_i \mathcal{N}(\mathbf{O}|\mu_i, \sigma_i)}{\sum_{j=1}^{k} \phi_j \mathcal{N}(\mathbf{O}|\mu_j, \sigma_j)}$; // Obtain the belief on the continuous observation over the discrete observations

end

Algorithm 2: GMM Classifier

parameters of the transition probability matrix T are given. We will discuss how to estimate these parameters in Section 6.2.4.

To do so, we need to introduce a definition, named the belief state, denoted as $b = b(S) \in \mathcal{B}$, which represents the likelihood that the system is in latent state S under a set of belief $\mathcal{B}$. First, define

$$Z(S', a, O') = Pr(O_{t+1} = O' | S_{t+1} = S',\ a) \tag{6.13}$$

as the distribution that the an observation will be captured when the engine is in state S' and action a_t was conducted. Notice that this is the decoding problem (as discussed in Section 2.1), which can be solved by the HMM using the historical observations. Then the transformation of the belief state to S' given action a and observation O can be further expressed in the transition expression as follows:

$$b_o^a(S') = \frac{Z(S', a, O) \sum\limits_{S \in \mathcal{S}} X(S, a, S') b(S)}{\sum\limits_{S' \in \mathcal{S}} \left\{ Z(S', a, O) \sum\limits_{S \in \mathcal{S}} X(S, a, S') b(S) \right\}} \tag{6.14}$$

Let $J_\pi(b_0)$ denote the expected total discounted reward given b_0 under policy $\pi \in \boldsymbol{\pi}$; that is

$$\underset{\boldsymbol{\pi}}{\text{Maximize}} \quad J_\pi(b) = \mathbf{E}[\sum_{t=1}^{\infty} \eta^t R(b, \pi) | b_0], \tag{6.15}$$

where

$$R(b,a) = \sum_{S \in \mathcal{S}} b(S) r(S,a),$$

$0 < \eta < 1$ is a discount factor and π includes all the decisions to determine the PM policies.

Let $V_t(b)$ denote the expected total achievable reward for the period t to the infinite horizon given our belief at time t, with $V_0(\cdot) = 0$, then

$$V_t(b) = \min_{a \in \boldsymbol{\pi}} \Big\{ R(b,a) + \eta \sum_{b'} Pr(O|a,b) V_{t+1}(b_o^a) \Big\}, \tag{6.16}$$

in which

$$Pr(O|a,b) = \sum_{S' \in \mathcal{S}} \Big\{ Z(S',a,O) \sum_{S \in \mathcal{S}} X(S,a,S') b(S) \Big\}.$$

On the right hand side of Equation (6.16), it explicitly describes the two components of the reward: the reward for the current state at time t, and the discounted reward of reaching all the possible system states starting from the next time period. Then the optimal policies starting from time zero can be calculated by using the value iteration approach (Puterman, 2005). In order to attain the optimal policy and optimal value, we use Point-Based Value Iteration (PBVI) algorithm (Pineau *et al.*, 2005), which generates set of *alpha*-vectors with a set of belief points, which enables generating the optimal solutions.

6.2.5 Obtain the Actions for Observing Signals

By integrating the results of the POMDP and the GMM algorithms, finally we can obtain the actions for all the given degradation signals.

For a given set of observing signals, it will first be transformed to a set of discrete observations using the feature abstraction approach as discussed in Section 4.2. Then the GMM algorithm will deliver the set of estimated probabilities, which represents that the belief that the observation matches with the predefined observations in the

Input: A POMDP tuple $(h_S, A, tr, R, \Omega, O, b_0)$

Output: α, B

Function `PBVI()`**:**
  $B \longleftarrow \{b_0\}$;
  while *Not converged* **do**
    Improve(V,B)
    B $\longleftarrow$ Expand(B)
  end
end
Function `Improve(`V, B`)`**:**
  while *Not converged* **do**
    foreach $b \in B$ **do**
      $\alpha \longleftarrow$ Backup(b,V)
      $V \longleftarrow V \cup \{\alpha\}$
    end
  end
end
Function `Expand(`B`)`**:**
  $B' \longleftarrow B$;
  foreach $b \in B$ **do**
    $\Gamma(b) \longleftarrow \{b^{a,o} | Pr(o|b,a) > 0\}$
    $B' \longleftarrow B' \cup argmax_{b' \in \Gamma(b)} \|B, b'\|_2$
  end
  return B'
end
Function `Backup(`b, V`)`**:**
  foreach $b_0 \in b$ **do**
    $\beta_{a,O} \longleftarrow argmax_{\alpha \in V} \alpha b_o^a$
    $\beta_a(s) \longleftarrow R(s,a) + \eta \sum_S \sum_{S'} Z(S', a, O) X(S, a, S') \beta_{a,O}$
    $\beta \longleftarrow argmax_{\beta_a} \beta_a b$
  end
  return β
end

Algorithm 3: PBVI Algorithm to Solve POMDP

POMDP framework, i.e., $\mathbf{B(o)} = \{B(o_1), B(o_2), , \ldots, B(o_J)\}$. Then the belief on the hidden states of this observation can be obtained usin the expressing $\bar{\mathbf{b}}(\mathbf{o}) = \mathbf{B}(\mathbf{o})\mathbf{b^T}(\mathbf{s})$. The action that the machine should taken under the observation o then will be the one associated to the α vector that brings the maximum reward, which is expressed as:

$$a = \arg\min_{a_i : i \in \{1,2,\ldots|\alpha|\}} \ \alpha_{\mathbf{i}} \mathbf{b}(o)\mathbf{B^T}(o). \tag{6.17}$$

Input: Observing signal z, $\mathbf{\Lambda}$,$\mathbf{a}$, emission matrix $\mathbf{B}$

Output: Action a

Function `Decision()`**:**

  $\mathbf{b'(o|z)} \longleftarrow$ `GMM`(z); // Obtain the belief on the discrete observations using GMM algorithms

  $\mathbf{b(s|z)} \longleftarrow \mathbf{Bb'(o|z)}$;// Get the hidden state belief

  $a \longleftarrow \arg\min_{a_l : l \in \{1,2,\ldots|\alpha|\}} \ \mathbf{b^T(s|z)}\alpha_{\mathbf{l}}$.

end

Algorithm 4: Algorithm to Obtain the Action for Observing Signals

The overall algorithm is summarized as shown in Figure 6.2. For the offline training, as shown on the left side, historical data sets are collected to abstract the features and thereafter used to obtain the parameters for the HMM model and the POMDP model, which will ultimately return the vectors for decision making given different observations. When the model is conducted for real-time control, the signal is abstracted into the features, which will be classified as the belief vectors on the discrete observations. The belief vector, together with the alpha vectors generated from the training stage, will be used to determine the ultimate control action at the exact time.

6.3 Case Studies

In this section, we aim to evaluate the proposed algorithm using a real bearing dataset from a laboratory experimental platform (Wang *et al.* (2018)) with multiple operating conditions. We will first give a brief introduction of the dataset in Section 6.3.1 and how to extract useful features in Section 6.3.2. We then like to show how the degradation dynamics and states can be estimated in Section 6.3.3. Finally, we will discuss how to optimize the preventative maintenance and multiple operating conditions in Section 6.3.4.

6.3.1 Introduction to Dataset

To validate the effectiveness of the proposed framework, we use the XJTU-SY bearing dataset developed from a laboratory experimental platform (Wang *et al.* (2018)). Accelerated degradation tests of bearings conducted on the platform, which provides data that characterizes the bearing degradation process during the entire operating lifetime. The testing bed of bearing degradation experiments is shown in Figure 6.4, which consists an alternating current (AC) induction motor, a support shaft, two support roller bearings, a hydraulic loading system and a motor speed controller. The acceleration degradation test is conducted for bearings under different operation conditions, typically different rotating speed and radial force, which can be controlled by the hydraulic loading system and the speed controller of the AC induction motor. The experimental operating conditions are listed out as in Table 6.1, as well as the performance of each condition, such as the lifetime and the Maximum Amplitude (MA).

The vibration signals from the horizontal direction, which also matches the loading direction, is captured to track the degradation process. The signals are collected with

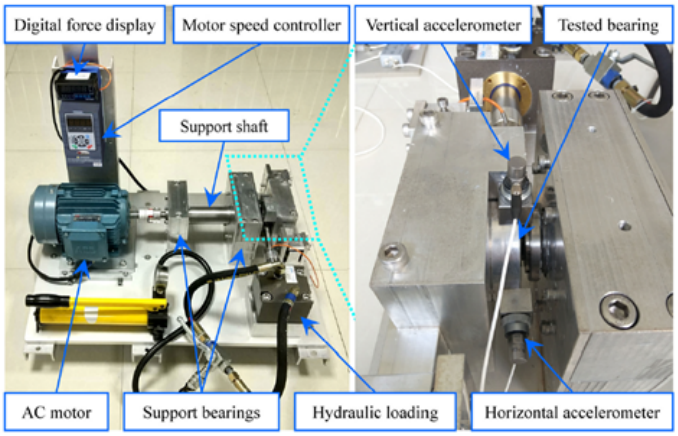


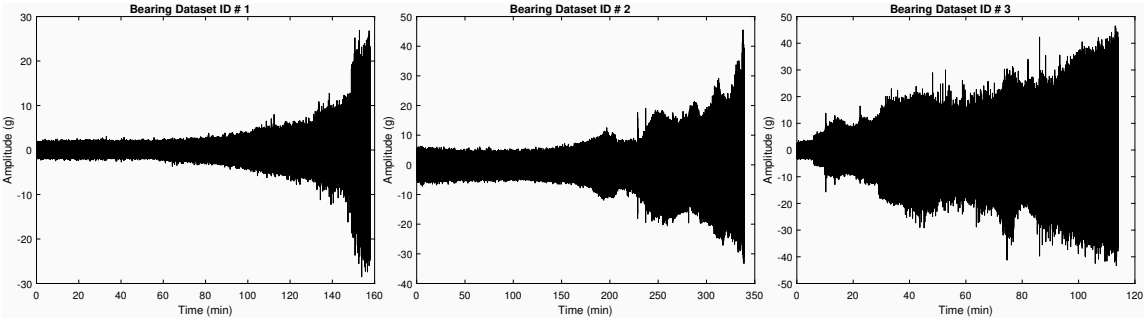
Figure 6.4: The Testbed of the Experiments

ID	Operating Condition	Bearing Lifetime	Max MA
1	35Hz/12kN	158 min	28 g
2	37.5Hz/11kN	339 min	45 g
3	40Hz/10kN	114 min	45 g

Table 6.1: Information on the Bearing Degradation Dataset

sampling frequency 25.6 kHz (32768 data points per minute). The degradation process is therefore monitored and forced to stop when the MA values of the signal reaches out of the predetermined safety threshold, which is considered as machine failure.

Also, the degradation plot to the dataset under different operating conditions are shown in Figure 6.5. Overall, it can be found out that an increasing trend is presented on the amplitudes of the vibration signals throughout the entire operating time. For some operating conditions, i.e., as shown on sub-figure (a) and (b), a fluctuation on the signal values occurs before the fast increase of the amplitudes.



(a) (b) (c)

Figure 6.5: The Plot of Vibration Signals under Different Operating Conditions

6.3.2 Feature Extraction from the Raw Signals

For the process to derive the degradation signals from the raw degradation data (i.e., the vibration data), the following consecutive steps are commonly applied.

Eleven statistical features are extracted from the vibration signals, i.e., root mean square, mean value, standard deviation, skewness, kurtosis, shape factor, peak to peak, energy, etc. The expressions of the features are listed out as shown in Table 6.2, in which x_i is the i^{th} sample of the timed signal, and N is the total number of the samples. These features indicate the amplitude and reflect the distribution of a signal over the time domain.

6.3.3 Parameter Estimation

We then apply the proposed constrained-HMM models to classify the original signal into six hidden unobserable states, with the last state representing the failure observable state. In Figure 6.6, four representing features are selected under the three operating conditions. The values of these features are plotted under each operation condition, with the colored bars representing the level of the mean values under different hidden states. It can be found out that the value of all these features indicate

an upward trend overwhelmingly, which is consistent with the trend of the original signal as well. Such behaviour is also adaptive to the left and right constraint in the HMM model. Besides, the range of these features are also different under different conditions. For example, under the feature "Std", the signal value is below 10 for condition 0 (lower capacity) while it surpasses 15 for condition 3 (higher capacity). Therefore, for the lower capacity case, the system has less probability to retain on the inferior hidden states than the other conditions before the failure of the machine is observed. Furthermore, the posterior probabilities for the hidden states are shown as in Figure 6.7. From the results, we can see that the system is changing states gradually. The first state and last state are able to capture the initial and failure signal pretty clear. However, there are much more noise and in the middle states. For example, state 4 is related to signals close to both start and end. The reason is that we assume that multiple conditions will share the same emission matrix. Thus, one state might be corresponding to signals from different time.

6.3.4 Multiple Condition Preventative Maintenance Policy

We then consider the control of preventative maintenance and optimize the three operating conditions. Based on the estimated six latent states, we can generate the PM policies for the engine. The objective is to find best of the combination of the capacities and PM such that the total long-term production cost is the minimum. The

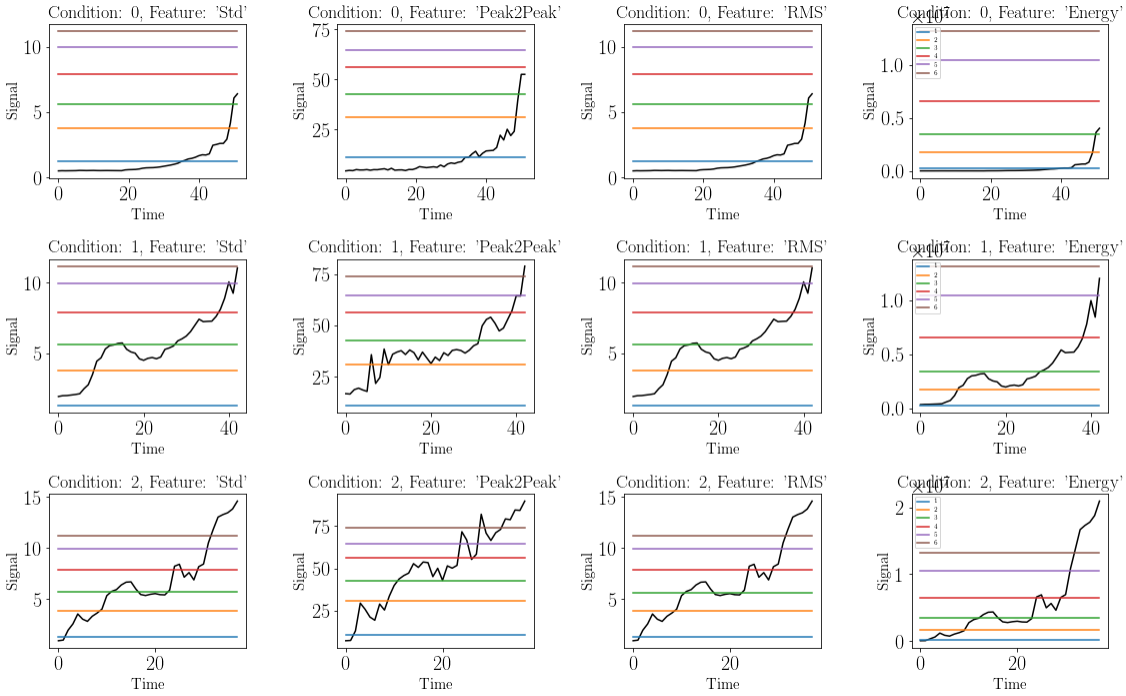


Figure 6.6: State Estimation for All Conditions

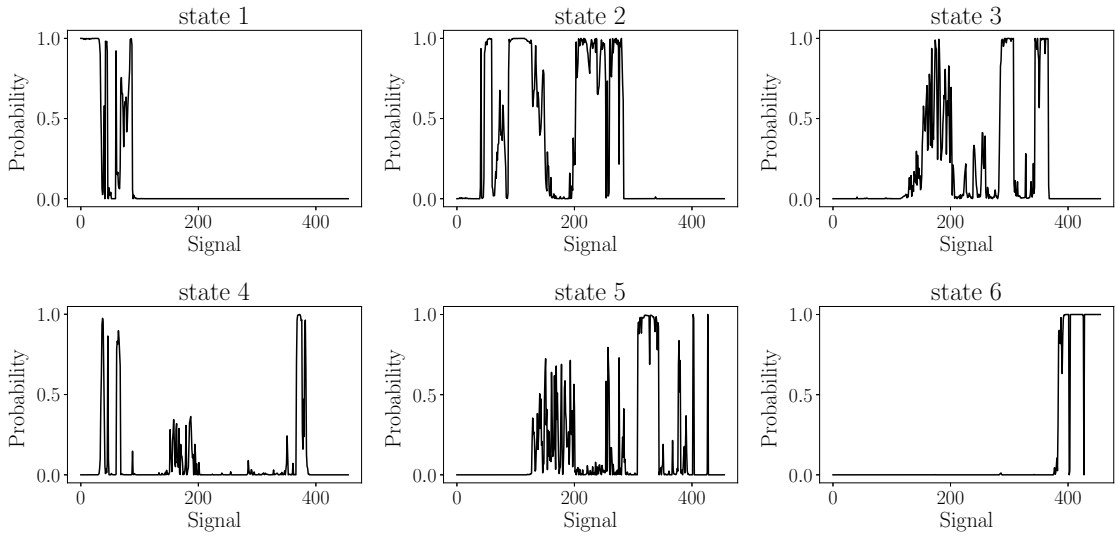


Figure 6.7: Posterior Probability for the Hidden States

state transition matrix for all the operating conditions can be expressed as follows:

$$\mathbf{A_1} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & F \end{array} \\ \begin{bmatrix} 0.9330 & 0.0670 & 0 & 0 & 0 & 0 \\ 0 & 0.8735 & 0.1265 & 0 & 0 & 0 \\ 0 & 0 & 0.4639 & 0.5361 & 0 & 0 \\ 0 & 0 & 0 & 0.0844 & 0.9156 & 0 \\ 0 & 0 & 0 & 0 & 0.1091 & 0.8909 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\mathbf{A_2} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & F \end{array} \\ \begin{bmatrix} 0.8119 & 0.1881 & 0 & 0 & 0 & 0 \\ 0 & 0.0428 & 0.9572 & 0 & 0 & 0 \\ 0 & 0 & 0.9048 & 0.0952 & 0 & 0 \\ 0 & 0 & 0 & 0.7707 & 0.2293 & 0 \\ 0 & 0 & 0 & 0 & 0.6425 & 0.3575 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$\mathbf{A_3} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ F \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & F \end{array} \\ \begin{bmatrix} 0.5606 & 0.4394 & 0 & 0 & 0 & 0 \\ 0 & 0.4445 & 0.5555 & 0 & 0 & 0 \\ 0 & 0 & 0.8823 & 0.1177 & 0 & 0 \\ 0 & 0 & 0 & 0.5407 & 0.4593 & 0 \\ 0 & 0 & 0 & 0 & 0.7195 & 0.2805 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

The emission matrix is listed out as follows:

$$\mathbf{B} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ F \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0.1983 & 0.0000 & 0.8013 & 0.0000 & 0.0003 \\ 0.0025 & 0.0098 & 0.5522 & 0.0000 & 0.4355 \\ 0.0000 & 0.2810 & 0.1381 & 0.0000 & 0.5809 \\ 0.0000 & 0.7224 & 0.0279 & 0.0014 & 0.2359 \\ 0.0000 & 0.7531 & 0.0001 & 0.1630 & 0.0838 \\ 0.0000 & 0.7651 & 0.0000 & 0.1907 & 0.0442 \end{bmatrix} \end{array}$$

Besides, the cost matrix can be found out as follows:

$$\mathbf{c} = \begin{array}{c} Action/State \\ 1 \\ 2 \\ 3 \\ PM \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & F \end{array} \\ \begin{bmatrix} 1.2 & 1.2 & 1.2 & 1.2 & 1.2 & -25 \\ 1.3 & 1.3 & 1.3 & 1.3 & 1.3 & -25 \\ 1.5 & 1.5 & 1.5 & 1.5 & 1.5 & -25 \\ -6 & -6 & -6 & -6 & -6 & -25 \end{bmatrix} \end{array}$$

By solving the POMDP problem, we can obtain the collection of belief vectors (The α vectors in the algorithm) and the corresponding actions (a). These belief vectors are the sampled points which partitioned the entire belief space. Figure 6.4 lists out 5 of the belief vectors obtained by the PBVI algorithm.

Furthermore, in Figure 6.4, we list out five sample beliefs, and the generated actions obtained by the algorithm. It can be found out that when the system in the better operating conditions, i.e., higher belief on state 1 and 2, the algorithm would suggest lower capacities. For the belief with larger values intermediate states, a larger capacity is suggested. For the system approaching to failure while still on operation, the algorithm will recommend conducting PM. Besides, in Figure 6.8, the change of the actions and system states as time elapses are presented. It can be found out that

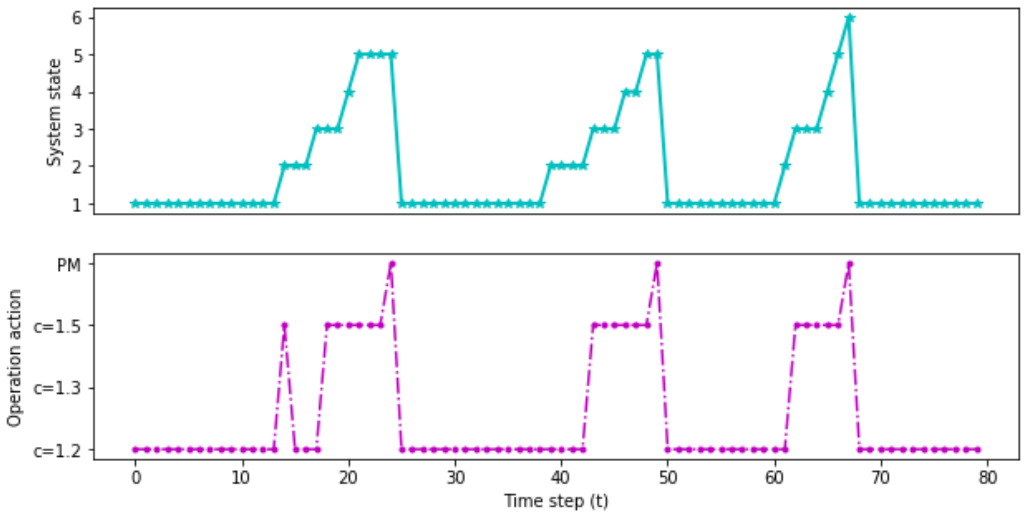


Figure 6.8: Illustrative Examples on System States and Corresponding Actions

when the system is in good operating states, i.e., state 1 and state 2, the system is tended to maintain lower capacity. However, when the system is approaching to inferior states, i.e., state 3, 4, and 5, larger capacities are suggested. The system then is recovered to the best operating activities through the maintenance when it is almost to failure. As it is indicated from the transition matrices, when the system is in state 1 or 2, the probability that the system retains in the same operating state is significantly higher under the operating condition $c = 1.2$ is higher than the other two conditions (i.e., 0.9330 in state 1 of $\mathbf{A_1}$ comparing to 0.5606 for condition $c = 1.5$ in matrix $\mathbf{A_3}$). By using the lower capacity, the degradation process is slower enough to generate more rewards than the other two operating conditions. For state 4 and 5, however, condition $c = 1.5$ has larger transition probabilities than $c = 1.2$. The algorithm jointly considers the degradation process and the instant capacity rewards, which results in the decision for adapting the condition with the largest capacity.

We then conduct simulation experiments to compare the performance of the POMDP policies with the other policies, i.e., simply following the single condition.

500 simulation experiments are conducted, with 50,000 cycle time in each simulation. The results are shown as in Table 6.5. It can be found out that the POMDP is an effective framework on maintaining high performance of the manufacturing system with different operating capacities and maintenance options.

To implement the model in real-time with the degradation signals, at the beginning of a cycle, by collecting and transforming the real-time degradation signals, features can be abstracted and are used as the observation input. Then by using the α-vectors, the matching belief states can be identified and the corresponding actions can be obtained. If PM is needed, then the bearing replacement is conducted. Otherwise, if capacity switch is needed, then the operator will change the machine capacity following the obtained action. After the action, new signals will be captured and the above procedures will be repeated.

6.4 Conclusion

In this chapter, we develop a systematic framework to generate the optimal capacity and PM policies, given the fact that the observation on the system state is imperfect. A semi-supervised left-to-right constrained HMM model is built up to estimate the parameters of degrading unit under different operating conditions. The information from multiple sensors is utilized to identify the system state. Furthermore, we propose a POMDP method based on the developed HMM to determine the optimal preventative maintenance strategy for the system under consideration. In the case study, We utilize the bearing degradation data set to demonstrate the effectiveness of the proposed method. The results show that the proposed method is effective in identifying the progression of the system states and capture the system dynamics with details. In addition, the generated policy is effective in optimizing engine performance in the long term.

In the future, this work can be extended to more complicated systems with multiple degrading units, the behaviors of which are tightly coupled. One typical example is a manufacturing system with multiple machines where each machine exhibits similar degrading dynamics. Analyzing such complicated systems will expose great challenges on system modeling and computational efficiency. Second, the structure of the models can be further extended, i.e., the HMM model can be extended to Hidden Semi-Markov Model (HSMM) with multiple layers. Such an extension enables better estimation on different time and observation frequency of the degradation process. Besides, new system improvement and control methods will be needed to incorporate the additional complexity these modeling extensions will bring.

Feature	Expression
RMS	$X_{rms} = (\frac{1}{N}\sum_{i=1}^{N} x_i^2)^{\frac{1}{2}}$
Mean	$X_m = \frac{1}{N}\sum_{i=1}^{N} x_i$
Standard deviation (Std)	$X_{std} = (\frac{1}{N}\sum_{i=1}^{N}(x_i - X_m)^2)^{\frac{1}{2}}$
Skewness	$X_{ss} = \frac{1}{X_{rms}^3}\sum_{i=1}^{N}(x_i - X_m)^3$
Kurtosis	$X_{ks} = \frac{1}{X_{rms}^4}\sum_{i=1}^{N}(x_i - X_m)^4$
Peak to peak (P-P)	$X_{pp} = x_{max} - x_{min}$
Crest factor	$X_{cf} = \frac{x_{max}}{X_{rms}}$
Shape factor	$X_{sf} = \frac{x_{rms}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Impulse factor	$X_{if} = \frac{x_{max}}{\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert}$
Margin factor	$X_{mf} = \frac{x_{max}}{(\frac{1}{N}\sum_{i=1}^{N}\lvert x_i\rvert)^2}$
Energy	$X_e = \sum_{i=1}^{N} x_i^2$

Table 6.2: Feature Descriptions of the Dataset

Belief	Action	Value function					
		State 1	State 2	State 3	State 4	State 5	F
1	C=1.2	6.32	5.58	4.24	-8.79	-27.3	-84.16
2	C=1.2	6.02	4.52	4.26	1.08	-11.7	-55.55
3	C=1.5	5.27	2.8	2.68	1.21	-10.89	-54.56
4	C=1.5	5.87	4.1	3.02	0.85	-3.87	-41.73
5	PM	-0.95	-0.95	-0.95	-0.95	-0.95	-19.95
...(200+)	...	...	...	...	...	...	...

Table 6.3: A list of Belief Vectors and Actions

Sample ID	Action	Belief					
		State 1	State 2	State 3	State 4	State 5	F
1	C=1.2	0.8185	0.0888	0.0927	0	0	0
2	C=1.5	0	0.2227	0.6484	0.0931	0.0146	0.0213
3	C=1.5	0	0	0.3551	0.2606	0.3169	0.0674
4	PM	0	0	0	0.3373	0.1655	0.4972
5	PM	0	0	0	0	0.6291	0.3709

Table 6.4: Sample Belief Values and Its Corresponding Actions

	Proposed Method	C1	C2	C3
Mean	14617.09	11127.31	5846.04	-9213.23
Std	630.71	653.41	616.38	774.27

Table 6.5: Performance Comparisons

Chapter 7

CONCLUSION AND FUTURE WORK

In this dissertation, analytical approaches are conducted to evaluate the performance of the manufacturing systems, considering the machine degradation process. Specifically, for the serial production systems, models are developed to evaluate the overall system performance. Bottleneck identification tools are created to find the machines and states that has the biggest influence on the up-time, and down-time, if a slight improvement is achievable. Furthermore, flexible maintenance options are considered to further improve the system performance, through recovering the system states from any of the non-perfect operating state. A system conditional-based policy generating tool is developed to include all possible conditions of the two machine and one buffer system, and an aggregation method is developed to find applicable policies for the general serial production lines. Besides, for the flexible systems with the options on production capacity controls, we generate an efficient framework, which integrate the real-time degradation signals to optimally control the production rate and PM actions to maximize the long-term system performance. Numerical experiments have shown that the method is effective and better than the current state of the art.

In the future, the work can be extended through the following directions:

First, in terms of the degradation process, different types of degradation can be considered, not limited to the degradation of the machine conditions. For example, the position accuracy of the tools can also lead to the performance deviation from the target, which can also be considered. Furthermore, such multiple factors can impact the degradation process, such as the temperature, workload, speed,etc. These

information should also be considered. Moreover, in terms of the channels to collect the real-time information, multiple signals will be considered.

Second, in terms of the structure of the manufacturing systems, more complicated structures can be considered. Beyond the structure, the serial production system considered in this work, other types of production systems, such as the assembly lines and production networks are expected to discussed. Besides, since the two machines and one buffer case, and also the serial production lines are the typical structures in the large production systems, the results and findings discussed in this work is expected to extend. Meanwhile, considering the complexity in the large networks, approximation methods are also expected to develop.

Third, for production controls, to make full use of the real-time information, it is expected that more systematic control strategies are developed on the basis of the real-time system performance evaluation model. Besides, for the complex systems, a hybrid modeling and optimization approach is expected. Typically analytical models are fast to run but can have reduced accuracy. On the other hand simulation models can achieve high accuracy, but only at the cost of large simulation time and number of replications. Traditionally, the research has been focusing on the development of models able to achieve a satisfactory trade off between accuracy and computational effort. Nevertheless, such an approach implies the choice of a single model to approximate the system behavior. The lack of a generic model that can deliver high accuracy and low computational cost for production systems should be addressed.

REFERENCES

Alaswad, S. and Y. Xiang, "A review on condition-based maintenance optimization models for stochastically deteriorating system", Reliability Engineering & System Safety **157**, 54–63 (2017).

AlDurgam, M. M. and S. O. Duffuaa, "Optimal joint maintenance and operation policies to maximise overall systems effectiveness", International Journal of Production Research **51**, 5, 1319–1330 (2013).

Ambani, S., S. M. Meerkov and L. Zhang, "Feasibility and optimization of preventive maintenance in exponential machines and serial lines", IIE transactions **42**, 10, 766–777 (2010).

Barlow, R. and L. Hunter, "Optimum preventive maintenance policies", Operations research **8**, 1, 90–100 (1960).

Belmansour, A.-T. and M. Nourelfath, "An aggregation method for performance evaluation of a tandem homogenous production line with machines having multiple failure modes", Reliability Engineering & System Safety **95**, 11, 1193–1201 (2010).

Bian, L. and N. Gebraeel, "Stochastic modeling and real-time prognostics for multi-component systems with degradation rate interactions", IIE Transactions **46**, 5, 470–482 (2014).

Biller, S., J. Li, S. P. Marin, S. M. Meerkov and L. Zhang, "Bottlenecks in bernoulli serial lines with rework", IEEE Transactions on Automation Science and Engineering **7**, 2, 208–217 (2010).

Blake, M. P. and W. S. Mitchell, *Vibration and acoustic measurement handbook* (Spartan Books, 1972).

Brundage, M. P., Q. Chang, Y. Li, J. Arinez and G. Xiao, "Sustainable manufacturing performance indicators for a serial production line", IEEE Transactions on Automation Science and Engineering **13**, 2, 676–687 (2016).

Byon, E. and Y. Ding, "Season-dependent condition-based maintenance for a wind turbine using a partially observed markov decision process", IEEE Transactions on Power Systems **25**, 4, 1823–1834 (2010).

Chakraborty, T., B. C. Giri and K. Chaudhuri, "Production lot sizing with process deterioration and machine breakdown", European Journal of Operational Research **185**, 2, 606–618 (2008).

Chiang, S.-Y., C.-T. Kuo, J.-T. Lim and S. Meerkov, "Improvability of assembly systems i: Problem formulation and performance evaluation", Mathematical Problems in Engineering **6**, 4, 321–357 (2000).

Chiang, S.-Y., C.-T. Kuo and S. M. Meerkov, "c-bottlenecks in serial production lines: identification and application", Mathematical problems in engineering **7**, 6, 543–578 (2001).

Cholette, M. E. and D. Djurdjanovic, "Degradation modeling and monitoring of machines using operation-specific hidden markov models", IIE Transactions **46**, 10, 1107–1123 (2014).

Colledani, M. and T. Tolio, "Integrated quality, production logistics and maintenance analysis of multi-stage asynchronous manufacturing systems with degrading machines", CIRP Annals-Manufacturing Technology **61**, 1, 455–458 (2012).

de Jonge, B., R. Teunter and T. Tinga, "The influence of practical factors on the benefits of condition-based maintenance over time-based maintenance", Reliability engineering & system safety **158**, 21–30 (2017).

Doksum, K. A. and A. Hbyland, "Models for variable-stress accelerated life testing experiments based on wener processes and the inverse gaussian distribution", Technometrics **34**, 1, 74–82 (1992).

Douer, N. and U. Yechiali, "Optimal repair and replacement in markovian systems", Stochastic Models **10**, 1, 253–270 (1994).

Eruguz, A. S., T. Tan and G.-J. van Houtum, "A survey of maintenance and service logistics management: Classification and research agenda from a maritime sector perspective", Computers & Operations Research (2017).

Fitouhi, M.-C., M. Nourelfath and S. B. Gershwin, "Performance evaluation of a two-machine line with a finite buffer and condition-based maintenance", Reliability Engineering & System Safety **166**, 61–72 (2017).

Gebennini, E. and S. B. Gershwin, "Modeling waste production into two-machine–one-buffer transfer lines", IIE Transactions **45**, 6, 591–604 (2013).

Gebraeel, N. Z., M. A. Lawley, R. Li and J. K. Ryan, "Residual-life distributions from component degradation signals: A bayesian approach", IIE Transactions **37**, 6, 543–557 (2005).

Goldratt, E. M., *Theory of constraints* (North River Croton-on-Hudson, 1990).

Gorjian, N., L. Ma, M. Mittinty, P. Yarlagadda and Y. Sun, "A review on degradation models in reliability analysis", in "Engineering Asset Lifecycle Management", pp. 369–384 (Springer, 2010).

Gouiaa-Mtibaa, A., S. Dellagi, Z. Achour and W. Erray, "Joint integrated maintenance-quality policy with reworking activity", in "International Conference on Industrial Engineering and Systems Management", pp. 1336–1342 (2015).

Groenevelt, H., L. Pintelon and A. Seidmann, "Production lot sizing with machine breakdowns", Management Science **38**, 1, 104–123 (1992).

Gu, X., *Real-Time Maintenance Policies in Manufacturing Systems.*, Ph.D. thesis, Doctoral dissertation, The University of Michigan (2016).

Guner, H. U., R. B. Chinnam and A. Murat, "Simulation platform for anticipative plant-level maintenance decision support system", International Journal of Production Research **54**, 6, 1785–1803 (2016).

Hao, L., K. Liu, N. Gebraeel and J. Shi, "Controlling the residual life distribution of parallel unit systems through workload adjustment", IEEE Transactions on Automation Science and Engineering **14**, 2, 1042–1052 (2015).

Hao, L., K. Liu, N. Gebraeel and J. Shi, "Controlling the residual life distribution of parallel unit systems through workload adjustment", IEEE Transactions on Automation Science and Engineering **14**, 2, 1042–1052 (2017).

Heng, A., S. Zhang, A. C. Tan and J. Mathew, "Rotating machinery prognostics: State of the art, challenges and opportunities", Mechanical systems and signal processing **23**, 3, 724–739 (2009).

Icten, Z. G., S. M. Shechter, L. M. Maillart and M. Nagarajan, "Optimal management of a limited number of replacements under markovian deterioration", IIE Transactions **45**, 2, 206–214 (2013).

Inman, R. R., D. E. Blumenfeld, N. Huang, J. Li and J. Li, "Survey of recent advances on the interface between production system design and quality", IIE transactions **45**, 6, 557–574 (2013).

Ja, S.-S., V. G. Kulkarni, A. Mitra and J. G. Patankar, "A nonrenewable minimal-repair warranty policy with time-dependent costs", IEEE Transactions on Reliability **50**, 4, 346–352 (2001).

Jacobs, D. and S. M. Meerkov, "Mathematical theory of improvability for production systems", Mathematical Problems in Engineering **1**, 2, 95–137 (1995a).

Jacobs, D. and S. M. Meerkov, "A system-theoretic property of serial production lines: improvability", International Journal of Systems Science **26**, 4, 755–785 (1995b).

Jardine, A. K., D. Lin and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance", Mechanical systems and signal processing **20**, 7, 1483–1510 (2006).

Joo, S.-J., "Scheduling preventive maintenance for modular designed components: A dynamic approach", European Journal of Operational Research **192**, 2, 512–520 (2009).

Ju, F., J. Li, G. Xiao and J. Arinez, "Integrated model of productivity and quality in serial production lines with repairs", in "2013 IEEE International Conference on Automation Science and Engineering, CASE 2013", (2013a).

Ju, F., J. Li, G. Xiao and J. Arinez, "Quality flow model in automotive paint shops", International Journal of Production Research **51**, 21, 6470–6483 (2013b).

Ju, F., J. Li, G. Xiao, J. Arinez and W. Deng, "Modeling, analysis, and improvement of integrated productivity and quality system in battery manufacturing", IIE Transactions **47**, 12, 1313–1328 (2015).

Kang, N., F. Ju and L. Zheng, "Transient analysis of geometric serial lines with perishable intermediate products", IEEE Robotics and Automation Letters **2**, 1, 149 – 156 (2017).

Kang, Y. and F. Ju, "Integrated analysis of productivity and machine condition degradation: Performance evaluation and bottleneck identification", IISE transactions (2018).

Keizer, M. C. O., S. D. P. Flapper and R. H. Teunter, "Condition-based maintenance policies for systems with multiple dependent components: A review", European Journal of Operational Research (2017).

Kimemia, J. and S. B. Gershwin, "An algorithm for the computer control of a flexible manufacturing system", AIIE Transactions **15**, 4, 353–362 (1983).

Klutke, G.-A. and Y. Yang, "The availability of inspected systems subject to shocks and graceful degradation", IEEE Transactions on Reliability **51**, 3, 371–374 (2002).

Kouedeu, A. F., J.-P. Kenné, P. Dejax, V. Songmene and V. Polotski, "Production planning of a failure-prone manufacturing/remanufacturing system with production-dependent failure rates", Applied Mathematics **5**, 10, 1557 (2014).

Kouki, M., S. Dellagi, Z. Achour and W. Erray, "Optimal integrated maintenance policy based on quality deterioration", in "IEEE International Conference on Industrial Engineering and Engineering Management", pp. 838–842 (2014).

Le Bihan, H. and Y. Dallery, "A robust decomposition method for the analysis of production lines with unreliable machines and finite buffers", Annals of Operations Research **93**, 1-4, 265–297 (2000).

Lee, S., X. Gu and J. Ni, "Stochastic maintenance opportunity windows for unreliable two-machine one-buffer system", Expert Systems with Applications **40**, 13, 5385–5394 (2013a).

Lee, S., L. Li and J. Ni, "Markov-based maintenance planning considering repair time and periodic inspection", Journal of Manufacturing Science and Engineering **135**, 3, 031013 (2013b).

Lee, S. and J. Ni, "Joint decision making for maintenance and production scheduling of production systems", The International Journal of Advanced Manufacturing Technology **66**, 5-8, 1135–1146 (2013).

Li, H. and A. Parlikad, "Study of dynamic workload assignment strategies on production performance", IFAC-PapersOnLine **50**, 1, 13710–13715 (2017).

Li, J., *Production variability in manufacturing systems: a systems approach* (PhD Thesis, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA., 2000).

Li, J. and S. M. Meerkov, "Production variability in manufacturing systems: Bernoulli reliability case", Annals of Operations Research **93**, 1-4, 299–324 (2000).

Li, J. and S. M. Meerkov, "Evaluation of throughput in serial production lines with non-exponential machines", in "Analysis, control and optimization of complex dynamic systems", pp. 61–90 (Springer, 2005).

Li, J. and S. M. Meerkov, *Production systems engineering* (Springer Science & Business Media, 2008).

Liao, H., E. A. Elsayed and L.-Y. Chan, "Maintenance of continuously monitored degrading systems", European Journal of Operational Research **175**, 2, 821–835 (2006).

Lin, G. C. and D.-C. Gong, "On a production-inventory system of deteriorating items subject to random machine breakdowns with a fixed repair time", Mathematical and Computer Modelling **43**, 7, 920–932 (2006).

Liu, B., Z. Xu, M. Xie and W. Kuo, "A value-based preventive maintenance policy for multi-component system with continuously degrading components", Reliability Engineering & System Safety **132**, 83–89 (2014).

Liu, K., N. Z. Gebraeel and J. Shi, "A data-level fusion model for developing composite health indices for degradation modeling and prognostic analysis", IEEE Transactions on Automation Science and Engineering **10**, 3, 652–664 (2013a).

Liu, X., J. Li, K. N. Al-Khalifa, A. S. Hamouda, D. W. Coit and E. A. Elsayed, "Condition-based maintenance for continuously monitored degrading systems with multiple failure modes", IIE Transactions **45**, 4, 422–435 (2013b).

Liu, Y. and H.-Z. Huang, "Optimal selective maintenance strategy for multi-state systems under imperfect maintenance", IEEE Transactions on Reliability **59**, 2, 356–367 (2010).

Liyanage, J. P. and U. Kumar, "Towards a value-based view on operations and maintenance performance management", Journal of Quality in Maintenance Engineering **9**, 4, 333–350 (2003).

Lu, C. J. and W. O. Meeker, "Using degradation measures to estimate a time-to-failure distribution", Technometrics **35**, 2, 161–174 (1993).

Lu, Z., W. Cui and X. Han, "Integrated production and preventive maintenance scheduling for a single machine with failure uncertainty", Computers & Industrial Engineering **80**, 236–244 (2015).

Luo, M., H.-C. Yan, B. Hu, J.-H. Zhou and C. K. Pang, "A data-driven two-stage maintenance framework for degradation prediction in semiconductor manufacturing industries", Computers & Industrial Engineering **85**, 414–422 (2015).

Matta, A. and F. Simone, "Analysis of two-machine lines with finite buffer, operation-dependent and time-dependent failure modes", International Journal of Production Research **54**, 6, 1850–1862 (2016).

Okamura, H., T. Dohi and K. S. Trivedi, "A refined em algorithm for ph distributions", Performance Evaluation **68**, 10, 938–954 (2011).

Pham, H. and H. Wang, "Imperfect maintenance", European journal of operational research **94**, 3, 425–438 (1996).

Pineau, J., G. Gordon and S. Thrun, "Point-based approximations for fast pomdp solving", in "Proc. Int. Symp. on Robotics Research", (2005).

Puterman, M. L., "Markov decision processes: Discrete stochastic dynamic programming (wiley series in probability and statistics)", (2005).

Rabiner, L. R., "A tutorial on hidden markov models and selected applications in speech recognition", Proceedings of the IEEE **77**, 2, 257–286 (1989).

Sloan, T. W. and J. G. Shanthikumar, "Using in-line equipment condition and yield information for maintenance scheduling and dispatching in semiconductor wafer fabs", IIE transactions **34**, 2, 191–209 (2002).

Tambe, P. P. and M. S. Kulkarni, "A novel approach for production scheduling of a high pressure die casting machine subjected to selective maintenance and a sampling procedure for quality control", International Journal of System Assurance Engineering and Management **5**, 3, 407–426 (2014).

Tan, B. and S. B. Gershwin, "Modelling and analysis of markovian continuous flow systems with a finite buffer", Annals of Operations Research **182**, 1, 5–30 (2011).

Teti, R., K. Jemielniak, G. ODonnell and D. Dornfeld, "Advanced monitoring of machining operations", CIRP Annals-Manufacturing Technology **59**, 2, 717–739 (2010).

Tolio, T., A. Matta and F. Jovane, "A method for performance evaluation of automated flow lines", CIRP Annals **47**, 1, 373–376 (1998).

Wang, B., Y. Lei, N. Li and N. Li, "A hybrid prognostics approach for estimating remaining useful life of rolling element bearings", IEEE Transactions on Reliability (2018).

Wang, H., "A survey of maintenance policies of deteriorating systems", European journal of operational research **139**, 3, 469–489 (2002).

Wang, J., J. Li, J. Arinez and S. Biller, "Quality bottleneck transitions in flexible manufacturing systems with batch productions", IIE Transactions **45**, 2, 190–205 (2013).

Wang, W., "An overview of the recent advances in delay-time-based maintenance modelling", Reliability Engineering & System Safety **106**, 165–178 (2012).

Wu, S. and M. J. Zuo, "Linear and nonlinear preventive maintenance models", IEEE Transactions on Reliability **59**, 1, 242–249 (2010).

Xia, T., X. Jin, L. Xi and J. Ni, "Production-driven opportunistic maintenance for batch production based on mam–apb scheduling", European Journal of Operational Research **240**, 3, 781–790 (2015).

Yan, H., K. Liu, X. Zhang and J. Shi, "Multiple sensor data fusion for degradation modeling and prognostics under multiple operational conditions", IEEE Transactions on Reliability **65**, 3, 1416–1426 (2016).

Yang, S., C. Wu and S. J. Hu, "Modeling and analysis of multi-stage transfer lines with unreliable machines and finite buffers", Annals of Operations Research **93**, 1-4, 405–421 (2000).

Yang, Z., D. Djurdjanovic and J. Ni, "Maintenance scheduling for a manufacturing system of machines with adjustable throughput", IIE transactions **39**, 12, 1111–1125 (2007).

Zhang, F. and A. K. Jardine, "Optimal maintenance models with minimal repair, periodic overhaul and complete renewal", IIE transactions **30**, 12, 1109–1119 (1998).

Zhang, M., O. Gaudoin and M. Xie, "Degradation-based maintenance decision using stochastic filtering for systems under imperfect maintenance", European Journal of Operational Research **245**, 2, 531–541 (2015).

Zhao, C. and J. Li, "Analysis and improvement of multiproduct bernoulli serial lines: Theory and application", IEEE Transactions on Systems, Man, and Cybernetics: Systems **45**, 9, 1218–1230 (2015).

Zhou, B., J. Yu, J. Shao and D. Trentesaux, "Bottleneck-based opportunistic maintenance model for series production systems", Journal of Quality in Maintenance Engineering **21**, 1, 70–88 (2015).

Zuo, M. J., R. Jiang and R. C. Yam, "Approaches for reliability modeling of continuous-state devices", IEEE transactions on reliability **48**, 1, 9–18 (1999).

APPENDIX A

PROOFS OF CHAPTER 3

A.1 Proof of Corollary 1

Induction is used to prove this corollary. First, if only one operating state exists for the machine, then $P = p_{1,F}$.

For a machine with two operating states, the failure state follows $p_{1,F} \leq p_{2,F}$. Denote P in this case as $P(2)$, where the numerical value represents the number of operating states. According to Equation (3.3),

$$\begin{aligned} P(2) &= \frac{1}{\sum_{i=1}^{2} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j} = \frac{1}{\frac{1}{1-p_{1,1}} + \frac{1}{1-p_{1,1}} \frac{p_{1,2}}{1-p_{2,2}}} \\ &= \frac{(1-p_{1,1})(1-p_{2,2})}{1-p_{2,2}+p_{1,2}} = \frac{(p_{1,F}+p_{1,2})p_{2,F}}{p_{2,F}+p_{1,2}} \\ &= \frac{p_{1,2}(p_{2,F}-p_{1,F})}{p_{2,F}+p_{1,2}} + p_{1,F} \\ &\geq p_{1,F}. \end{aligned} \tag{A.1}$$

Assume that the above relationship holds for machines with k operating states. Then the task is to prove that for machine with $k+1$ states, such relationship still holds. The idea is that the first operating state is kept while the remaining k operating states are aggregated into one state U and $P(k)$ could be obtained accordingly. It is assumed that for a machine with k operating states, $p_{2,F} \leq P(k)$. Then we aggregate state U and state 1, and obtain $P(k+1)$. Since $p_{1,F} \leq p_{2,F}$, by using the results for $P(2)$, it follows that $p_{1,F} = min\{p_{1,F}, P(k)\} \leq P(k+1)$. Thus the relationship holds for machine with $k+1$ operating states.

In order to complete the argument, it is necessary to prove that $P(k+1)$ obtained by the above strategy is the same as the one derived using the aggregated probability formula (Equation (3.3))in the integrated model. In other words, $P(k+1) = P$ when there are $k+1$ operating states. Let $p_{2,2} = 1 - P(k)$, then it follows that

$$\begin{aligned} P(k+1) &= \frac{1}{\sum_{i=1}^{2} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j} \\ &= \frac{1}{\frac{1}{1-p_{1,1}} + \frac{1}{1-p_{1,1}} \frac{p_{1,2}}{P(k)}} \\ &= \frac{1}{\frac{1}{1-p_{1,1}} + \frac{p_{1,2}}{1-p_{1,1}} \sum_{i=2}^{k+1} \frac{1}{1-p_{i,i}} \prod_{j=2}^{i-1} \alpha_j} \\ &= \frac{1}{\frac{1}{1-p_{1,1}} + \alpha_1 \sum_{i=2}^{k+1} \frac{1}{1-p_{i,i}} \prod_{j=2}^{i-1} \alpha_j} \\ &= \frac{1}{\sum_{i=1}^{k+1} \frac{1}{1-p_{i,i}} \prod_{j=2}^{i-1} \alpha_j}. \end{aligned} \tag{A.2}$$

It can be found that the above equation is equivalent to Equation (3.3), which concludes the proof. ■

A.2 Proof of Corollary 2

Let Y denote the denominator of P in Equation (3.3). It could be rewritten as follows:

$$\begin{aligned}
Y &= \sum_{i=1}^{W} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j \\
&= \sum_{i=1}^{k-1} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j + \frac{1}{1-p_{k,k}} \prod_{j=1}^{k-1} \alpha_j + \alpha_k \prod_{j=1}^{k-1} \alpha_j \sum_{i=k+1}^{M} \frac{1}{1-p_{i,i}} \prod_{j=k+1}^{i-1} \alpha_j \\
&= \sum_{i=1}^{k-1} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j + \frac{1}{1-p_{k,k}} \prod_{j=1}^{k-1} \alpha_j + \alpha_k \prod_{j=1}^{k-1} \frac{\alpha_j}{p^*_{k+1}} \\
&= \sum_{i=1}^{k-1} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j + \prod_{j=1}^{k-1} \alpha_j \Big(\frac{1}{1-p_{k,k}} + \frac{\alpha_k}{p^*_{k+1}} \Big) \\
&= \sum_{i=1}^{k-1} \frac{1}{1-p_{i,i}} \prod_{j=1}^{i-1} \alpha_j + \prod_{j=1}^{k-1} \frac{\alpha_j}{p^*_{k+1}} + \prod_{j=1}^{k-1} \alpha_j \Big(1 - \frac{p_{k,F}}{p^*_{k+1}} \Big) \frac{1}{1-p_{k,k}}.
\end{aligned} \tag{A.3}$$

The notation of p^*_{k+1} refers to the aggregated failure probability for states $k+1, k+2, \ldots, W$. Therefore, derivatives of Y and P with respect to $p_{k,k}$ could be found in the following equations:

$$\begin{aligned}
&\frac{\partial Y}{\partial p_{k,k}} = \prod_{j=1}^{k-1} \alpha_j \Big(1 - \frac{p_{k,F}}{p^*_{k+1}} \Big) \frac{1}{(1-p_{k,k})^2}, \\
&\frac{\partial P}{\partial p_{k,k}} = -\frac{1}{Y^2} \frac{\partial Y}{\partial p_{k,k}} = -\frac{1}{Y^2} \prod_{j=1}^{k-1} \alpha_j \Big(1 - \frac{p_{k,F}}{p^*_{k+1}} \Big) \frac{1}{(1-p_{k,k})^2}, \\
&\qquad k = 1, 2, \ldots, W.
\end{aligned} \tag{A.4}$$

Based on the proof of Corollary 1, $p_{k,F} \leq p^*_{k+1}$. Therefore, $\frac{\partial P}{\partial p_{k,k}} \leq 0$, which concludes the proof. ■

A.3 Proof of Corollary 3

Based on the closed formula derived in Equation (3.4), Corollary 3 can be proved by taking derivatives of the production rate based on the corresponding probability. For $N = 1$ case, the results can be expressed in the following closed formula explicitly.

$$\frac{\partial PR}{\partial P_2} = -K_1\Big((1-R_2)R_1 + (1-P_1)R_2\Big), \tag{A.5}$$

where $K_1 = \frac{R_1R_2}{(P_1+R_1)(P_2+R_2)^2(R_1+R_2-R_1R_2)} > 0$.

It has been shown that $\frac{\partial PR}{\partial P_2} \leq 0$. Therefore, for $N = 1$, production rate PR is monotonically decreasing with respect to P_2.

Similarly,

$$\frac{\partial PR}{\partial R_2} = K_2\Big[(1-P_1)R_2^2 + (1-R_2)^2R_1^2 + P_1R_2^2(P_2+R_2^2) + R_1R_2(1-R_2)(2+P_2)\Big], \tag{A.6}$$

where $K_2 = \frac{P_2}{(P_1+R_1)(P_2+R_2)^2(R_1+R_2-R_1R_2)^2} > 0$.

Since $\frac{\partial PR}{\partial R_2} \geq 0$, production rate PR is monotonically increasing with respect to R_2, when $N = 1$.

As for the monotonicity of PR with respect to R_1 and P_1, the problem could be transformed to analyzing the Φ function, according to Equation (3.4). The monotonicity of the Φ function with respect to R_1 and P_1 have been discussed in Li and Meerkov (2008). It shows that for $N = 1$, production rate PR is monotonically decreasing with respect to P_1 and increasing with respect to R_1. ■

A.4 Proof of Corollary 4

For each individual machine, when $p_{k,k}$ increases, P is non-increasing, according to Corollary 2. In the two-machine-one-buffer model, according to Corollary 3, the decrease of either P_1 or P_2 will lead to the increase of production rate. Therefore, production rate PR is monotonically increasing with respect to $p^{(i)}_{k,k}$, $k = 1, 2, \ldots, W_i$, $i = 1, 2$.

It is also assumed that $p^{(i)}_{k,F}$ remains constant. Since $p^{(i)}_{k,F}+p^{(i)}_{k,k+1}+p^{(i)}_{k,k} = 1$, production rate (PR) is monotonically decreasing with respect to $p^{(i)}_{k,k+1}$, $k = 1, 2, \ldots, W_i-1$, $i = 1, 2$. ■

A.5 Proof of Theorem 4

First, the following notations are introduced to facilitate the proof.

$$\begin{aligned} \gamma_1 &= R_1 + R_2 - R_1R_2, \\ \gamma_2 &= P_1 + R_1, \\ \gamma_3 &= P_2 + R_2, \end{aligned} \tag{A.7}$$

Then, according to Equations (3.4), (3.5), and (3.6),

$$\begin{aligned} ST_2 &= \frac{P_1 R_2 \theta_2}{\gamma_1 \gamma_2 \gamma_3}, \\ BL_1 &= \frac{P_2 R_1 \theta_1}{\gamma_1 \gamma_2 \gamma_3}. \end{aligned} \tag{A.8}$$

The proof of the theorem could be done by merging the terms in the derivatives into the expressions of desired probabilities, blockage and starvation.

$$\begin{aligned} \frac{\partial PR}{\partial T_{u_1}} &= e_2 (\frac{P_1 \theta_2}{\gamma_2^2 \gamma_1} + \frac{\theta_2}{\gamma_2 \gamma_1})(-P_1^2) \\ &= \frac{R_2}{\gamma_3} (\frac{\gamma_3}{\gamma_2 R_2} ST_2 - \frac{\gamma_3}{P_1 R_2} ST_2)(-P_1^2) \\ &= -P_1(1-e_1)ST_2 + P_1 ST_2 \\ &= P_1 e_1 ST_2. \end{aligned} \tag{A.9}$$

The derivative with respect to the second machine can be obtained as follows:

$$\begin{aligned} \frac{\partial PR}{\partial T_{u_2}} &= (\frac{e_2}{\gamma_3}(\frac{P_1\theta_2}{\gamma_2\gamma_1} - 1) + \frac{P_1 R_1 R_2}{\gamma_1\gamma_2\gamma_3})(-P_2^2) \\ &= (\frac{e_2}{\gamma_3}(\frac{P_1(\gamma_1 - P_2 R_1) - (P_1 + R_1)\gamma_1}{\gamma_2\gamma_1}) + \frac{P_1 P_2 R_2}{\gamma_1\gamma_2\gamma_3})(-P_2^2) \\ &= (\frac{e_2 R_1}{\gamma_3}(\frac{\gamma_1 + P_1 P_2}{\gamma_2\gamma_1}) - \frac{P_1 R_1 R_2}{\gamma_1\gamma_2\gamma_3})P_2^2 \\ &= (\frac{e_2 R_1}{\gamma_3}(\frac{\theta_1 + P_1 R_2 + P_1 P_2}{\gamma_2\gamma_1}) - \frac{P_1 R_1 R_2}{\gamma_1\gamma_2\gamma_3})P_2^2 \\ &= e_2 \frac{R_1 P_2 \theta_1}{\gamma_1\gamma_2\gamma_3} P_2 + (\frac{e_2 R_1}{\gamma_3}(\frac{P_1 R_2 + P_1 P_2}{\gamma_2\gamma_1}) - \frac{P_1 R_1 R_2}{\gamma_1\gamma_2\gamma_3})P_2^2 \\ &= P_2 e_2 BL_1. \end{aligned} \tag{A.10}$$

Therefore, it is equivalent to compare the value of $P_1 e_1 ST_2$ and $P_2 e_2 BL_1$ in contrast to comparing $\frac{\partial PR}{\partial T_{u_1}}$ and $\frac{\partial PR}{\partial T_{u_2}}$. ■

A.6 Proof of Theorem 5

The proof follows the similar idea and uses the same notations as in the proof of Theorem 4. First, we derive the partial derivative of PR with respect to machine 1.

$$\begin{aligned}
-\frac{\partial PR}{\partial T_{d_1}} &= e_2 R_1^2\left(\frac{P_1\theta_2}{\gamma_1\gamma_2^2} + \frac{P_1(\gamma_3-1)\theta_2}{\gamma_1\gamma_2} - \frac{P_1\theta_2(R_2-1)}{\gamma_1^2\gamma_2}\right) \\
&= R_1\frac{R_1}{\gamma_2}\left(e_2\frac{P_1\theta_2}{\gamma_1\gamma_2}\right) + e_2R_1^2\left(\frac{P_1[P_2+(R_2-1)]}{\gamma_1\gamma_2} - \frac{P_1(\gamma_1-P_2R_1)(R_2-1)}{\gamma_1^2\gamma_2}\right) \\
&= R_1e_1ST_2 + e_2R_1^2\left[\frac{P_1P_2}{\gamma_1\gamma_2} + \frac{P_1P_2R_1(R_2-1)}{\gamma_1^2\gamma_2}\right] \\
&= R_1e_1ST_2 + \frac{e_2P_1P_2R_1}{\gamma_1^2\gamma_2}[R_1\gamma_1 + R_1^2(R_2-1)] \\
&= R_1e_1ST_2 + \frac{e_2P_1P_2R_1}{\gamma_1^2\gamma_2}[R_1(R_1+R_2-R_1R_2) + R_1^2(R_2-1)] \\
&= R_1e_1ST_2 + \frac{e_2P_1P_2R_1^2R_2}{\gamma_1^2\gamma_2}.
\end{aligned} \tag{A.11}$$

The derivative with respect to the second machine can be obtained as follows:

$$\begin{aligned}
-\frac{\partial PR}{\partial T_{d_2}} &= R_2^2\{e_2[\frac{P_1(R_1-1)}{\gamma_1\gamma_2} - \frac{P_1(R_1-1)\theta_2}{\gamma_1^2}\gamma_2] - \frac{\frac{P_1\theta_2}{\gamma_2\gamma_1}-1}{\gamma_3} + \frac{R_2(\frac{P_1\theta_2}{\gamma_2\gamma_1}-1)}{\gamma_3^2}\} \\
&= R_2^2\{e_2[\frac{P_1(R_1-1)}{\gamma_1\gamma_2} - \frac{P_1(R_1-1)(\gamma_1-P_2R_1)}{\gamma_2\gamma_1^2}] + (\frac{P_1\theta_2}{\gamma_2\gamma_1}-1)(\frac{R_2}{\gamma_3^2}-\frac{1}{\gamma_3})\} \\
&= e_2R_2^2(\frac{P_1P_2R_1(R_1-1)}{\gamma_1^2\gamma_2}) + (\frac{P_1(\gamma_1-P_2R_1)-(P_1+R_1)\gamma_1}{\gamma_1\gamma_2})(\frac{R_2-(P_2+R_2)}{\gamma_3^2})R_2^2 \\
&= e_2R_2^2(\frac{P_1P_2R_1(R_1-1)}{\gamma_1^2\gamma_2}) + (\frac{(P_1P_2R_1+R_1\gamma_1)P_2}{\gamma_1\gamma_2\gamma_3^2})R_2^2 \\
&= e_2R_2^2(\frac{P_1P_2R_1(R_1-1)}{\gamma_2\gamma_1^2}) + (\frac{(P_1P_2+(\theta_1+P_1R_2)}{\gamma_2\gamma_1\gamma_3^2})P_2R_1R_2^2 \\
&= e_2R_2^2(\frac{P_1P_2R_1(R_1-1)}{\gamma_2\gamma_1^2}) + (\frac{P_1}{\gamma_2\gamma_1\gamma_3} + \frac{\theta_1}{\gamma_2\gamma_1\gamma_3^2})P_2R_1R_2^2 \\
&= e_2R_2^2(\frac{P_1P_2R_1(R_1-1)}{\gamma_2\gamma_1^2}) + e_2\frac{P_1P_2R_1R_2}{\gamma_2\gamma_1} + \frac{P_2R_1\theta_1}{\gamma_2\gamma_1\gamma_3}\frac{R_2}{\gamma_3}R_2 \\
&= e_2(\frac{P_1P_2R_1R_2^2(R_1-1)}{\gamma_2\gamma_1^2} + \frac{P_1P_2R_1R_2}{\gamma_1\gamma_2}) + R_2e_2BL_1 \\
&= \frac{e_2P_1P_2R_1R_2}{\gamma_2\gamma_1^2}[R_2(R_1-1)+\gamma_1] + R_2e_2BL_1 \\
&= \frac{e_2P_1P_2R_1^2R_2}{\gamma_2\gamma_1^2} + R_2e_2BL_1.
\end{aligned} \tag{A.12}$$

Therefore, it is equivalent to compare the value of $R_1 e_1 ST_2$ and $R_2 e_2 BL_1$ in contrast to comparing $\frac{\partial PR}{\partial T_{d_1}}$ and $\frac{\partial PR}{\partial T_{d_2}}$ directly, which concludes the proof. ■

A.7 Proof of Theorem 6

The chain rule shows that:

$$\left|\frac{\partial PR}{\partial p_{kk}}\right| = \left|\frac{\partial PR}{\partial P} \cdot \frac{\partial P}{\partial p_{k,k}}\right|. \tag{A.13}$$

For each machine, $|\frac{\partial PR}{\partial P}|$ is the same for all states. Therefore, the derivatives to identify the state bottleneck can be simplified by using $|\frac{\partial P}{\partial p_{k,k}}|$. Its closed form expression can be obtained directly from Equation (3.3). ■

APPENDIX B

PROOFS OF CHAPTER 5

B.1 Proof of Proposition 1

Let Y denote a random variable, presenting the time staying in the system. Then the moment generating function of this variable, noted as $M_Y(t)$, can be expressed as follows:

$$\begin{aligned} M_Y(t) = \mathbf{E}(e^{tY}) &= \int_{-\infty}^{\infty} e^{ty} f(y) dy \\ &= p \int_0^{\infty} e^{ty_1} e^{-\lambda_1 y_1} dy_1 \int_0^{\infty} \lambda_2 e^{-\lambda_2 y_2} dy_2 \\ &+ (1-p) \int_0^{\infty} e^{ty_2} \lambda_2 e^{-\lambda_2 y_2} dy_2 \\ &= p \frac{\lambda_1}{\lambda_1 - t} \frac{\lambda_2}{\lambda_2 - t} + (1-p) \frac{\lambda_2}{\lambda_2 - t}. \end{aligned} \tag{B.1}$$

Therefore, we can get the three moments of Y as follows:

$$\begin{aligned} m_1 &= \frac{dM_Y(t)}{dt}|_{t=0} = \frac{p}{\lambda_1} + \frac{1}{\lambda_2}, \\ m_2 &= \frac{d^2 M_Y(t)}{dt^2}|_{t=0} = \frac{2p}{\lambda_1^2} + \frac{2p}{\lambda_1 \lambda_2} + \frac{2}{\lambda_2^2}, \\ m_3 &= \frac{d^3 M_Y(t)}{dt^3}|_{t=0} = \frac{6p}{\lambda_1^3} + \frac{6p}{\lambda_1^2 \lambda_2} + \frac{6p}{\lambda_1 \lambda_2^2} + \frac{6}{\lambda_2^3}. \end{aligned} \tag{B.2}$$

Let ϕ_1,ϕ_2 and ϕ_3 be the first three moment of the IG distribution $\mathcal{IG}(\alpha, \beta)$. By using the moment generating function of IG distributions, we can find the value of the ϕ's as follows:

$$\begin{aligned} \phi_1 &= \alpha, \\ \phi_2 &= \frac{\alpha^2(\alpha+\beta)}{\beta}, \\ \phi_3 &= \frac{\alpha^3(\beta^2 + 3\beta\alpha + 3\alpha^2)}{\beta^2}. \end{aligned} \tag{B.3}$$

Therefore, we can solve the equations generated by matching the moments of the IG distribution and the PH distribution, thus determining the three unknown parameters in the PH distribution.

It can be verified that real solutions exist if and only if when $\alpha \geq \beta$. Specifically, when $\alpha = \beta$ and $p = 0$, the phase type distribution obtained is an exponential distribution.

When $\alpha < \beta$, matching three moments is not possible because no feasible solution exists. Rather, we match two moments for the distribution. In this case, three variables are in the PH distribution while only two moment functions are available for the moment matching. In other words, it is possible to have multiple feasible parameter settings to match an IG distribution into the PH distribution. In this section, we only provide one special solution to find the parameters in the PH distribution.

$$\begin{aligned}
&\kappa_5 = 2m_1^2 - m_2, \\
&p = min(1, \frac{\kappa_5 + \sqrt{\kappa_5^2 + \kappa_5}}{m_2}), \\
&\lambda_1 = \lambda_2 = \frac{1+p}{m_1}.
\end{aligned} \tag{B.4}$$

It can be verified that the λ and p values are feasible for all the situations when $\alpha < \beta$. This finishes the proof. ■

B.2 Dynamic and Steady State Equations

Following the similar idea as in Equation (5.8), we can derive the dynamic equations for all the other $X_{s_1 s_2}(h,t)$ as follows:

$$\begin{aligned}
X_{12}(h, t+\Delta t) =& X_{12}(h + (c_2 - c_1)\Delta t, t) e^{-(\lambda_{11}+\lambda_{22})\Delta t} \\
&+ e^{-\lambda_{22}\Delta t} \int_0^{\Delta t} X_{d2}(h + c_2\Delta t - c_1(\Delta t - \tau), t) p_1 \mu_1 \\
&\times e^{-\mu_1 \tau} d\tau + e^{-\lambda_{11}\Delta t} \int_0^{\Delta t} X_{1d}(h + c_2(\Delta t - \tau) \\
&- c_1 \Delta t, t)(1 - p_2)\mu_2 e^{-\mu_2 \tau} d\tau + e^{-\lambda_{11}\Delta t} \int_0^{\Delta t} X_{11}(h \\
&+ (c_2 - c_1)\Delta t, t) \lambda_{11} e^{-\lambda_{11}\tau} d\tau + O(\Delta t^2).
\end{aligned} \tag{B.5}$$

$$\begin{aligned}
X_{1d}(h, t+\Delta t) =& X_{1d}(h - c_1\Delta t, t) e^{-(\lambda_{11}+\mu_2)\Delta t} \\
&+ e^{-\mu_2\Delta t} \int_0^{\Delta t} X_{dd}(h - c_1(\Delta t - \tau), t) p_1 \mu_1 e^{-\mu_1 \tau} d\tau \\
&+ e^{-\lambda_{11}\Delta t} \int_0^{\Delta t} X_{12}(h + c_2(\Delta t - \tau) - c_1 \Delta t, t) \lambda_{22} \\
&e^{-\lambda_{22}\tau} d\tau + O(\Delta t^2).
\end{aligned} \tag{B.6}$$

$$\begin{aligned}
X_{21}(h, t+\Delta t) =& X_{21}(h + (c_2 - c_1)\Delta t, t) e^{-(\lambda_{12}+\lambda_{21})\Delta t} \\
&+ e^{-\lambda_{11}\Delta t} \int_0^{\Delta t} X_{d1}(h + c_2\Delta t - c_1(\Delta t - \tau), t)(1 - p_1) \\
&\times \mu_1 e^{-\mu_1 \tau} d\tau + e^{-\lambda_{11}\Delta t} \int_0^{\Delta t} X_{11}(h + (c_2 - c_1)\Delta t, t) \\
&\times \lambda_{11} e^{-\lambda_{11}\tau} d\tau + e^{-\lambda_{12}\Delta t} \int_0^{\Delta t} X_{2d}(h - c_1\Delta t + \\
&c_2(\Delta t - \tau), t) p_2 \mu_2 e^{-\mu_2 \tau} d\tau + O(\Delta t^2).
\end{aligned} \tag{B.7}$$

$$
\begin{aligned}
X_{22}(h,t+\Delta t) =& X_{22}(h+(c_2-c_1)\Delta t,t)\times \\
& e^{-(\lambda_{12}+\lambda_{21})\Delta t}+e^{-\lambda_{22}\Delta t}\int_0^{\Delta t} X_{12}(h+(c_2-c_1)\Delta t,t) \\
& \times p_1\lambda_{11}e^{-\lambda_{11}\tau}d\tau+e^{-\lambda_{22}\Delta t}\int_0^{\Delta t} X_{d2}(h+c_2\Delta t-c_1 \\
& \times(\Delta t-\tau),t)(1-p_1)\mu_1 e^{-\mu_1\tau}d\tau+e^{-\lambda_{12}\Delta t}\int_0^{\Delta t} X_{21} \\
& \times(h+(c_2-c_1)\Delta t,t)p_1\lambda_{21}e^{-\lambda_{21}\tau}d\tau+e^{-\lambda_{12}\Delta t} \\
& \times\int_0^{\Delta t} X_{2d}(h+c_2(\Delta t-\tau)-c_1\Delta t,t)(1-p_2)\mu_2 e^{-\mu_2\tau}d\tau.
\end{aligned}
\tag{B.8}
$$

The simplified steady state dynamic equation, following the idea of Equation (5.11) and (5.12), is shown as follows:

$$
(c_1-c_2)\frac{\partial X_{12}(h)}{\partial h}=-(\lambda_{11}+\lambda_{22})X_{12}(h)+p_1\mu_1X_{d2}(h)+(1-p_2)\mu_2X_{1d}(h). \tag{B.9}
$$

$$
c_1\frac{\partial X_{1d}(h)}{\partial h}=-(\lambda_{11}+\mu_2)X_{1d}(h)+p_1\mu_1X_{dd}(h)+\lambda_{12}X_{12}(h). \tag{B.10}
$$

$$
\begin{aligned}
(c_1-c_2)\frac{\partial X_{21}(h)}{\partial h} =& -(\lambda_{12}+\lambda_{21})X_{12}(h)+ \\
& \lambda_{11}X_{11}(h)+p_2\mu_2X_{2d}(h)+(1-p_1)\mu_1X_{d1}(h).
\end{aligned}
\tag{B.11}
$$

$$
\begin{aligned}
(c_1-c_2)\frac{\partial X_{22}(h)}{\partial h} =& -(\lambda_{12}+\lambda_{22})X_{12}(h)+\lambda_{11}X_{12}(h) \\
& +\lambda_{21}X_{21}(h)+(1-p_2)\mu_2X_{2d}(h)+(1-p_1)\mu_1X_{d2}(h).
\end{aligned}
\tag{B.12}
$$

$$
c_1\frac{\partial X_{2d}(h)}{\partial h}=-(\lambda_{12}+\mu_2)X_{12}(h)+\lambda_{11}X_{d2}(h)+\lambda_{22}X_{22}(h)+(1-p_1)\mu_1X_{dd}(h). \tag{B.13}
$$

$$
-c_2\frac{\partial X_{d1}(h)}{\partial h}=-(\mu_1+\lambda_{21})X_{d1}(h)+\lambda_{12}X_{21}(h)+p_2\mu_2X_{dd}(h). \tag{B.14}
$$

$$
-c_2\frac{\partial X_{d2}(h)}{\partial h}=-(\mu_1+\lambda_{22})X_{d2}(h)+\lambda_{12}X_{22}(h)+\lambda_{21}X_{d1}(h)+(1-p_2)\mu_2X_{dd}(h). \tag{B.15}
$$

$$
0=-(\mu_1+\mu_2)X_{dd}(h)+\lambda_{12}X_{2d}(h)+\lambda_{22}X_{d2}(h). \tag{B.16}
$$

The following equations determine the boundary conditions.

$$
\begin{aligned}
Y_{12}(0,t+\Delta t) =& Y_{12}(0,t)e^{-(\lambda_{11}+\lambda_{22})\Delta t}+\int_0^{(c_2-c_1)\Delta t} X_{12}(h,t)e^{-(\lambda_{11}+\lambda_{22})\Delta t}dh \\
& +\int_0^{\Delta t} Y_{d2}(0,t)p_1\mu_1e^{-\mu_1\tau}d\tau+\int_0^{\Delta t} Y_{11}(0,t)\lambda_{21}e^{-\lambda_{21}\tau}d\tau+O(\Delta t^2).
\end{aligned}
\tag{B.17}
$$

$$Y_{1d}(0,t) = 0 \tag{B.18}$$

$$\begin{aligned} Y_{21}(0,t+\Delta t) = Y_{21}(0,t)e^{-(\lambda_{12}+\lambda_{21})\Delta t} + \int_0^{(c_2-c_1)\Delta t} X_{21}(h,t)e^{-(\lambda_{12}+\lambda_{21})\Delta t}dh \\ + \int_0^{\Delta t} Y_{11}(0,t)\lambda_{11}e^{-\lambda_{11}\tau}d\tau + \int_0^{\Delta t} Y_{d1}(0,t)(1-p_1)\mu_1 e^{-\mu_1\tau}d\tau + O(\Delta t^2). \end{aligned} \tag{B.19}$$

$$\begin{aligned} Y_{22}(0,t+\Delta t) =& Y_{22}(0,t)e^{-(\lambda_{12}+\lambda_{22})\Delta t} + \int_0^{(c_2-c_1)\Delta t} X_{22}(h,t)e^{-(\lambda_{12}+\lambda_{22})\Delta t}dh \\ &+ \int_0^{\Delta t} Y_{12}(0,t)\lambda_{11}e^{-\lambda_{11}\tau}d\tau + \int_0^{\Delta t} Y_{d2}(0,t)(1-p_1)\mu_1 e^{-\mu_1\tau}d\tau \\ &+ \int_0^{\Delta t} Y_{21}(0,t)\lambda_{21}e^{-\lambda_{21}\tau}d\tau + O(\Delta t^2). \end{aligned} \tag{B.20}$$

$$Y_{2d}(0,t) = 0. \tag{B.21}$$

$$\begin{aligned} Y_{d1}(0,t+\Delta t) =& Y_{d1}(0,t)e^{-(\mu_1+\lambda_{21})\Delta t} + \int_0^{c_2\Delta t} X_{d1}(h,t)e^{-(\mu_1+\lambda_{21})\Delta t}dh + \\ & \int_0^{\Delta t} Y_{21}(0,t)\lambda_{12}e^{-\lambda_{12}\tau}d\tau + \int_0^{\Delta t} Y_{dd}(0,t)p_2\mu_2 e^{-\mu_2\tau}d\tau + O(\Delta t^2). \end{aligned} \tag{B.22}$$

$$\begin{aligned} Y_{d2}(0,t+\Delta t) =& Y_{d2}(0,t)e^{-(\mu_1+\lambda_{22})\Delta t} + \int_0^{c_2\Delta t} X_{22}(h,t)e^{-(\mu_1+\lambda_{22})\Delta t}dh + \\ & \int_0^{\Delta t} Y_{22}(0,t)\lambda_{12}e^{-\lambda_{12}\tau}d\tau + \int_0^{\Delta t} Y_{d1}(0,t)\lambda_{21}e^{-\lambda_{21}\tau}d\tau \\ & + \int_0^{\Delta t} Y_{dd}(0,t)(1-p_2)\mu_2 e^{-\mu_2\tau}d\tau + O(\Delta t^2). \end{aligned} \tag{B.23}$$

$$Y_{dd}(0,t+\Delta t) = Y_{dd}(0,t)e^{-(\mu_1+\mu_2)\Delta t} + \int_0^{\Delta t} Y_{d2}(0,t)\lambda_{22}e^{-\lambda_{22}\tau}d\tau + O(\Delta t^2). \tag{B.24}$$

The simplified steady state dynamic equation, following the idea of Equation (5.11) and (5.12), is shown as follows:

$$(\lambda_{11}+\lambda_{21})Y_{11}(0) = (c_2-c_1)X_{11}(0) + p_1\mu_1 Y_{d1}(0). \tag{B.25}$$

$$(\lambda_{11}+\lambda_{22})Y_{12}(0) = (c_2-c_1)X_{12}(0) + \lambda_{21}Y_{11}(0) + p_1\mu_1 Y_{d2}(0). \tag{B.26}$$

$$(\lambda_{12}+\lambda_{21})Y_{21}(0) = (c_2-c_1)X_{21}(0) + \lambda_{11}Y_{11}(0) + (1-p_1)\mu_1 Y_{d1}(0). \tag{B.27}$$

$$(\lambda_{12}+\lambda_{22})Y_{22}(0) = (c_2-c_1)X_{22}(0) + \lambda_{11}Y_{12}(0) + \lambda_{21}Y_{21}(0) + (1-p_1)\mu_1 Y_{d2}(0). \tag{B.28}$$

$$(\mu_1+\lambda_{21})Y_{d1}(0) = -c_1X_{d1}(0) + \lambda_{12}Y_{21}(0) + p_2\mu_2 Y_{dd}(0). \tag{B.29}$$

$$(\mu_1+\lambda_{22})Y_{d2}(0) = -c_1X_{d2}(0) + \lambda_{12}Y_{22}(0)+ \lambda_{21}Y_{d1}(0) + (1-p_2)\mu_2Y_{dd}(0). \tag{B.30}$$

$$(\mu_1+\mu_2)Y_{dd}(0) = \lambda_{22}Y_{d2}(0). \tag{B.31}$$

The following equations determines the boundary conditions when buffer level is full.

$$Y_{12}(N,t+\Delta t) = Y_{12}(N,t)e^{-(\lambda_{11}+\lambda_{22})\Delta t} + \int_0^{(c_1-c_2)\Delta t} X_{12}(N-h,t)e^{-(\lambda_{11}+\lambda_{22})\Delta t}dh + \int_0^{\Delta t} Y_{2d}(N,t)p_1\mu_1e^{-\mu_1\tau}d\tau + \int_0^{\Delta t} Y_{11}(N,t)\lambda_{21}e^{-\lambda_{21}\tau}d\tau + O(\Delta t^2). \tag{B.32}$$

$$Y_{1d}(N,t+\Delta t) = Y_{1d}(N,t)e^{-(\mu_2+\lambda_{11})\Delta t} + \int_0^{c_2\Delta t} X_{1d}(N,t)e^{-(\mu_2+\lambda_{11})\Delta t}dh + \int_0^{\Delta t} Y_{12}(N,t)\lambda_{22}e^{-\lambda_{22}\tau}d\tau + \int_0^{\Delta t} Y_{dd}(N,t)p_1\mu_1e^{-\mu_1\tau}d\tau + O(\Delta t^2). \tag{B.33}$$

$$Y_{21}(N,t+\Delta t) = Y_{21}(N,t)e^{-(\lambda_{12}+\lambda_{21})\Delta t} + \int_0^{(c_1-c_2)\Delta t} X_{21}(N-h,t)e^{-(\lambda_{12}+\lambda_{21})\Delta t}dh + \int_0^{\Delta t} Y_{11}(N,t)\lambda_{11}e^{-\lambda_{11}\tau}d\tau + \int_0^{\Delta t} Y_{d2}(N,t)p_2\mu_2e^{-\mu_2\tau}d\tau + O(\Delta t^2). \tag{B.34}$$

$$Y_{22}(N,t+\Delta t) = Y_{22}(N,t)e^{-(\lambda_{12}+\lambda_{22})\Delta t} + \int_0^{(c_2-c_1)\Delta t} X_{22}(N-h,t)e^{-(\lambda_{12}+\lambda_{22})\Delta t}dh + \int_0^{\Delta t} Y_{12}(N,t)\lambda_{11}e^{-\lambda_{11}\tau}d\tau + \int_0^{\Delta t} Y_{2d}(N,t)(1-p_2) \times \mu_2e^{-\mu_2\tau}d\tau + \int_0^{\Delta t} Y_{21}(N,t)\lambda_{21}e^{-\lambda_{21}\tau}d\tau + O(\Delta t^2). \tag{B.35}$$

$$Y_{2d}(N,t+\Delta t) = Y_{2d}(N,t)e^{-(\mu_2+\lambda_{22})\Delta t} + \int_0^{c_2\Delta t} X_{22}(N-h,t)e^{-(\mu_1+\lambda_{22})\Delta t}dh + \int_0^{\Delta t} Y_{1d}(N,t) \times \lambda_{11}e^{-\lambda_{11}\tau}d\tau + \int_0^{\Delta t} Y_{22}(N,t)\lambda_{22}e^{-\lambda_{22}\tau}d\tau + \int_0^{\Delta t} Y_{dd}(N,t)(1-p_1)\mu_1e^{-\mu_1\tau}d\tau + O(\Delta t^2). \tag{B.36}$$

$$Y_{d1}(N,t) = 0. \tag{B.37}$$

$$Y_{d2}(N,t) = 0. \tag{B.38}$$

$$Y_{dd}(N, t+\Delta t) = Y_{dd}(N,t)e^{-(\mu_1+\mu_2)\Delta t} + \int_0^{\Delta t} Y_{2d}(N,t)\lambda_{22}e^{-\lambda_{22}\tau}d\tau + O(\Delta t^2). \quad \text{(B.39)}$$

The simplified steady state dynamic equation, following the idea of Equation (5.11) and (5.12), is shown as follows:

$$(\lambda_{11}+\lambda_{21})Y_{11}(N) = (c_1-c_2)X_{11}(N) + p_2\mu_2 Y_{1d}(N). \quad \text{(B.40)}$$

$$(\lambda_{11}+\lambda_{22})Y_{12}(N) = (c_1-c_2)X_{12}(N) + \lambda_{21}Y_{11}(N) + (1-p_2)\mu_2 Y_{1d}(N). \quad \text{(B.41)}$$

$$(\lambda_{11}+\mu_2)Y_{1d}(N) = c_1 X_{1d}(N) + \lambda_{22}Y_{12}(N) + p_1\mu_1 Y_{dd}(N). \quad \text{(B.42)}$$

$$(\lambda_{12}+\lambda_{21})Y_{21}(N) = (c_1-c_2)X_{21}(N) + \lambda_{22}Y_{12}(N) + p_1\mu_1 Y_{dd}(N). \quad \text{(B.43)}$$

$$(\lambda_{12}+\lambda_{22})Y_{22}(N) = (c_1-c_2)X_{22}(N) + \lambda_{11}Y_{12}(N) + \lambda_{21}Y_{21}(N) + (1-p_2)\mu_2 Y_{2d}(N). \quad \text{(B.44)}$$

$$(\lambda_{12}+\lambda_{22})Y_{2d}(N) = c_1 X_{2d}(N) + \lambda_{22}Y_{22}(N) + (1-p_1)\mu_1 Y_{dd}(N). \quad \text{(B.45)}$$

$$(\mu_1+\mu_2)Y_{dd}(N) = \lambda_{12}Y_{2d}(N). \quad \text{(B.46)}$$

■