Event-based Modeling and Control of An Advanced Manufacturing System

A Dissertation Presented

by

Yang Li

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

August 2014

Stony Brook University

The Graduate School

Yang Li

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Qing Chang, Dissertation Advisor,
Assistant Professor, Department of Mechanical Engineering

Qiaode Jeff Ge, Chairperson of Defense,
Professor, Department of Mechanical Engineering

Shikui Chen, Committee Member,
Assistant Professor, Department of Mechanical Engineering

Jorge Arinez, Outsider Member,
Manufacturing Systems Research Lab, General Motors R&D

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Event-based Modeling and Control of An Advanced Manufacturing System

by

Yang Li

Doctor of Philosophy

in

Mechanical Engineering

Stony Brook University

2014

Although the widely application of advanced IT technology has enabled production information to become increasingly transparent, detailed and real-time, the utilization of the information for system modeling and control remains largely unexplored. In fact, instantaneously transforming information gathered from a vast array of sources into useful knowledge for effective decision making has been identified as one of the six grand challenges in the vision for manufacturing 2020 and beyond. It is necessary to have a real-time integrated system modeling and control method, which utilizes the production information to quickly respond to unpredictable disturbance to manufacturing system and ensure smooth production and high productivity. This dissertation is devoted to this end.

In this dissertation, the impact of disruption events on system productivity in both serial and parallel production lines is quantitatively evaluated. Built on the analysis, an event-based modeling (EBM) approach is developed to estimate the systematic impact of individual machine and supporting activity (e.g. material handling, quality inspection and maintenance). EBM instantaneously captures the system dynamics using distributed sensor information. To evaluate the performance of a production line segment, standalone throughput (SAT) definition and an event-based estimation method are developed. A market demand driven system model is established to unify the analysis of system productivity and market demand satisfaction. A supervisory control algorithm is developed to continuously improve system productivity and market demand satisfaction (MDS). A Markov decision process (MDP) model is built to assist the control decision making in the supervisory control algorithm.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my most sincere gratitude to my advisor, Professor Qing Chang, for her continuous support, guidance and encouragement throughout my graduate study. I have enjoyed this research under her experienced and visionary supervision. Without her, I would never have had the chance to begin, continuous, and complete my research. I would also like to thank Dr. Stephan Biller and Dr. Guoxian Xiao from General Electric and General Motors for their cooperation.

I would also like to thank my committee members, Dr. Qiaode Jeffrey Ge, Dr. Shikui Chen and Dr. Jorge Arinez, for their support and valuable comments to my dissertation.

I greatly appreciate my friends from Intelligent Systems Laboratory, Stony Brook University, especially Michael Brundage and Shiyao Wang, as well as my other friends, Dr. Xiangyun Li, Dr. Xu Zhong, Dr. Hai Wei, Peng Li, Jinjing Xie, Xin Ge and Barray Chuang.

Finally, I would like to thank my family, especially my parents and my wife, Maomao Feng. Their understanding, unconditional support and encouragement have given me the confidence to face challenges and fight through difficulties over the years.

Chapter 1

INTRODUCTION

1.1 Motivation

In the globalized and interconnected market, demand fluctuation along with the requirements of high product quality, low cost, short leading time and high customization has leaded to an increase in the complexity of manufacturing systems [1]. A production system must quickly ramp up the newly developed technologies, new tools and equipment if it is to meet the various challenges. However, new technology insertions and frequent changes in its manufacturing processes also result in a production system usually staying in a transient state which is unstable, more unpredictable and dynamic comparing with a steady state production system. This normally leads to a low productivity and quality. To improve system performance and productivity, it is necessary to have a real-time modeling and control methodology to quickly realize system dynamics under disruption events and make control decisions to continuously improve system performance.

Production system modeling and control have been studied extensively during the past fifty years and numerous results have been presented and successfully applied in system planning, designing and controlling [2-4]. Most of these studies focus on steady state analysis based on long-term system performance measurements and assuming production systems being static. However, disruption events may derive the system deviate from the optimal design and expected target. As a result, it is not uncommon to observer that a production system cannot achieve its predicted performance.

Although the advancement in IT technology has enabled the production information to become increasingly transient, detailed and in real-time, the utilization of the information for real-time modeling and control has been largely lagged behind [4-7]. The information is either used to calculate the average system parameters, such as mean time to repair (MTTR) and mean time between failures (MTBF), or estimate the isolated performance indicators or matrices, such as throughput and machine idling time. It is mostly used for isolated local control, but is not connected to efficiently collect, disseminate, and interpret the information at an overall system level. To fill the gap, a system level integrated modeling approach is needed to capture the production system dynamic processes and evaluate production system performance fully utilizing the rich real-time information. And an optimal control methodology is necessary to continuously improve system performance based on the real-time analysis. This dissertation is devoted to this end.

1.2 Problem Addressed and Solutions

In this work, an event-based methodology is developed to model and control a multi-stage production system in real-time.

Firstly, the impacts of disruption events are quantified. Disruption events are arguably the single most significant reason to cause system inefficiency. Accurate estimation of the impacts of disruption events is of significance to find the true causes (e.g. machines or supporting activities) of system inefficiency and provide guidance for system control. In this work, we develop a data-driven method to quantify the impacts of disruption events in a general production line in real-time based on online information such as machine random failures, buffer levels, etc. Our analysis indicates that not all the production loss at any single machine is permanent. Only the production loss happened at the last slowest machine[1] contributes to the permanent production loss of the system. Therefore, the impacts of a disruption event can be quantified as the production loss at the last slowest machine caused by the event, which is denoted as the permanent production loss caused by the event.

Based on the analysis, we develop an event-based modeling (EBM) approach to capture system dynamics in real-time. EBM naturally integrates the two most significant system concerns: system capacity and production loss and quantifies the impacts of machines and supporting activities (e.g. material handling, maintenance, etc.) to system productivity with a single unified index, i.e. permanent production loss. And the permanent production loss caused by a machine or a supporting activity is proved to be the summation of the permanent production loss caused by the disruption events resulted from the machine or supporting activity.

It is not uncommon that a complex multistage manufacturing system is segmented into several subsystems for efficient local management. It is important to evaluate the performance of each subsystem to improve overall system productivity. Therefore, we introduce the standalone throughput (SAT) of a production line segment to quantify the capability of a subsystem, where SAT of a production line segment is defined as the productivity of the line segment while not being affected by its upstream and downstream machines. A data-driven method based on online production information is developed to estimate the SAT of a line segment in real-time.

[1] The slowest machines are defined as the machines with the largest machine cycle time and the last slowest machine denotes the slowest machine that is closest to the end-of-line machine.

To improve system productivity and market demand satisfaction, a supervisory control algorithm is developed. The supervisory control algorithm is a feedback control process by periodically identification and mitigation of machine capacity bottleneck (MC-BN) and machine failure bottleneck (MF-BN). Real-time EBM based and stochastic EBM based methods are developed to identify MC-BNs and MF-BNs. Markov decision model is established to decide which method should be used at different system state. The numerical analysis has proved that the proposed supervisory control algorithm will lead to the largest improvement in productivity compared with other control algorithms [8].

The remainder of this dissertation is structured as follows: Chapter 2 introduces the literature review; Chapter 3 discusses the methods to evaluate the systematic impacts of disruption events in different production line configurations; an integrated system modeling approach, i.e. EBM, is developed in Chapter 4; SAT definition and an estimation method are discussed in Chapter 5; Chapter 6 develops a market demand supervisory control algorithm in a production system; Chapter 7 will discuss an optimal control in a production system; the conclusion and future work are discussed in Chapter 8.

Chapter 2

LITERATURE REVIEW

In this chapter, the relevant literature is discussed. Section 2.1 reviews available literature on how disruption events impact system dynamics. Section 2.2 introduces the literature about system modeling and analysis. Section 2.3 reviews the existing works about system improvement and control. Section 2.4 summaries the conclusions.

2.1 Impacts of Disruption Events

The literature dealing with the analysis of impacts of disruption events is relatively limited. Most of the existing analysis focuses on the evaluation of disruption events impacts with stochastic methods or simulation [3, 9-13]. Recently, a data-driven method has been developed to quantify the impact of disruption events to system productivity in serial production systems [3, 7, 10].

Paper [13] investigates the impact of disruptions on the operation of a supply chain. A network-based modelling methodology is presented to determine the impact of a disruption or change to supply chain and the whole system. The modeling approach, i.e. disruption analysis network (DA-NET), models how disruption or change disseminates through a supply chain system and quantifies the impact of the attributes.

Paper [12] presents a disruption recovery model for a single stage production and inventory system. The model is formulated as a constrained non-linear optimization program and solved using both a heuristic method and an evolutionary algorithm. It is proved that the heuristic method is able to accurately solve the model with significantly less time compared to the evolutionary algorithm. And it is shown that the optimal recovery schedule is dependent on the shortage cost parameters and the extent of the disruption.

Paper [9] discusses the disruption management in industries. The authors proposed an approach to minimize the impact of disrupting events on the whole system, which is based on an analysis of disrupting events and the characterization of the recovery process and on a cooperative repair method. The method is based on a cooperative distributed problem solving approach supported by a multi-agent system framework.

Paper [11] considers the propagation of disruption events and the capability of a system that can recover from the events. Two important measures of resilience (i.e. throughput settling time (TST) and overtime to recover (OTTR)) for a general serial-parallel production system with finite buffers are defined and analyzed. The exact analysis for TST and OTTR in a two stage system is derived and an approximation method based on system decomposition is developed to analyze the TST and OTTR in general multi-stages systems. The paper concludes that parallel systems are more resilience than serial systems and larger buffers in serial-parallel system can make system more resilience.

Event-based analysis evaluate the impacts of disruption events to system productivity based on measured online information such as buffer levels, machine random downtime, and machine starvation and blockage. Paper [10] discusses the impacts of single isolated disruption event to system productivity in a production line with single slowest machine. It has been proved that the impact of an isolated disruption event only apparent in a relatively long run if the duration exceeds a threshold that the slowest machine is starved, blocked or down because of the event. Paper [7] further extends the analysis to a generic scenario where disruption events occur concurrently. The analysis unified the analysis to both isolated disruption event and multiple disruption events. It indicates that any stoppage event at the unique slowest machine, due to starvation, blockage or machine failure, incurs production loss to the system that is impossible to recover in the future.

2.2 System Modeling and Analysis

A production system is a complex system. The complexity of a production system comes from the complex process line layout, randomness in the process, including machine random breakdowns, processing time variation, and the random number of parts produced by the system. Extensive works have been done in the area of production system modeling and analysis, which are important for design, operation and management of production systems [14-17]. Studies can be grouped into analytical methods and simulation-based methods. In this section, we will introduce the simulation models and analytical models.

2.2.1 Simulation Models

Simulation models are widely adopted to evaluate performance of complex manufacturing systems [14, 18-23]. Data-driven modeling and simulation is a method that allows users to create and run simulation model without doing programming [24]. In this method, the information specifying the model must be presented to generate the appropriate models. It enables users, rather than operation specialists, to prepare and run simulations, and achieve the ability to reconfigure the models for assessing changed or alternative scenarios [14, 18]. Paper [21] presents a conceptual framework to generate a WITNESS simulation model from graph-based process plans and resource configurations for a job shop manufacturing system. Ford Motor Company developed an assembly simulation tool that allows semi-automatic model generation which is built from data in an EXCEL spreadsheet. However, data-driven simulation is not a replacement for the general purpose simulation tool but an assembly simulation tool that is valuable in certain circumstances with a relatively quicker model generation and minimum simulation expertise.

Event relationship graph (ERG) is a simulation model technique for a discrete-event system. Paper [19] presents a linear programming formulation for a single-server queuing system and the solutions represented the dynamic system trajectory. Constraints of the mathematical formulation are derived from the edges of the ERG. Event relationship graph modeling has certain advantages in terms of simplicity and efficiency in simulation. However, for complicated DEDS with random events, such as random machine failures in production line, the linear programming formulation is very difficult to apply.

Parallel and distributed simulation is another approach for simulating manufacturing systems [20, 23]. The method has advantages in terms of computing efficiency. However, simulation tasks need to be divided into many sub-tasks to be executed concurrently. The method is very suitable for loosely coupled system with weak interactions. It cannot be directly applied to closely coupled systems. In general, with increasing complexity of a manufacturing system, modeling and simulation of a production process becomes more challenging and requires more expert knowledge and effort.

2.2.2 Analytical Models

Based on the different line configurations, the analytical modeling and analysis of production systems in large-volume manufacturing can be categorized into three: serial production lines, assembly systems and parallel systems. Additionally, the transient analysis of production line performance is also discussed in this section.

- Serial Line

For analytical modeling approach, exact analytical results only exist for the two-machine one buffer system and the system with infinite buffer capacity or without buffers. For longer systems, approximation methods based on the results of analysis of two machines system are investigated [16].

Papers [15, 25, 26] develop a decomposition method to analyze serial production lines. The decomposition method is developed based on the representation of $M-1$ buffer systems, in which, each buffer system includes two machines and one buffer. In each two-machine, one-buffer system, two pseudo-machines are introduced to model the upstream and downstream lines of the buffer. For example, in the *i*th two-machine, one-buffer system, pseudo-machine $M_u(i)$ models the line upstream of the buffer b_i, and $M_D(i)$ models the line downstream from the buffer $\mathrm{b_i}$. The parameters of the pseudo-machines are determined by solving a group of equations based on the conservation of flow, flow rate-idle time, and resumption of flow relationships. These equations together with boundary conditions provide a total of $4(M-1)$ equations with $4(M-1)$ unknowns. Computational algorithms have been introduced to solve the equations, such as Dallery-David-Xie (DDX) algorithms and the accelerated DDX (ADDX) algorithm [27, 28]. In most cases, the ADDX algorithms can converge and provide a faster speed and more accurate estimation [29].

References [30, 31] develop an aggregation method consisting of forward and backward aggregation. In the backward aggregation, the last two machines and the buffer between them are aggregated into a single machine and then the new machine is aggregated with the machine before it. The aggregation is repeated until all the machines and buffers between them are aggregated into a single machine. In forward aggregation, the aggregation process begins from the first two machines and the buffer in between and is repeated until all machines and buffers becomes a single machine. The system throughput is estimated as the throughput of the single machine in backward or forward aggregation approach. The advantage of the method is that all the aggregation procedures have been analytically proved to be convergent and the accuracy of the aggregation procedures has been determined either analytically or numerically. However, the method is developed based on assumptions that machines have either Bernoulli, geometric or exponential machine reliability model. It is hard to extend the method to production systems with different machine reliability distribution.

References [32, 33] investigate non-exponential distribution machine systems. Decomposition methods based on the Markov chain approach have been applied to approximate the performance of the system. However, such approaches are typically computationally intensive and are difficult to extend to complex systems.

- Assembly Lines

Approximate solutions for assembly systems can be obtained by generalizing the results of throughput analysis of serial lines. The aggregation and decomposition methods presented in serial lines can be extended to the analysis of assembly systems.

References [15, 34] develop efficient decomposition methods for calculating the throughput of tree structured assembly system. The method decomposes the assembly/disassembly system into two-machine, one buffer lines. Pseudo-machines are used to model the part of the line upstream of the buffer and the part of the line downstream from the buffer. Similarly, based on the equations of the conservation of flow, flow rate-idle time, and resumption of flow relationships, a total of $4(M-1)$ equations with $4(M-1)$ unknowns can be obtained. The same computational algorithms, like DDX, can be used to solve the decomposition equations.

Similar to the serial line case, an aggregation method was developed in [35-37] to study assembly systems. The idea behind the approximation is as follows. Firstly, a virtual serial line consisting of one component line and the assembly line are considered. In the virtual serial line, the first machine m_{01} in the assembly line is modified such that the impacts from other component lines are included. Then, using a serial line evaluation method, the probability for the buffer before machine m_{01} not being empty can be calculated. Now consider a virtual serial line composed of another components line and the assembly line, where the first machine in an assembly line is again modified by considering the probability that the buffer before machine m_{01} is empty. Then the probability that the buffer is empty is calculated. The probability is used for the second iteration and continued. The process is repeated until the iterations are converged. The result is the estimates of the production rate.

Paper [38] investigates an unbalanced, continuous flow assembly system with two inputs. A new system, which is identical to the original, is defined with the assumption that the assembly machine is completely reliable. The throughput of the original system is approximated using the availability of the new defined system multiplied the availability of the assembly machine and its speed.

- Parallel Lines

Production lines with parallel structures are widely used in many production systems to achieve a greater productivity or reliability. A parallel structure can be split into several sub-lines in parallel. In case one machine in a parallel structure fails, parts can still go through other sub-lines to keep production moving, but with a slower rate. Most analyses study parallel lines by equivalence, i.e. aggregating parallel machines into an equivalent single machine.

Paper [29] considers series-parallel flow lines where each stage consists of multiple parallel machines and finite buffers. The multiple machines in each stage are approximated by an equivalent single machine. The system performance is analyzed using equivalent machine and ADDX algorithm is used to estimate the throughput. Similar ideas are used in References [39] and [40], where [39] consider inhomogeneous lines and [40] extends the study to non-identical parallel machine case.

Paper [37] proposes an overlapping decomposition method to estimate the throughput of parallel lines which allows multiple machines and buffers in sub-lines in parallel structures. The method takes into consideration of the effects from other part of a parallel production line by manipulating the starvation and blockage incidents at the first and last machine in each sub-line as downtime events, and a recursive procedure is applied to estimate the production rate of the system.

- Transient Analysis of A Production System

Transient characteristics have significant manufacturing implications [41]. Production transients characterize the process of reaching the steady state system output. System often operates at transient regimes. For example, in the process of continuous improvement, the machine parameters change every day or even every shift. The steady state can hardly be reached. Before the steady can be reached, the system can suffer significant production loss. Therefore, accurate evaluation of transients in production systems is of practical importance.

The transient behavior of the transfer line after a sudden station breakdown can be approximated using metamodels in the form of first-order continuous exponential delays function [42]. Earlier works in References [43-45] have studied the transient behavior of communication networks. The main focus of the research is on the transient evolution of the probability distribution of the buffer levels to its stationary distribution. Papers [42, 46, 47] further extend the results to transfer lines with Bernoulli machines and characterizes of the transients in production rate and work-in-process inventory. In paper [48], the second largest eigenvalue of the transient matrices is used to characterize the transients in geometric production lines. Markov chain approach is adopted to develop the transition matrices. However, the large dimensionality of the transition matrices impedes the method to be applied to systems with more than two machines.

2.3 Production System Control and Improvement

The control and improvement of a production system is another important topic in the analysis of production system. In the past two decades, a lot of literature have been devoted to the topic [49-51].

In [52-59], the continuous improvement of Bernoulli and Exponential lines has been addressed. The continuous improvement is divided into two categories: constrained improvement and unconstrained improvement. In constrained improvement, limited resources, such as buffer capacities, work force, etc., are re-allocated in order to improve the system production. Criteria are presented to determine if a system is improvable and to provide a characterization of unimprovable allocations. In unconstrained improvement, system is improved through the identification and elimination of bottlenecks by allocating additional resources, where a bottleneck is defined as a machine whose performance impedes the system output in the strongest manner. A bottleneck identification tool based on machine blockages and starvations is also developed.

Paper [60] studies the performance of the Kanban, minimal blocking, basestock, CONWIP, and hybrid Kanban-CONWIP control policies in a serial production line based on simulations analysis. Two main performance measures, which are service level and the amount of work-in-progress (WIP), are measured and compared in both constant and changing demand rates cases. The results indicate that the best parameter choices for the hybrid policy are lower than the other three policies while the same service level is maintained and the WIP difference among the policies grows as the demands on service level increase. The study also finds that the CONWIP and hybrid policies give significantly better response to changes in the demand rate.

Paper [61] discusses the design and analysis of Kanban-controlled pull-system with exponential machines and finite buffers. The closed formulas are derived for system performance matrices, which are the customer service level, finished goods inventory, and the release throttling. The system-theoretic properties are investigated and methods to evaluate the lean number of Kanban for a desired level of customer service are derived.

Paper [62] addresses the optimal control of production rate in a failure prone production system in order to minimize the discounted inventory cost. The paper formulates the problem as an optimal controlling of a continuous-time system with jump Markov disturbances and an infinite horizon discounted cost criterion. The analysis indicates that the optimal solution is simply characterized by a certain critical number, i.e. optimal inventory level. The production system should produce parts at their maximum rate if the current inventory level is less than the optimal inventory level; the production system should not produce any parts if the current inventory level is higher than the optimal inventory level; the production system should produce parts exactly equal to demand if the current inventory level is equal to the optimal inventory level.

Paper [63] studies the optimal control of a single stage periodic-review inventory production system with state-dependent random yield. The system is subject to stochastic demand and its order size is determined based on an order-up-to logic. The paper develops an approximation to estimate the near optimal order-up-to level and maintenance interval. Three methods are developed to decompose the problem and compute the policies parameters either sequentially or separately. The approximation method and the three methods are compared and the result indicates that the approximation method can lead to the best system performance as well as lowest system cost.

Markov decision theory has been widely used for production system control. In [64], the control of preventive maintenance is studied. The paper uses the embedding technique from Markov decision theory to find a class of control limit policies. Paper [65] formulates an integrated decisions of maintenance and production using a Markov Decision Process. A double-threshold policy is presented and exact and approximate methods for evaluating the performance of this policy and computing tis optimal parameters are derived. In [66], the authors studied the optimal policy for modular product reassembly within a remanufacturing setting where a firm receives production returns with variable quality and reassembles products of multiple classes to customer orders. The problem is formulated as a Markov decision process and the structure of the optimal control policy is proved to be a state-dependent threshold-based control rule.

2.4 Conclusion

In spite of those efforts, the study of real-time modeling and control of a production system is still limited. Most of the system modeling analysis focuses on using stochastic method to estimate the system performance in steady-state. Although there are some studies discussing the real-time analysis of system performance, they are still too specific to be applied in general production systems. An integrated real-time system modeling approach has not been well developed. Additionally, it can be observed that most existing control algorithms are developed based on certain heuristic standards or stochastic analysis by assuming that systems are in steady-state. There is no existing scheme that integrates the real-time modeling and control of a production system. This dissertation is devoted to this end.

Chapter 3

REAL-TIME ANALYSIS OF DISRUPTION EVENTS IN SERIAL PRODUCTION SYSTEMS

3.1 Introduction

Accurate estimation of the impact of disruption events, especially those that cause the most production loss, is of significant importance for deciding where to allocate limited resources in a multi-stage manufacturing systems as well as the establishing of a real-time system modeling method. In this section, we focus on the quantitative evaluation of the impact of disruption events in serial production lines. We begin with the analysis of the impact of disruption events in a serial production line with single slowest machine and then extend to more generic scenarios where multiple slowest machines exist.

3.2 System Description, Assumptions and Notations

A serial manufacturing system typically consists of a series of machines separated by finite buffers. A group of operations are performed on unfinished products based on a predetermined sequence order. Additionally, supporting activities, such as maintenance and material handling, are also integral components of the overall production system. Figure 3.1 shows a typical example of the interdependency and dynamic interactions of machines and supporting activities. The smooth and efficient operation of the production line not only relies on the timely completion of required production operations at each machine and the coordination among multiple machines, but also the smooth coordination between machines and supporting activities.

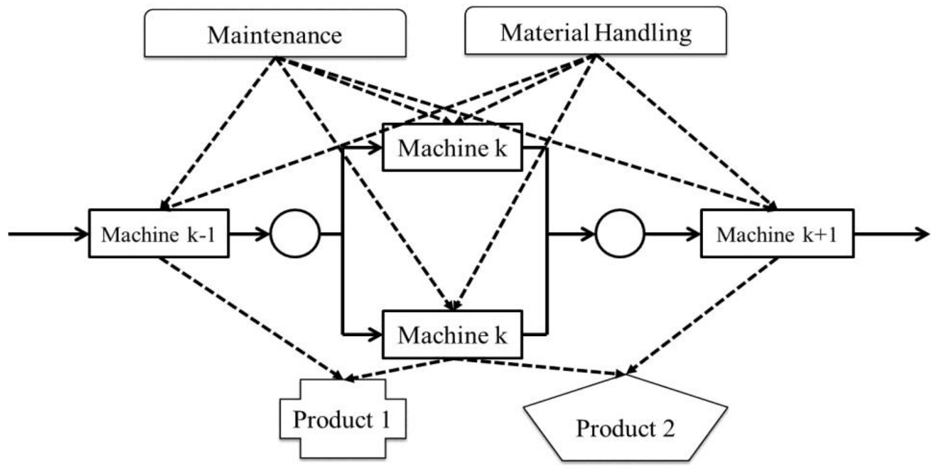


Figure 3.1 Demonstration of the interdependency and dynamic interactions among production and supporting activities

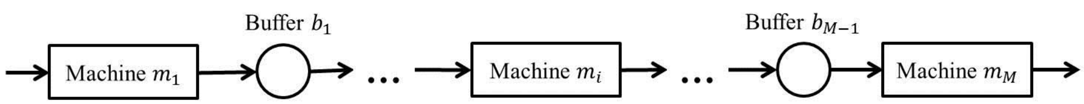


Figure 3.2 A serial production line consisting of M machines and M-1 buffers

In this dissertation, a simplified serial production line consisting of M machines and $M-1$ buffers shown in Figure 3.2 is considered. The following definitions and assumptions on an integrated battery production system are adopted.

1) A buffer $b_i, i = 1, \dots, M-1$, refers to a material handling device (e.g., boxes or lift forks, conveyors, etc.), which stores and moves unfinished jobs from upstream machine to downstream machine. Each buffer has a finite buffer capacity B_i.
2) $b_i(t)$ denotes the buffer level of buffer b_i at time t.
3) A production line consists of both manual and automatic machines, where m_i denotes the i^{th} machine in the production line. Automatic machines usually have constant processing speeds; while the processing speeds of manual machines could vary. For both scenarios, we can always find the smallest processing time of a machine m_i, which is referred to as the base cycle time T_i. For each machine, any production cycle that is longer than the base cycle

time will be treated as a special random downtime event, referred to as over-cycle event. A machine can be down because of tool wear, power outages, as well as over-cycle.

4) The rated speed of a machine m_i is $\frac{1}{T_i}$ and the instantaneous speed of machine m_i at time t is denoted as $s_i(t)$. A machine can produce product at a speed not greater than its rated speed, i.e. $s_i(t) \leq \frac{1}{T_i}$.
5) A machine $m_{M_k^*}$ is defined as the slowest machine if it has the largest base cycle time among all the machines, i.e. $M_k^* = \arg\max_{i=1,\dots,M} (T_i), 1 \leq k \leq M$. There can be one or multiple slowest machines in a system. m_{M^*} denotes the slowest machine that is closest to the end-of-line machine m_M. It is also denoted as the last slowest machine.
6) A machine is starved if the machine is up and its immediate upstream buffer is empty. A machine is blocked if the machine is up, its immediate downstream buffer is full and the immediate downstream machine does not take a product from the buffer. A machine cannot be down when it is starved or blocked.
7) $X(T) = \int_0^T s_M(t)dt$ denotes the output of the production system during time period of $(0, T]$.
8) Supporting activities $m_{M+1}, \dots, m_{M+N}$, (e.g., maintenance, material handling, etc.) are referred to as processes which provide necessary supports to ensure the proper functioning of various machines. Any supporting activities that fail to finish their tasks timely will result in the stoppage of the corresponding machines. For example, if material handling staff fails to deliver parts to a machine on time, the machine will be down because of material shortage. We define those events as supporting activity failure event. Specifically, m_{M+1} refers to material handling in this paper.
9) Quality inspection equipment, such as sensors, inspection machines, etc., is installed to detect and trace the root causes of product defects. In battery production lines, products that are

detected with quality defects will be scrapped directly. Anytime there is a scrapped part at a check point (a sensor or an inspection machine), all the operation time of the upstream machines devoted to the scrapped product can be considered as downtime caused by the quality failure. We define those downtime events as quality failure events.

10) A disruption event $\vec{e}_i$ refers to a machine random downtime event, over-cycle event, quality failure event or supporting activity failure event. $\vec{e}_i = \left(m_{i_1}, m_{i_2}, t_i, d_i\right), 1 \leq i_1 \leq M + N, 1 \leq i_2 \leq M$, denotes the failure of m_{i_1}results in m_{i_2} being down at time t_i for d_i. If $i_1 = i_2$, $\vec{e}_i = \left(m_{i_1}, m_{i_2}, t_i, d_i\right)$ denotes a machine random downtime event, over-cycle event, or quality failure event at machine $m_{i_1}, 1 \leq i_1 \leq M$ which begins at t_i and lasts for a time period of d_i. On the other hand, if $i_1 \neq i_2$, $\vec{e}_i = \left(m_{i_1}, m_{i_2}, t_i, d_i\right)$ denotes a quality failure event ($1 \leq i_1 \leq M$) or supporting activity failure event ($M + 1 \leq i_1 \leq M + N$) caused by m_{i_1} which results in machine m_{i_2} being down at time t_i for a time period of d_i. E denotes a sequence of disruption events, i.e. $E = [\vec{e}_1, \ldots, \vec{e}_n]$.

11) $TL(\vec{e}_i)$ denotes permanent production time loss caused by disruption event $\vec{e}_i$; $PL(\vec{e}_i)$ denotes the permanent production loss attributed to disruption event $\vec{e}_i$; PL_i denotes permanent production loss attributed to $m_i, 1 \leq i \leq M + N$.

3.3 Dynamics in Serial Production Lines with Single Slowest Machines

In this section, we will start with the concept of the opportunity window. Then we will give the expression of permanent production time loss of each disruption event, where the permanent production time loss denotes the system production time loss caused by a disruption event that is impossible to recover in the future. The opportunity window $W(m_{i_2})$ is defined as the longest possible downtime duration of machine m_{i_2} that does not result in permanent production loss in the system [7, 10]. It can be expressed as

$$W(m_{i_2}) = \sup\{d \geq 0: \text{s.t.} \exists T^*(d),\ \int_0^T s_M(t; \vec{e}) = \int_0^T s_M(t)dt\,,\ \forall T \geq T^*(d)\}$$

where $\int_0^T s_M(t;\ \vec{e})dt$ and $\int_0^T s_M(t)dt$ are the production volume of the end-of-line machine S_M, whose output is defined as the output of the system, at time T, with and without inserted downtime event $\vec{e} = (m_{i_1}, m_{i_2}, t_i, d_i)$.

If all the machines operate perfectly, the opportunity window of any machine m_{i_2} is the time it takes for the buffers between machines m_{i_2} and m_{M^*} to become empty if machine m_{i_2} in upstream of machine m_{M^*} or full if machine m_{i_2} is downstream of machine m_{M^*}. And the opportunity window for machine m_{M^*} is zero. Therefore, we have

$$W(m_{i_2}) = \begin{cases} T_{M^*} \sum_{k=i_2}^{M^*-1} b_k(t) & i_2 < M^* \\ 0 & i_2 = M^* \\ T_{M^*} \sum_{k=M^*}^{i_2-1} B_k - b_k(t) & i_2 > M^* \end{cases}$$

Not every downtime event can cause the permanent production time loss of the whole production line. Only those events whose durations exceed their opportunity windows[2] contribute to the permanent production loss to the system. The smallest downtime duration d_i^* for a machine m_{i_2} can be found as

$$d_i^* = \inf\{d \geq 0: \text{s.t.} T_{M^*} \int_{t_i}^{t_i+d_i} s_{M^*}(t; E)dt = W(m_{i_2})\}$$

[2] In other words, those disruption events cause the unique slowest machine to be starved, blocked or breakdown.

where $\int_{t_i}^{t_i+d_i} s_{M^*}(t;E)dt$ denotes the production volumes of machine m_{M^*} during time $(t_i, t_i + d_i]$. d_i^* is the time it takes for the buffers between machines m_{i_2} and m_{M^*} to become empty ($m_{i_2} < m_{M^*}$) or full ($m_{i_2} > m_{M^*}$). If the actual downtime duration $d_i \leq d_i^*$, there is no permanent production time loss. If the actual downtime duration $d_i > d_i^*$, the permanent production time loss equal to $d_i - d_i^*$. The permanent production time loss due to disruption event $\vec{e} = \left(m_{i_1}, m_{i_2}, t_i, d_i\right)$ is defined as

$$L(\vec{e}) = \begin{cases} d_i - d_i^*, & d_i > d_i^* \\ 0, & d_i \leq d_i^* \end{cases}$$

3.4 Dynamics in Serial Production Lines with Multiple Slowest Machines

Last section has discussed the impact of disruption events in serial production systems with single unique slowest machine. The analysis indicates that any stoppage event at the unique slowest machine, due to starvation, blockage, or machine failure, incurs production time loss to the system that is impossible to recover in the future. However, it is not unusual that a system can have multiple slowest machines. In such a scenario, there are more than one reference slowest machines that need to be considered to evaluate the impact of disruption events. Therefore, the last slowest machine S_{M^*} is selected as a reference machine. We will show that any stoppage at the last slowest machine S_{M^*} contributes to the production time loss of the line. It is noted that the output of the end-of-line machine is used as the output of a production line [17], i.e. $X(T) = \int_0^T s_M(t)dt$.

Proposition 3.1 *Given a realization of a production process subject to a sequence of disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ *and suppose* $\max_{l=1,\dots,n}\{t_l + d_l\} < T$*,if the last slowest machine* m_{M^*} *stops for a duration of D during* $(0, T]$*, then for the end-of-line machine* m_M*,* $\exists\, T^* \geq T$*, s.t.*

$$\int_0^{T'} s_M(t)dt - \int_0^{T'} s_M(t;E)dt = \mathrm{D}/T_{M^*}, \forall T' > T^* \tag{3.1}$$

where $\int_0^{T'} s_M(t)dt$ *and* $\int_0^{T'} s_M(t;E)dt$ *are the output of the end-of-line machine* m_M *without and with disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ *during* $(0, T']$.

Proof: For the line segment between machines m_{M^*} and m_M, applying the principle of conservation of flow, we have

$$\int_0^{T'} s_{M^*}(t)dt - \int_0^{T'} s_M(t)dt = \sum_{k=M^*}^{M-1} b_k(T') - \sum_{k=M^*}^{M-1} b_k(0) \tag{3.2}$$

$$\int_0^{T'} s_{M^*}(t;E)dt - \int_0^{T'} s_M(t;E)dt = \sum_{k=M^*}^{M-1} b_k(T';E) - \sum_{k=M^*}^{M-1} b_k(0;E) \tag{3.3}$$

where $\int_0^{T'} s_{M^*}(t)dt$ and $\int_0^{T'} s_{M^*}(t;E)dt$ denote the output of the last slowest machine m_{M^*} without and with disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ during $(0, T']$. $\int_0^{T'} s_M(t)dt$ and $\int_0^{T'} s_M(t;E)dt$ denote the output of the end-of-line machine m_M without and with disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ during $(0, T']$. Without disruption events, machine m_{M^*} is the unique slowest machine in the line segment between machines m_{M^*} and m_M, i.e. $T_{M^*}(t) > \max(\mathrm{T}_{M^*+1}, \dots, T_M)$. Machine m_{M^*} has the least output among machines $m_{M^*}, \dots, m_M$, i.e. $\int_0^{T'} s_{M^*}(t)dt < \min\{\int_0^{T'} s_{M^*+1}(t)dt, \dots, \int_0^{T'} s_M(t)dt\}$. The buffer levels between machines m_{M^*} and m_M will decrease gradually until they become empty, i.e. $\sum_{k=M^*}^{M-1} b_k(T_1) = 0$, after a period of T_1. Therefore, $\forall T' > T_1$, we have

$$\int_0^{T'} s_{M^*}(t)dt = \int_0^{T'} s_M(t)dt - \sum_{k=M^*}^{M-1} b_k(0) \tag{3.4}$$

Similarly, when the production line is subject to a sequence of disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, $\exists T_2$ such that $\forall T' > T_2 \geq T$, the buffer levels between the machines S_{M^*} and S_M is empty since there is no disruption events after time T. Therefore, the equation 3.3 becomes

$$\int_0^{T'} s_{M^*}(t;E)dt = \int_0^{T'} s_M(t;E)dt - \sum_{k=M^*}^{M-1} b_k(0;E) \tag{3.5}$$

Without disruption events, the last slowest machine will neither be starved nor blocked and its speed is always $s_{M^*}(t) = \frac{1}{T_{M^*}}$. The term $\int_0^{T'} s_{M^*}(t)dt$ in Equation 3.4 can be calculated as $\int_0^{T'} s_{M^*}(t)dt = \frac{T'}{T_{M^*}}$. Similarly, the term $\int_0^{T'} s_{M^*}(t;E)dt$ in Equation 3.5 can be calculated as $\int_0^{T'} s_{M^*}(t;E)dt = \frac{T'-D}{T_{M^*}}$. Given the production line has the same initial buffer levels, which is $\sum_{k=M^*}^{M-1} b_k(0) = \sum_{k=M^*}^{M-1} b_k(0;E)$, through Equations 3.4 and 3.5, we have $\exists T^* \geq \max\{T_1, T_2\}$,

$$\int_0^{T'} s_M(t)dt - \int_0^{T'} s_M(t;E)dt = \int_0^{T'} s_{M^*}(t)dt - \int_0^{T'} s_{M^*}(t;E)dt = \mathrm{D}/T_{M^*}, \forall T' > T^*$$

End of the proof.

Proposition 3.1 indicates that the production time loss caused by a disruption event can be quantified as the time that the last slowest machine is starved or blocked by the event. We denote the time as the permanent production time loss caused by the disruption event. Considering an arbitrary disruption event $\vec{e}_i = \left(m_{i_1}, m_{i_2}, t_i, d_i\right) \in E$, we want to find the exact time that the last slowest machine is starved or blocked by event $\vec{e}_i$, or in other words, the permanent production time loss caused by the event. For ease of expression, W_i is adopted to denote the time that machine m_{M^*} begins to be starved or blocked by event $\vec{e}_i$, which is the opportunity window of machine m_{i_2}. We will discuss the scenario that $d_i > W_i$; otherwise, the permanent production time loss will simply be zero.

In the case of $i_2 < M^*$, applying the principle of conservation of flow to the line segment between machines m_{i_2} and m_{M^*} during $(t_i, t_i + d_i]$, yields

$$\int_{t_i}^{t_i+d_i} s_{m_{i_2}}(t)dt - \int_{t_i}^{t_i+d_i} s_{M^*}(t;E)dt = \sum_{k=i_2}^{M^*-1} b_k(t_i + d_i;E) - b_k(t_i;E) \tag{3.6}$$

where $\int_{t_i}^{t_i+d_i} s_{m_{i_2}}(t)dt$ and $\int_{t_i}^{t_i+d_i} s_{M^*}(t;E)dt$ denote the output of machines m_{i_2} and m_{M^*} during $(t_i, t_i + d_i]$, respectively. Since machine m_{i_2} is down during $(t_i, t_i + d_i]$, the term $\int_{t_i}^{t_i+d_i} s_{m_{i_2}}(t)dt$ is zero. Equation 3.6 becomes

$$\int_{t_i}^{t_i+d_i} s_{M^*}(t;E)dt = \sum_{k=i_2}^{M^*-1} b_k(t_i;E) - b_k(t_i + d_i;E) \tag{3.7}$$

$\forall d \geq W_i$, all the buffers $B_{i_2+1}, \dots, B_{M^*}$ are empty at $t_i + d$, i.e. $\sum_{k=i_2}^{M^*-1} b_k(t_i + d;E) = 0$. The Equation 3.7 can be reorganized as

$$\int_{t_i}^{t_i+d_i} s_{M^*}(t;E)dt = \sum_{k=i_2}^{M^*-1} b_k(t_i;E) \tag{3.8}$$

Therefore, W_i can be represented as

$$W_i = \inf\{d \geq 0 : s.t. \int_{t_i}^{t_i+d} s_{M^*}(t;E)dt = \sum_{k=i_2}^{M^*-1} b_k(t_i;E)\}.$$

Machine m_{i_2} resumes operation at time $t_i + d_i$. Since the buffers between machines m_{i_2} and m_{M^*} are empty, it takes $\sum_{k=i_2}^{M^*-1} T_k$ for an unfinished job to reach the last slowest machine m_{M^*}. The permanent production time loss $TL(\vec{e}_i)$ caused by the disruption event $\vec{e}_i$ can be represented as

$$TL(\vec{e}_i) = d_i - W_i + \sum_{k=i_2}^{M^*-1} T_k \quad (3.9)$$

Similarly, if $i_2 > M^*$, following the same procedure by applying the principle of conservation of flow to the line segment between machines m_{i_2} and m_{M^*} except letting $\sum_{k=M^*}^{i_2-1} b_k(t_i + d; E) = \sum_{k=M^*}^{i_2-1} B_k$. W_i can be represented as

$$W_i = \inf\{d \geq 0 : s.t. \int_{t_i}^{t_i+d} s_{M^*}(t; E)dt = \sum_{k=M^*}^{i_2-1} B_k - b_k(t_i; E)\}.$$

Machine m_{i_2}will resume operation at time $t_i + d_i$. Machines $m_{M^*}, \ldots, m_{i_2}$ receive an unfinished job from their upstream buffers immediately. The last slowest machine m_{M^*} is no longer blocked. The permanent production time loss caused by the disruption event becomes

$$TL(\vec{e}_i) = d_i - W_i \quad (3.10)$$

In the case of $i_2 = M^*$, since the breakdown of the last slowest machine m_{M^*} directly contribute to the overall permanent production loss of the production line, the permanent production time loss resulted from the disruption event can be calculated as

$$TL(\vec{e}_i) = d_i \quad (3.11)$$

We can combine Equations 3.9, 3.10 and 3.11 into a single equation

$$TL(\vec{e}_i) = \begin{cases} \max\{d_i - W_i + \sum_{k=i_2}^{M^*-1} T_k, 0\}, & i_2 < M^* \\ d_i, & i_2 = M^* \\ \max\{d_i - W_i, 0\}, & i_2 > M^* \end{cases} \quad (3.12)$$

where W_i can be expressed as

$$W_i = \begin{cases} \inf\left\{ \begin{array}{c} d \geq 0: s.t. \int_{t_i}^{t_i+d} s_{M^*}(t; E)dt = \\ \sum_{k=i_2}^{M^*-1} b_k(t_i; E) \end{array} \right\}, & i_2 < M^* \\ 0, & i_2 = M^* \\ \inf\left\{ \begin{array}{c} d \geq 0: s.t. \int_{t_i}^{t_i+d} s_{M^*}(t; E)dt = \\ \sum_{k=M^*}^{i_2-1} B_k - b_k(t_i; E) \end{array} \right\}, & i_2 > M^* \end{cases} \tag{3.13}$$

The value of W_i can be obtained through easy calculation based on the sensor information, i.e. the least time it takes for the buffers between machines m_{i_2} and m_{M^*} to become empty or full.

3.5 Conclusion

The analysis provides a quantitative way to evaluate the impact of each disruption event to the whole production system in terms of permanent production time loss. It is important to note that a serial production line with single unique slowest machine is a special case of a general serial line with multiple slowest machines. The above discussion unifies the disruption event analysis of serial production lines with single or multiple slowest machines.

Chapter 4

REAL-TIME ANALYSIS OF DISRUPTION EVENTS IN PARALLEL PRODUCTION SYSTEMS

4.1 Introduction

Manufacturing production systems with parallel structures are widely used in many production systems to achieve a greater productivity or reliability. A parallel structure can be split into several sub-lines in parallel. In case one machine in a parallel structure fails, parts can still go through other sub-lines to keep production moving, but with a slower rate. Although tremendous efforts have been devoted to the performance evaluation on serial transfer lines, difficulties still exist in studying production lines with parallel structures because of their complex configuration. In addition, transient characteristics can be potentially very useful in real-time production control [10, 41]. The transient analysis of production lines with parallel structures is even more difficult. Therefore, in this section, we reveal the impact of each disruption event in a parallel production system and characterize its transient behavior to guide for real-time production control. We will use downtime events as examples. Other disruption events caused by supporting activities can be similarly analyzed.

4.2 System Description, Assumptions and Notations

For ease of expression, the continuous flow model is adopted in this chapter to analyze the dynamics of a parallel production line, because the production dynamics can be conveniently described by integral or differential equations [67, 68]. The method can be easily extended to

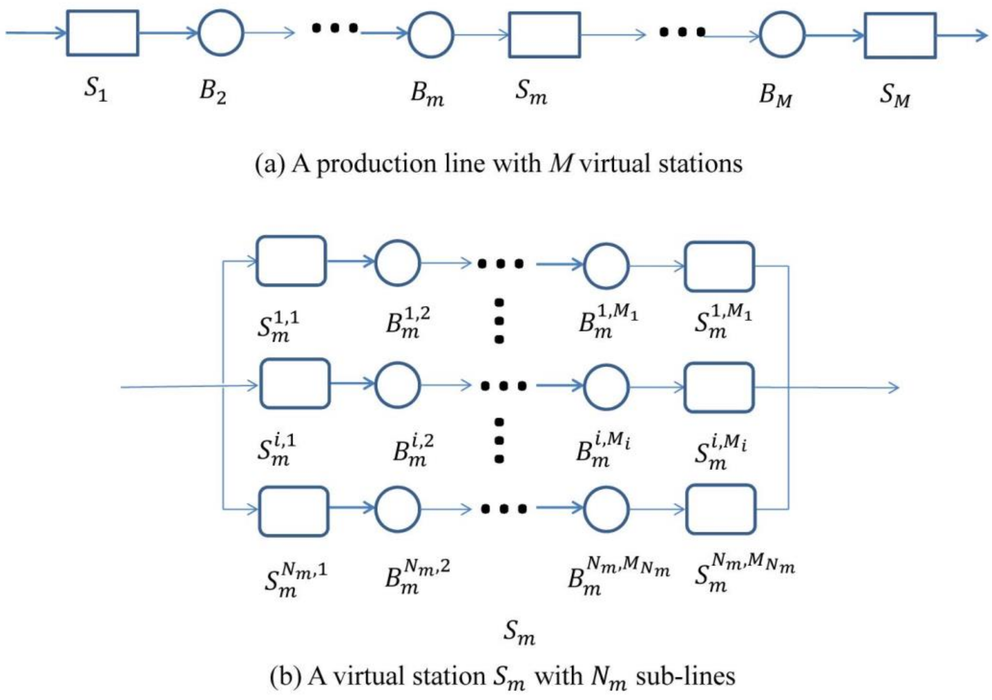


Figure 4.1 Demonstration of the interdependency and dynamic interactions among production and supporting activities

discrete event systems. The continuous flow model assumes that the quantity of parts in a buffer varies continuously from zero to its capacity. A production line with multiple parallel structures or single machines is illustrated in Figure 4.1 (a), and Figure 4.1 (b) represents the zoom-in detail of a parallel structure on the line. The following assumptions are made in this paper:

1) In Figure 4.1 (a), each rectangle represents a single machine or a parallel structure denoted as m_i, $1 \leq i \leq M$. m_i is defined as a virtual machine, each circle represents a buffer denoted as b_i, $1 \leq i \leq M - 1$. There are M virtual machines and $M - 1$ buffers in a parallel production line. B_i is used to denote the maximum capacity of buffer b_i and $b_i(t)$ is used to denote the buffer level at time instant t.

2) As shown in Figure 4.1 (b), a virtual machine m_i consists of N_i serial sub-lines, $N_i \geq 1$. For the *j*th sub-line in a virtual machine, $1 \leq j \leq N_i$, there are M_j single machines and $M_j - 1$ buffers, $M_j \geq 1$. Each rounded rectangle represents a single machine denoted as $m_i^{j,k}$, which is the *k*th machine in the *j*th sub-line, $1 \leq k \leq M_j$. Each circle represents a buffer denoted as $b_i^{j,k}$, which is the *k*th buffer in the *j*th sub-line. $B_i^{j,k}$ is used to denote the maximum capacity of buffer $b_i^{j,k}$ and $b_i^{j,k}(t)$ is used to denote the buffer level at time instant *t*.

3) $s_i^{j,k}(t)$ denotes the speed of machine $m_i^{j,k}$ at time instant *t* and $T_i^{j,k}$ denotes the cycle time, with $\frac{1}{T_i^{j,k}}$ being the rated speed.

4) m_i^{j,M_j^*} is denoted as the last slowest machine in the *j*th sub-line of parallel structure m_i, i.e., $M_j^* = \arg\min_{1 \leq k \leq M_j} \frac{1}{T_i^{j,k}}$, which is closed to the end-of-line machine.

5) For each parallel structure, the first machine in each sub-line, $m_i^{j,1}, 1 \leq j \leq N_i$, has equal probability to take the last part in buffer b_{i-1}, if it is not blocked. Similarly, buffer b_i receive the last fraction of part (to make the buffer full) from any unstarved machines $m_i^{j,M_j}, 1 \leq j \leq N_i$, with equal probabilities.

6) Machines cannot fail when they are idle.

7) In the main production line, as shown in Figure 1 (a), the first virtual machine can never be starved and the last virtual machine can never be blocked, while starvation and blockage definition for a virtual machine will be provided in Section 4.3.

8) The parallel production system is subject to a sequence of disruption events, denoted as $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, where $\vec{e}_i = \left(m_i^{j,k}, t_i, d_i\right)$ denotes machine $m_i^{j,k}$ is down at t_i and lasts for a time period of d_i.

4.3 Dynamics of A Single Parallel Structure

To start the analysis of a production line with mixed single machines and parallel structures, a virtual machine concept is introduced as described in Section 4.2, Figures 4.1 (a) and 4.1 (b). A virtual machine m_i is defined as a machine including N_i sub-lines, $N_i \geq 1$. Therefore, a virtual machine could be a single machine with only one work-in-process when $N_i = 1$, or a parallel structure with multiple work-in-process when $N_i > 1$.

Consider an isolated virtual machine m_i with N_i sub-lines, it is assumed that the production of each sub-line is independent from each other. This is a realistic assumption for industrial parallel production lines. Therefore, the superposition property is applied to a virtual machine in terms of its throughput and the throughput of every sub-line.

Each sub-line in a virtual machine is a serial line. Therefore, the theories developed for serial lines from previous studies can be applied. Based on Chapter 3, we define the opportunity window $W_i^{j,k}$ of a machine $m_i^{j,k}$ at the *i*th sub-line of a virtual machine m_i, as the longest downtime duration for the machine at time *t* without causing permanent production loss at the end-of-line machine in the sub-line, i.e. $W_i^{j,k} = \sup\{d \geq 0 \colon \mathrm{s.t.} \exists T^*(d), \forall T > T^*(d), \int_0^T s_i^{j,M_j}(t)dt = \int_0^T s_i^{j,M_j}(t;\vec{e})dt\}$, where $\int_0^T s_i^{j,M_j}(t;\vec{e})dt$ and $\int_0^T s_i^{j,M_j}(t)dt$ are the production volume of the end-of-line machine m_i^{j,M_j} with and without an inserted disruption event $\vec{e} = \left(m_i^{j,k}, t_i, d_i\right)$ respectively.

It has been proved in serial lines that not all the disruption events can cause permanent production loss to the system. Only those that last longer than their opportunity windows can cause permanent production loss to the whole line. Furthermore, if a disruption event causes a stoppage event of the last slowest machine, then the production loss on the last slowest machine is the permanent production loss for the line and all other machines. This conclusion applies to any sub-line of a virtual machine. Therefore, the permanent production loss on a sub-line can be similarly evaluated. Suppose there is an event $\vec{e} = \left(m_i^{j,k}, t_i, d_i\right)$ at the ith sub-line of virtual machine m_i, whose downtime duration d is larger than its corresponding opportunity window $W_i^{j,k}$. Then in the sub-line, for any machine $m_i^{j,l}, 1 \leq l \leq M_j$, there is always a $T^* \geq t + d$, which depends on the location of the last slowest machine m_i^{j,M_j^*}, such that

$$\int_0^T s_i^{j,l}(t)dt - \int_0^T s_i^{j,l}(t;\ \vec{e})dt = \int_0^T s_i^{j,M_j^*}(t)dt - \int_0^T s_i^{j,M_j^*}(t;\ \vec{e})dt\,, \forall T \geq T^* \tag{4.1}$$

where $\int_0^T s_i^{j,l}(t;\ \vec{e})dt$ and $\int_0^T s_i^{j,l}(t)dt$ are the production volume of machine $m_i^{j,l}$ with and without disruption event $\vec{e}$ and $\int_0^T s_i^{j,M_j^*}(t;\ \vec{e})dt$ and $\int_0^T s_i^{j,M_j^*}(t)dt$ are the production volume of machine m_i^{j,M_j^*} with and without disruption event $\vec{e}$. It is noted that $s_i^{j,M_j^*}(t;\ \vec{e})$ can be measured in normal production and the rate $s_i^{j,M_j^*}(t)$ is just the rated speed $\frac{1}{T_i^{j,M_j^*}}$ of machine m_i^{j,M_j^*}. Equation 4.1 indicates that in a sub-line if a downtime event causes production loss to the last slowest machine, then the production system suffer the same amount of production loss.

Since a virtual machine m_i satisfies superposition property, the permanent production loss at a virtual machine is simply the summation of the permanent production loss at each sub-line. We use L_i to denote the permanent production loss at a virtual machine m_i during $(0, T]$ amd it can be expressed as

$$L_i = \sum_{j=1}^{N_i} \int_0^T s_i^{j,M_j^*}(t)dt - \int_0^T s_i^{j,M_j^*}(t;\ \vec{e})dt \tag{4.2}$$

A set $\widetilde{m}_i = \{m_i^{1,M_1^*}, \dots, m_i^{N_i,M_{N_i}^*}\}$ is defined to include the slowest machine in each sub-line of a virtual machine m_i. The notations $\tilde{s}_i(t)$ and $\tilde{s}_i(t;\, \vec{e})$ are denoted as $\tilde{s}_i(t) = \sum_{j=1}^{N_i} s_i^{j,N_j}(t)$ and $\tilde{s}_i(t;\, \vec{e}) = \sum_{j=1}^{N_i} s_i^{j,N_j}(t;\, \vec{e})$ to represent the summation of production rates of all machines in the set $\widetilde{m}_i$ without and with disruption event $\vec{e}$. $\frac{1}{T_i}$ is defined as $\frac{1}{T_i} = \sum_{j=1}^{N_i} \frac{1}{T_i^{j,M_j^*}}$, which is the summation of the rated speed of the slowest machine in each sub-line of the virtual machine m_i. Then Equation 4.2 can be rewritten as

$$L_i = \int_0^T \tilde{s}_i(t)dt - \int_0^T \tilde{s}_i(t;\, \vec{e})dt \tag{4.3}$$

Similarly, the value of the rate $\tilde{s}_i(t;\, \vec{e})$ can be measured in normal production and the rate $\tilde{s}_i(t)$ is just $\frac{1}{T_i}$. Therefore, any disturbance event resulting in a decrease of the rate $\tilde{s}_i(t;\, \vec{e})$ contribute to a permanent production loss of the virtual machine m_i.

In serial lines, it is well defined that a machine is starved if it is up and the upstream buffer is empty, and a machine is blocked if it is up with the downstream buffer is full and the downstream machine does not take a part from the buffer. However, there are no straight forward definitions for starvation and blockage of a virtual machine. For a virtual machine m_i, even when its upstream buffer is empty, virtual machine m_i may still have work-in-process (WIP) which may maintain the same production rate of the virtual machine. Similarly, if the downstream buffer of m_i is full, it may still have available space which can accept parts from its upstream virtual machines. Therefore, a virtual machine can be reasonably treated as being starved or blocked if the duration of the upstream buffer being empty or the downstream buffer being full is long enough to cause the rate $\tilde{s}_i(t;\, \vec{e})$ to decrease.

Definition 4.1 *Assume buffer* b_{i-1} *in front of a virtual machine* $m_i, i \geq 2$, *is empty, this event can be denoted as* $\vec{e}_s = (b_{i-1} = 0, t, d)$ *to represent buffer* b_{i-1} *is empty at time t for a duration of d.* m_i *is defined to be starved if* $\vec{e}_s$ *causes permanent production loss at virtual machine* m_i *as evaluated in Equation 4.3.*

Definition 4.2 *Assume buffer* b_i *in the downstream of a virtual machine* $m_i, i \geq 2$, *is full, and the downstream machine* m_{i+1} *does not take a part from the buffer* b_i, *this event can be denoted as* $\vec{e}_b = (b_i = B_i, t, d)$ *to represent buffer* b_i *is full at time t for a duration of d.* m_i *is defined to be blocked if* $\vec{e}_b$ *causes permanent production loss at virtual machine* m_i *as evaluated in Equation 4.3.*

Note that the definitions are applicable for a virtual machine consisting of single machine or multiple sub-lines. In the case of single machine, disruption events $\vec{e}_s$ and $\vec{e}_b$ causes immediate stoppage of the virtual machine m_i.

4.4 Analysis of A Sequence of Concurrent Disruption Events in A Parallel Production Line

In this Section, we will discuss how disruption events impact a parallel production system with multiple virtual machines. It is not unusual that there may be multiple slowest virtual machines in a parallel production line, which is defined as $m_{M_i^*}$, $1 \leq i \leq M$, with the smallest speed $\frac{1}{T_{M_i^*}}$, i.e. $M_i^* = \arg\min_{1 \leq j \leq M} \frac{1}{T_j}$. It has been proved in a serial line with multiple slowest machines that any disruption events resulting in a starvation or blockage of the last slowest virtual machine contributes to the permanent production loss of the line. A similar conclusion can be obtained in parallel production systems.

Similarly, the throughput of the end-of-line virtual machine is used as the throughput of a line, and permanent production loss at the end-of-line virtual machine is defined as the permanent production loss of the line. Among the multiple slowest virtual machine, the last slowest virtual machine, or in other words, the one closest to the end-of-line virtual machine, is naturally selected. For convenience, M^* is still used to refer to the last slowest virtual machine.

Proposition 4.1 *Given a realization of a production process subject to a sequence of disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ *and suppose* $\max_{l=1,\dots,n}\{t_l + d_l\} < T$*, for the end-of-line virtual machine* m_M*,* $\exists\, T^* \geq T$, s.t.

$$\int_0^{T'} \tilde{s}_M(t)dt - \int_0^{T'} \tilde{s}_M(t;\, E)dt = \int_0^{T'} \tilde{s}_{M^*}(t)dt - \int_0^{T'} \tilde{s}_{M^*}(t;\, E)dt\,,\ \forall T' > T^* \qquad (4.4)$$

Proof: For convenience, a notation b_{k-l} is used to represent a set of buffers which includes all the buffers on the path from a set of machines *k*, such as $\widetilde{m}_i$, to another set of machines *l*. B_{k-l} denotes the summation of the buffer capacities of all the buffers in set b_{k-l}, and $b_{k-l}(t)$ is the summation of their buffer levels at time *t*.

For the line segment between the sets $\widetilde{m}_{M^*}$ and $\widetilde{m}_M$, applying the conservation of flow, we have

$$\int_0^{T} \tilde{s}_{M^*}(t)dt = \int_0^{T} \tilde{s}_M(t)dt + (b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(T) - b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0))$$

$$\int_0^{T} \tilde{s}_{M^*}(t;E)dt = \int_0^{T} \tilde{s}_M(t;E)dt + (b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(T;E) - b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0;E))$$

Without downtime events, the virtual machine m_{M^*} is treated as the unique slowest machine in the line segment from virtual machines m_{M^*} to m_M, i.e. $\frac{1}{T_{M^*}} < \min\{\frac{1}{T_{M^*+1}}, \dots, \frac{1}{T_M}\}$. The total buffer levels $b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(T')$ between $\widetilde{m}_{M^*}$ and $\widetilde{m}_M$ will decrease gradually until they become zero after a time period of T_1^*. Therefore, $\forall T' \geq T_1^*$, we have

$$\int_0^{T'} \tilde{s}_{M^*}(t)dt = \int_0^{T'} \tilde{s}_M(t)dt + -b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0)$$

When a production process is subject to a set of disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, there exists T_2^*, and $b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(T'; E) = 0$, $\forall T' \geq T_2^*$, since there is no disruption events after T. Therefore, $\forall T' \geq T_2^*$, we have

$$\int_0^{T'} \tilde{s}_{M^*}(t; E)dt = \int_0^{T'} \tilde{s}_M(t; E)dt - b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0; E)$$

Given the fact that the initial conditions are exactly the same, i.e. $b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0) = b_{\widetilde{m}_{M^*}-\widetilde{m}_M}(0; E)$, we have $\forall T' \geq \max(T_1^*, T_2^*)$

$$\int_0^{T'} \tilde{s}_M(t)dt - \int_0^{T'} \tilde{s}_M(t;\ E)dt = \int_0^{T'} \tilde{s}_{M^*}(t)dt - \int_0^{T'} \tilde{s}_{M^*}(t;\ E)dt$$

End of the proof.

Note that the value of the rate $\tilde{s}_{M^*}(t; E)$ can be measured in normal production and the rate $\tilde{s}_{M^*}(t)$ is just $\frac{1}{T_{M^*}}$. The Proposition 4.1 indicates that if a disruption event causes a production loss to the last slowest virtual machine m_{M^*}, then it also causes the same amount of permanent production loss to the production line. This naturally applies to a production line with a unique slowest virtual machine since the slowest one is the last slowest one.

Evaluating the impact of disruption events to a production line with multiple virtual machines are very important in real-time production control, such as prioritizing the limited resources to the most needed location. To understand the permanent production loss caused by an arbitrary disruption event, analyzing the starvation and blockage condition of the last slowest virtual machine is important. For convenience, we use $ST_i(t) = 1$ and $BL_i(t) = 1$ to denote virtual machine m_i is starved and blocked at a given time t, and $ST_i(t) = 0$ and $BL_i(t) = 0$ to denote virtual machine m_i is not starved and blocked at time t.

Proposition 4.2 *Given a realization of the production process subject to a sequence of disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, the necessary and sufficient condition for permanent production loss because of a disruption event $\vec{e}_f = \left(m_i^{j,k}, t_f, d_f \right)$, $m < M^*$ is*

$$ST_i(t; E) = 0 \text{ and } ST_x(t; E) = 1, \ \forall x \in (i, \ M^*] \tag{4.5}$$

where $t' < t < t_l + d_l$ and $t' = \inf\left\{t \geq t_l : \sum_{l=k}^{M_i-1} b_i^{j,l}(t; E) = 0\right\}$. t' is the time for buffers between the machine $m_i^{j,k}$ and the end-of-line machine m_i^{j,M_j} in the jth sub-line of the virtual machine m_i to become empty.

Proof: The output rate of the virtual machine m_i is $s_{i_{out}}(t) = \sum_{l=1}^{N_i} s_i^{l,M_l}(t)$. $s_{i_{out}}(t)$ will not be decreased because of the disruption event $\vec{e}_f$ until the last machine m_i^{j,M_j} in the sub-line j being starved, i.e. $s_i^{j,M_j}(t) = 0$ or in other words, $\sum_{l=k}^{M_j-1} b_i^{j,l}(t) = 0$. Let $t' = \inf\left\{t \geq t_l : \sum_{l=k}^{M_i-1} b_i^{j,l}(t; E) = 0\right\}$, we will prove the sufficient condition by contradiction.

Finally, we suppose there is a virtual machine between virtual machines m_i and m_{M^*} not being starved, i.e. $\exists x \in (i, M^*]$, $ST_x = 0$. Then, the permanent production loss is caused by disruption events at the virtual machine m_x rather than disruption event $\vec{e}_f$. Then we suppose that the virtual machine m_i is not starved, i.e. $ST_i = 1$. The permanent production loss is caused by disruption events at its upstream virtual machines, which also contradicts with the assumption that permanent production loss occurs due to the disruption event $\vec{e}_f$. Therefore, $t >$

t': $ST_i(t;E) = 0$ and $ST_x(t;E) = 1$, $\forall x \in (i,\ M^*]$ is the sufficient condition for existing permanent production loss caused by the disruption event $\vec{e}_f$.

The necessary condition is straight forward. If $\exists t > t'$: $ST_i(t;E) = 0$ and $ST_x(t;E) = 1$, $\forall x \in (i,\ M^*]$, then the slowest virtual machine is starved because of the disruption event $\vec{e}_f$.

End of the proof.

Therefore, one can always find the smallest possible downtime duration d_l^* such that Equation 4.5 is satisfied, i.e.

$$d_f^*(t) = \inf\{t > t':\ ST_i = 0 \text{ and } ST_x(t;E) = 1, \forall x \in (i, M^*]\} \tag{4.6}$$

Proposition 4.3 *Given a realization of the production process subject to a sequence of disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, *the necessary and sufficient condition for permanent production loss because of a disruption event* $\vec{e}_f = \left(C_0, m_i^{j,k}, t_f, d_f\right)$, $m > M^*$ *is*

$$BL_i(t;E) = 0 \text{ and } BL_x(t;E) = 1,\ \forall x \in [M^*,\ i) \tag{4.7}$$

where $t' < t < t_l + d_l$ *and* $t' = \inf\left\{t \geq t_l: \sum_{l=1}^{k-1} b_i^{j,l}(t;E) = \sum_{l=1}^{k-1} B_i^{j,l}\right\}$. t' *is the time for buffers between the machine* $m_i^{j,1}$ *and the end-of-line machine* $m_i^{j,k}$*in the jth sub-line of the virtual machine* m_i *to become full.*

Proof: The input rate of the virtual machine m_i is $v_{i_{in}}(t) = \sum_{l=1}^{N_i} v_i^{l,1}(t)$. $v_{i_{in}}(t)$ will not be decreased because of the disruption event $\vec{e}_f$ until the first machine $m_i^{j,1}$ in the sub-line *j* being blocked, i.e. $v_i^{j,1}(t) = 0$ or in other words, $\sum_{l=1}^{j-1} b_i^{j,l}(t) = \sum_{l=1}^{j-1} B_i^{j,l}$. Let $t' = \inf\left\{t \geq t_l: \sum_{l=1}^{j-1} b_i^{j,l}(t) = \sum_{l=1}^{j-1} B_i^{j,l}\right\}$, we will prove the sufficient condition by contradiction.

Finally, we suppose there is a virtual machine between virtual machines m_i and m_{M^*} not being blocked, i.e. $\exists x \in [M^*, m_i)$, $BL_x = 0$. Then, the permanent production loss is caused by disruption events at the virtual machine m_x rather than disruption event $\vec{e}_f$. Then we suppose that the virtual machine m_i is not blocked, i.e. $BL_i = 1$. The permanent production loss is caused by disruption events at its downstream virtual machines, which also contradicts with the assumption that permanent production loss occurs due to the disruption event $\vec{e}_f$. Therefore, $t > t'$: $BL_i(t; E) = 0$ and $BL_x(t; E) = 1,\ \forall x \in [M^*,\ i)$ is the sufficient condition for existing permanent production loss caused by the disruption event $\vec{e}_f$.

The necessary condition is straight forward. If $\exists t > t'$: $BL_i(t; E) = 0$ and $BL_x(t; E) = 1$, $\forall x \in [M^*,\ i)$, then the slowest virtual machine is blocked because of the disruption event $\vec{e}_f$.

End of the proof

Similarly, one can always find the smallest possible downtime duration d_l^* such that Equation 4.8 is satisfied. i.e.

$$d_f^*(t) = \inf\{t > t' \colon\ BL_i(t; E) = 0 \text{ and } BL_x(t; E) = 1,\ \forall x \in [M^*,\ i)\} \tag{4.8}$$

Proposition 4.4 *Given a realization of the production process subject to a sequence of disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, *the necessary and sufficient condition for permanent production loss because of a disruption event* $\vec{e}_f = \left(C_0, m_i^{j,k}, t_f, d_f\right),\ m = M^*$ *is*

$$BL_i(t; E) = 0 \text{ and } ST_i(t; E) = 0,\ i = M^* \tag{4.9}$$

where $t' < t < t_l + d_l$ *and*

$$t' = \begin{cases} \inf\left\{t \geq t_l : \sum_{l=k}^{M_j^*-1} b_i^{j,l} = 0\right\}, & k < M_j^* \\ t_l, & k = M_j^* \\ \inf\left\{t \geq t_l : \sum_{l=M_j^*}^{k-1} b_i^{j,l} = \sum_{l=M_j^*}^{k-1} B_i^{j,l}\right\}, & k > M_j^* \end{cases}$$

t' is the time for buffers between the machine $m_i^{j,k}$ and the last slowest machine m_i^{j,M_j^*} in the jth sub-line of the virtual machine m_i to become empty ($i < M_j^*$) or full ($i > M_j^*$).

Proof: We choose $k < M_j^*$ as an example. The case when $j \geq M_i^*$ can be easily proved in a similar way. The rate $\tilde{v}_{M^*}(t;E)$ of the virtual machine m_{M^*} will not decrease because of the disruption event $\vec{e}_f$ until the machine $m_{M^*}^{j,M_j^*}$ is starved, i.e. $v_{M^*}^{j,M_j^*}(t;E) = 0$, or in other words, $\sum_{l=j}^{M_j^*-1} b_{M^*}^{j,l}(t;E) = 0$. Let $t' = \inf\{t \geq t_l : \sum_{l=j}^{M_j^*-1} b_{M^*}^{j,l}(t;E) = 0\}$, we will prove the sufficient by contradiction.

Firstly, we suppose that the last slowest virtual machine m_{M^*} is starved or blocked. The permanent production loss is caused by disruption events at its upstream and downstream virtual machines, which is contradicted with the assumption that the permanent production loss occurs because of the disruption event $\vec{e}_f$. Therefore, $t > t' : BL_{M^*}(t;E) = 0$ and $ST_{M^*}(t;E) = 0$ is the sufficient condition for existing permanent production loss caused by the disruption event $\vec{e}_f$.

The necessary condition is straight forward. $\forall t > t'$, we have $\tilde{v}_{M^*}(t;E) = \tilde{v}_{M^*}(t)$. If $>$ t': $BL_{M^*}(t;E) = 0$ and $ST_{M^*}(t;E) = 0$, the last slowest virtual machine is neither starved nor blocked. Permanent production loss can only occur because of the disruption event $\vec{e}_f$.

End of the proof

Therefore, we can always find the slowest possible downtime duration d_f^* of the disruption event $\vec{e}_f$ such that Equation 4.9 is satisfied

$$d_f^*(t) = \inf\{t > t':\ BL_i(t; E) = 0 \text{ and } ST_i(t; E) = 0,\ i = M^*\} \quad (4.10)$$

The smallest possible downtime duration d_f^* of a disruption event which will not cause permanent production loss for the line can be evaluated using Equations 4.6, 4.8 and 4.10. The threshold d_f^* is actually the opportunity window of the machine $m_i^{j,k}$. If the actual downtime duration d_l is less than the threshold d_f^*, there is no permanent production loss. However, if the downtime duration d_l exceeds d_f^*, the permanent production loss due to $\vec{e}_f$ becomes nonnegative. The opportunity window d_f^* for machine $m_i^{j,k}$ is the time it takes, from time t_l, for the last slowest virtual machine m_{M^*} to just become starved ($m < M^*$) or blocked ($m > M^*$).

However, to attribute the permanent production loss to associated disruption events in a parallel production system becomes more complicated than in a serial line. In serial lines, the last slowest virtual machine only operates at two speeds: its rate speed and zero. The permanent production time loss caused by a disruption event $\vec{e}_f = \left(m_i, t_f, d_f\right)$ in a serial line can be expressed as $d_f - d_f^*$ where d_f is the downtime duration and d_f^* is the opportunity window of the machine m_i at time t. However, in a parallel production system, the last slowest virtual machine can operate at more than two rates, and its rate $\tilde{s}_{M^*}(t; E)$ keeps changing. Therefore, it is very difficult to evaluate the permanent production loss based on the changing $\tilde{s}_{M^*}(t; E)$. For a disruption event $\vec{e}_f = \left(m_i^{j,k}, t_f, d_f\right)$, the value $d_f - d_f^*$ does not represent an exact permanent production time loss, but only indicates the longest possible duration of a permanent production loss caused by $\vec{e}_f$. There is no simple proportional relationship between the permanent production loss and the value $d_f - d_f^*$ in parallel systems.

For ease of discussion, a notation $L(\vec{e}_f)$ is used to represent a permanent production loss caused by a disruption event $\vec{e}_f$. For a parallel system subjecting to a sequence of disruption events $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$, when E contains only one disruption event $\vec{e}_1 = \left(m_i^{j,k}, t_1, d_1\right)$, the permanent production loss of the system caused by $\vec{e}_1$ can be calculated based on Proposition 4.1 as

$$L(\vec{e}_1) = \begin{cases} \int_{t_1+d_1^*}^{t_1+d_1} \tilde{s}_{M^*}(t)dt - \int_{t_1+d_1^*}^{t_1+d_1} \tilde{s}_{M^*}(t; \vec{e}_1)dt, & d_1 > d_1^* \\ 0, & d_1 \leq d_1^* \end{cases}$$

where d_1^* is the opportunity window of the machine $m_i^{j,k}$, which can be determined from Equations 4.6, 4.8 and 4.10 depend on the location of $m_i^{j,k}$.

If E contains more than one disruption event, the rate $\tilde{s}_{M^*}(t; E)$ keeps changing subject to disruption events $\vec{e}_1, \dots, \vec{e}_n$. In addition, it is possible that two or more disruption events can overlap, i.e. $\exists p \neq q,\ \left[t_p, t_p + d_p\right) \cap \left[t_q, t_q + d_q\right) \neq \phi$. In this case, it is even more difficult to evaluate the permanent production loss and attribute the loss to associated downtime events. For convenience, a notation $\tau\left(t;\ \vec{e}_f\right) = \{0, 1\}$ is adopted to denote whether a disruption event $\vec{e}_f$ causes permanent production loss. Specifically, $\tau\left(t;\ \vec{e}_f\right) = 0$ indicates that the event $\vec{e}_f$ does not cause permanent production loss and $\tau\left(t;\ \vec{e}_f\right) = 1$ indicates that the event $\vec{e}_f$ causes permanent production loss. Usually, the starvation, blockage and breakdown of each machine can be obtained from production lines. Using real data or simulation, it can be determined if a virtual machine is starved or blocked based on Definitions 4.1 and 4.2. In addition, $\tau\left(t;\ \vec{e}_f\right)$ can be determined based on Propositions 4.2, 4.3 and 4.4. Then the permanent production loss caused by a disruption event $\vec{e}_f$ to the overall system can be calculated as

$$L(\vec{e}_1) = \begin{cases} \int_{t_1+d_1^*}^{t_1+d_1} \tau\big(t;\ \vec{e}_f\big)[\tilde{s}_{M^*}(t) - \tilde{s}_{M^*}(t; \vec{e}_1)]dt\,, & d_1 > d_1^* \\ 0, & d_1 \leq d_1^* \end{cases}$$

It is possible, by coincidence, two or more different disruption events may result in overlapping permanent production loss to associated disruption events by using any allocation rules, such as even distribution of the amount of the production loss.

4.5 Conclusion

In this Section, the concept of virtual machine is introduced to represent a single machine or a parallel structure in a production line. The dynamic of a single virtual machine is studied. The superposition property of a virtual machine determines that any permanent production loss at a sub-line is also permanent to the whole virtual machine. This research unifies the analysis of serial production lines and parallel production lines in terms of disruption events impacts. The analysis suggests that the impacts of any disruption events are only apparent when the last slowest virtual machine is starved or blocked.

Chapter 5

EVENT-BASED MODELING OF DISTRIBUTED SENSOR NETWORKS IN PRODUCTION SYSTEMS

5.1 Introduction

Multistage manufacturing systems are characterized by their complex dynamics subject to constant changes caused by technology insertion, engineering modifications, as well as disruption events. To support daily operation, distributed sensors are used to provide real-time data describing the status of each process. Despite the big potential in improving productivity, the advantages of distributed sensor networks are not fully realized for overall system efficiency due to a lack of system level modeling method.

Motivated by this need, an event-based modeling (EBM) method is developed to evaluate the performance of all the stations and supporting activities in real-time based on the sensor data (e.g., buffer levels and machine random failures). This approach quantifies the systematic impacts of all the production and supporting activities with single unified index. It provides a severity ranking of production and supporting activities which is very useful for plant managers to allocate the limited resources to where they are needed the most.

5.2 A Virtual Multiple Layers Sensor Framework

To serve for the system monitoring and modeling, a virtual multiple layers sensor framework is established. Figure 5.1 shows a certain battery production line with a distributed sensor network. The sensor network is “sliced” into three virtual layers and each layer has its unique function.

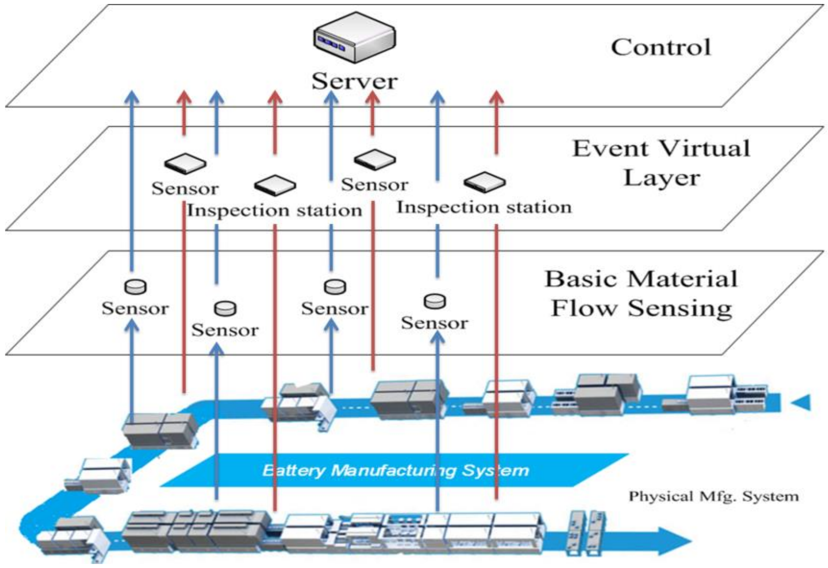


Figure 5.1 A production line with sensor system

The first virtual layer is defined as material flow sensing layer which includes inductive proximity sensors, counter sensors, etc. to track all time-stamped material flow. Both buffer levels and machines' speeds are measured by the sensors in the first virtual layer. The second virtual layer is defined as event diagnosis layer which includes thermal sensors, pressure sensors, inspection machines, etc. to track the disruption events resulted from machines and supporting activities. The third virtual layer is system modeling layer which interprets the sensed information from the first two layers to formulate an integrated system model.

5.3 Mathematical System Dynamics Description

Event-based modeling (EBM) is developed to capture all the activities, resources and disruption events from the multiple layers sensor framework in a production system[3]. The material flow sensing layer captures the material flow and the event diagnose layer captures the disruption events and possible changes. The interactions among machines and supporting activities can be

[3] For ease of discussion, a serial production line defined in Chapter 3 is adopted. The approach can be easily extended to complex production lines with parallel structures.

treated as “internal forces”, and the disruptions can be treated as “external forces”. Therefore, the system dynamics with sensor data can be represented by a state space equation as:

$$\dot{X}(t) = f(t, X(t), U(t)) \tag{5.1}$$

where $U(t) = \vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ denotes a sequence of disruption events during period $(0, t]$.

To solve the state space equation 5.1, the following homogeneous and nonhomogeneous functions are considered.

$$\dot{X}(t) = f(t, X(t)) \tag{5.2}$$

$$\dot{X}(t) = f(t, X(t), U(t)) \tag{5.3}$$

Equation 5.2 describes a virtual scenario that there are no disruption events in a production line. The output of the system is constrained by the base cycle time of the slowest machine, i.e. T_{M^*} [7, 17]. Therefore, the complementary solution $X_c(t)$ for the homogeneous function can be denoted as

$$X_c(t) = t/T_{M^*} \tag{5.4}$$

$X_c(t)$ is the largest production output that can be possibly produced by the system during $(0, t]$ and is denoted as base output.

Equation 5.3, on the other hand, describes the impact of disruption events $\vec{E}$ to the overall system performance. The particular solution $X_p(t)$ for the nonhomogeneous function is the complementary of $X_c(t)$ and is expressed as

$$X_p(t) = X(t) - X_c(t) \tag{5.5}$$

$X_p(t)$ measures the production loss caused by disruption events $\vec{E}$ and is denoted as output loss.

The dynamic EBM naturally integrates two important components: the capacity of the system (i.e. $X_c(t)$) and the impact of random disruptions to the system (i.e. $X_p(t)$). They reflect two important considerations into the decision making process: what we expect and what we actually have. Tracing and reducing the root causes of $X_p(t)$ is the goal in order to improve system performance. Next, we will develop an algorithm to evaluate $X_p(t)$ and to trace it to individual machine and supporting activity.

5.4 The systematic impact of machines and supporting activities

It has been shown in Section 5.3 that the systematic impact of each disruption event can be quantified as the permanent production time loss caused by the event. Therefore, system production loss $X_p(t)$ can be quantified using the concept of permanent production time loss caused by each disruption event. To find the relationship, we have the following Proposition.

Proposition 5.1 *Given a realization of a production process subject to a sequence of disruption events* $\vec{E} = [\vec{e}_1, \dots, \vec{e}_n]$ *within* $(0, T]$, *the system output loss* X_p *is*

$$X_p(t) = -| \cup_{i \in n^S} [t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)) | / T_{M^*} \tag{5.6}$$

where $n^S = \{i = 1, \dots, n, s.t., TL(\vec{e}_i) > 0\}$ *and* $d_i^* = \inf\{d > 0: s.t. TL(\vec{e}_i) > 0\}$.

Proof: Let's consider a disruption event $\vec{e}_i = (m_{i_1}, m_{i_2}, t_i, d_i)$, based on our earlier discussion, the last slowest machine m_{M^*} is down ($i_2 < M^*$), starved ($i_2 = M^*$) or blocked ($i_2 > M^*$) by the event at time $t_i + d_i^*$. If $d_i > d_i^*$, machine m_{M^*} will stop at time $t_i + d_i^*$ until $t_i + d_i^* + TL(\vec{e}_i)$. We thus attribute the stoppage of the last slowest machine m_{M^*} during time $[t_i +$

$d_i^*, t_i + d_i^* + TL(\vec{e}_i))$ to the disruption event. On the other hand, if $d \leq d^*$, no stoppage of the slowest machine will be resulted from the disruption event. Therefore, set of time intervals during which the last slowest machine m_{M^*} stops as a result of disruption events $\vec{E}$ can be reconstructed as follows:

$$I_{M^*} = \{\left[t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)\right), i = 1, \ldots, n, s.t., TL(\vec{e}_i) > 0\}$$

Since any stoppage of the slowest machine m_{M^*} can only result from disruption events $\vec{E}$, the last slowest machine can only stop during the time intervals contained in the set I_{M^*}. Therefore, the overall stoppage time D of the last slowest machine m_{M^*} during $(0, T]$ can be denoted as

$$D = | \cup_{i \in n^S} [t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)) \, |$$

It has been proved in Proposition 3.1 that the system output loss $X_p(t)$ can be measured as

$$X_p(t) = X(T'; E) - X(T') = -\,\mathrm{D}/T_{M^*}$$

Therefore, the system output loss $X_p(t)$ can be presented as

$$X_p(t) = -| \cup_{i \in n^S} \left[t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)\right) |/T_{M^*}$$

End of the proof.

Based on Proposition 5.1, $X_p(t)$ can be attributed to each disruption event $\vec{e}_i$ proportionally to the associate permanent production time loss $TL(\vec{e}_i)$. If a disruption event $\vec{e}_i$ does not overlap with other disruption event, the output loss due to the disruption event is $PL(\vec{e}_i) = -|[t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)|/T_{M^*} = -TL(\vec{e}_i)/\,T_{M^*}$. If $\vec{e}_i$ overlaps with other disruption events, i.e. $\exists j \neq i$, $\left[t_i + d_i^*, t_i + d_i^* + TL(\vec{e}_i)\right) \cap [t_j + d_j^*, t_j + d_j^* + TL(\vec{e}_j)) \neq \phi$, the output loss in the overlapping period will be equally shared among the corresponding events. We will use $PL(\vec{e}_i)$ to denote the output loss attributed to event $\vec{e}_i$.

The systematic impact of each machine and supporting activity can also be quantified. The permanent production loss of each disruption event can be further aggregated and attributed to the corresponding machines or supporting activities. We assume that there is a sequence of disruption events $\vec{e}_{i,1}, \dots, \vec{e}_{i,n_i}$ caused by m_i. The permanent production loss caused by $m_i, 1 \leq i \leq M + N$, can be represented as

$$PL_i = \sum_{k=1}^{n_i} PL(\vec{e}_{i,k}) \quad (5.7)$$

By using permanent production loss as a unified index, the impact to the overall system from individual machine and supporting activity can be quantified. It provides a natural severity ranking of machines and supporting activities, which is very important in real-time production control, such as prioritizing the limited resources to where they are needed most.

5.5 A Case Study

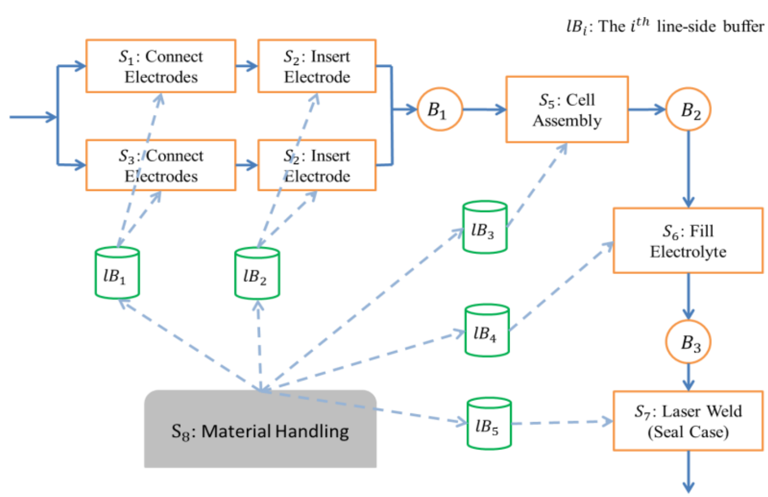


Figure 5.2 A battery assembly line with 7 machines and 3 buffers

In the case study, a battery assembly line segment is used as shown in Figure 5.2, which is based on a true battery assembly line. The assembly line consists of 7 machines that perform various operations, such as filling, assembling and welding. The empty space between adjacent machines serves as buffers where unfinished product can be temporally stored until the next machine is available. At each machine, certain parts are assembled to unfinished products. Parts are delivered from central stocking area to appropriate machines by a material handling staff and stored in a buffer beside the machine, which is called a line-side buffer $lb_i, 1 \leq i \leq 5$. The line-side buffer level decreases gradually as parts are consumed at a machine. When the line-side buffer levels reach a certain value, a requirement is sent out to the central stocking area asking for parts. The level is called a reorder point and it refers to the duration over which the remaining parts would last for assembling, such as 15 minutes.

We use a simulation model to generate production data, which otherwise is obtained from the distributed sensor system. For confidentiality reasons, we mock up the parameters of machines and buffers, which are shown in Tables 5.1 and 5.2. Mean time to repair (MTTR) and mean time between failures (MTBF) are used to generate system variation and they are assumed to be an exponential distribution. For ease of discussion, it is assumed that there is single type of part that will be assembled at each assembly machine and the number of part being assembled to the unfinished products is fixed to be 1. A material handling staff can only deliver one type of part at each trip which can fill up a line-side buffer. There are 2 material handling staffs in the central stocking area. The line-side buffer capacity, delivering time from the central stocking area to each machine and the reorder point of each line-side buffers are listed in Tables 5.3 to 5.5.

Table 5.1 Parameters of machines

Machines	Cycle time (sec/cell)	MTTR (Min)	MTBF (Min)
m_1	40	10	70
m_2	30	15	70
m_3	40	10	80
m_4	30	19	75
m_5	25	20	70
m_6	24.4	20	85
m_7	25	11	70

Table 5.2 Parameters of buffers

Buffers	Buffer capacity
b_1	72
b_2	39
b_3	39

Table 5.3 Parameters of the line-side buffers

Line-side buffers	Buffer capacity
lb_1	325
lb_2	325
lb_3	350
lb_4	375
lb_5	400

Table 5.4 Delivering time to each line-side buffer

Line-side buffers	Delivering time (Min)
lb_1	25
lb_2	25
lb_3	40
lb_4	45
lb_5	55

Table 5.5 Reorder point of each line-side buffer

Line-side buffers	Reorder point (Min)
lb_1	25
lb_2	25
lb_3	39
lb_4	43
lb_5	55

Table 5.6 Permanent production loss of each machine

Machines	Permanent production loss (Parts)
m_1	0
m_2	3
m_3	0
m_4	38
m_5	110
m_6	93
m_7	151
m_8	56

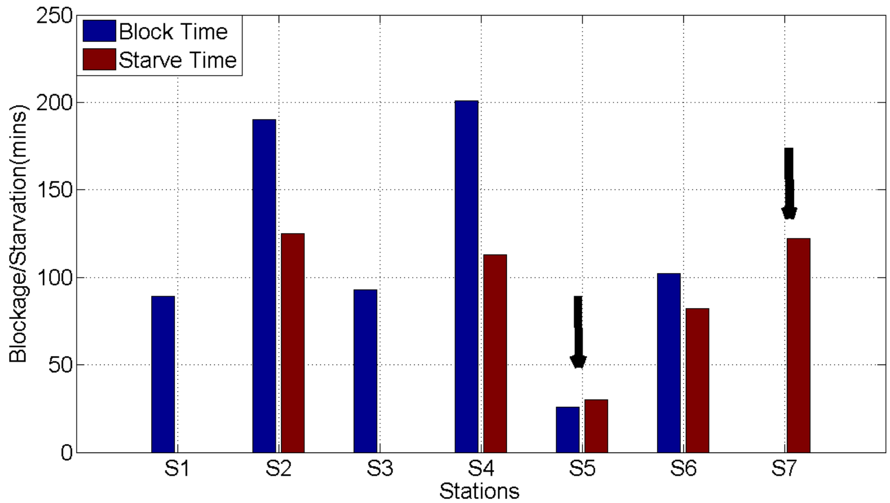


Figure 5.3 The accumulated blockage and starvation time of each machine

The simulation working time is one shift (8 hours per shift). In Table 5.6, we list the permanent production loss attributed to each machine and supporting activity (i.e. material handling in this case study). In the early works [10], the bottleneck is proven to be the machine whose upstream machines are more likely to be blocked and downstream machines are more likely to be starved. Based on the accumulated blockage and starvation time of each machine, which is shown in Figure 5.3, machines m_5 and m_7 (marked with arrow) are identified as the bottlenecks. In Table 5.6, machines m_5 and m_7 cause the most permanent production loss. The machines causing the highest permanent production loss indicates the exact same bottlenecks.

Although the bottleneck identification methods can identify all the local bottlenecks, the significance of bottlenecks is just based on heuristics. In addition, bottleneck methods do not take into account of the impact from supporting activities and quality issues, and thus may not be able to identify the real causes of system inefficiency when supporting activities or quality are big concerns.

On the other hand, the analysis based on EBM provides the severity ranking of both machines and supporting activities in terms of the permanent production loss, i.e. $m_7 > m_5 >$

$m_6 > m_8 > m_4 > m_2$, where m_8 denotes the material handling process. It can help plant floor managers to identify the machines or supporting activities that contribute the most to system output loss. This integrated model and analysis provide more detailed information for resource and budget allocation. For example, the maintenance work can be prioritized based on event or machine severity ranking.

Table 5.7 Comparison of system output improvement using policy1, policy 2 and policy 3

	Policy 1	Policy 2	Policy 3
Improvement	[178, 190]	[29, 33]	[118,123]

An alternative way to improve system output is to make certain machines produce extra products before a production shift, which is not an unusual practice in the plant floor operation. Then a question arises about which machines should work for extra time and how many extra products or extra time those machines should work. In practice, the machines and the number of extra products are determined based on the engineers' experience or heuristic rules. Now EBM provides a quantifiable solution through evaluating the permanent production loss of each machine. Specifically, machines with the highest accumulated permanent production loss should be selected and the numbers of extra products should equal to the permanent production loss of the selected machines. We further illustrate this conclusion with the following two numerical experiments.

The first numerical experiment is used to illustrate that the machines with the highest accumulated permanent production loss should be selected. Three policies are compared. In policy 1, machines m_5 and m_7, which cause the most permanent production loss, are selected. In policy 2, machines m_1 and m_4, which have the smallest standalone throughput, are selected. In policy 3, machines m_4 and m_5, which have the most accumulated downtime, are selected. The

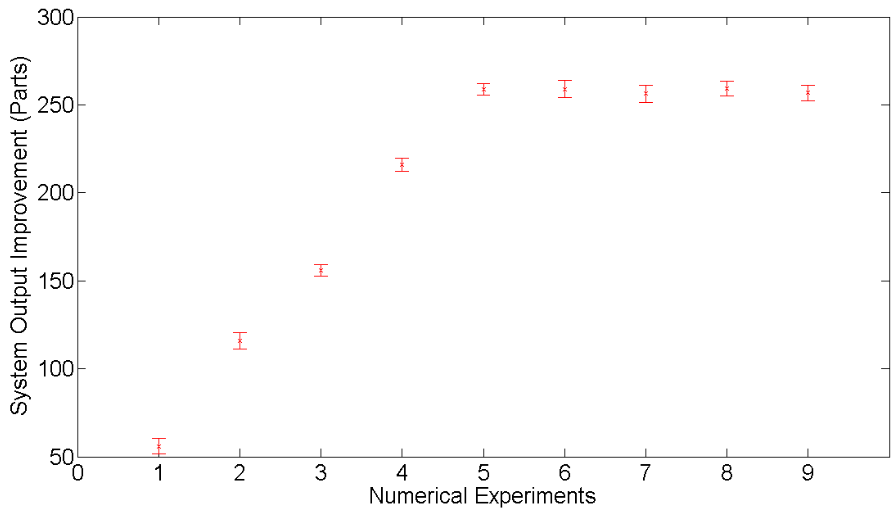


Figure 5.4 95% CI of system output improvement of each numerical experiment

selected machines in the three policies are required to produce 100 extra products for a fair comparison and sensitivity analysis. Simulation models are developed to compare the system output improvement applying the three continuous improvement policies. With 50 replications and eight hours of working time in simulation, table 5.7 illustrates the 95% confidence interval (CI) of the system output improvement of each policy. The result indicates that EBM based policy (policy 1) can lead to the largest production improvement.

The second numerical experiment is used to demonstrate that the number of extra products should be the permanent production loss of the selected machines. Consequently, the system output improvement in this scenario is expected to be the summation of the extra products produced by each selected machine. Nine experiments are tested. In the i^{th} experiment, the numbers of extra products produced on m_5 and m_7 equal to $10 + 20i$ and $1 + 30i$, $1 \le i \le 9$. Each experiment has a simulation time of one shift with 50 replications. Figure 5.4 illustrates the simulation results with 95% CI of system output improvement of each experiment. The result indicates that the system output improvement increases with the number of extra products until reaches the maximum value as in experiment 5, where the number of extra products equals to the permanent production loss of the selected machines (i.e. permanent production loss is 261 parts). Any extra parts beyond the permanent production loss (i.e. experiment $i,\ i \geq 6$) have no extra contribution to the system output improvement. It is clear that the maximum system output improvement from simulation matches the permanent production loss evaluation from EBM model. The case study also demonstrates a numerical validation of EBM model.

5.6 Conclusion

In this chapter, a sensor system with three virtual layers is developed for multi-stage manufacturing systems. The first two virtual layers convert the material flow into information flow and send the information to the third virtual layer. The sensor information is further transferred into true knowledge in the third layer through an integrated system modeling approach, i.e. EBM. Dynamic EBM naturally integrates two important system considerations, i.e. system capacity and production loss. A unified system performance index, i.e. permanent production loss caused by disruption events, machines and supporting activities provides a natural severity ranking of all the machines and supporting activities.

Chapter 6

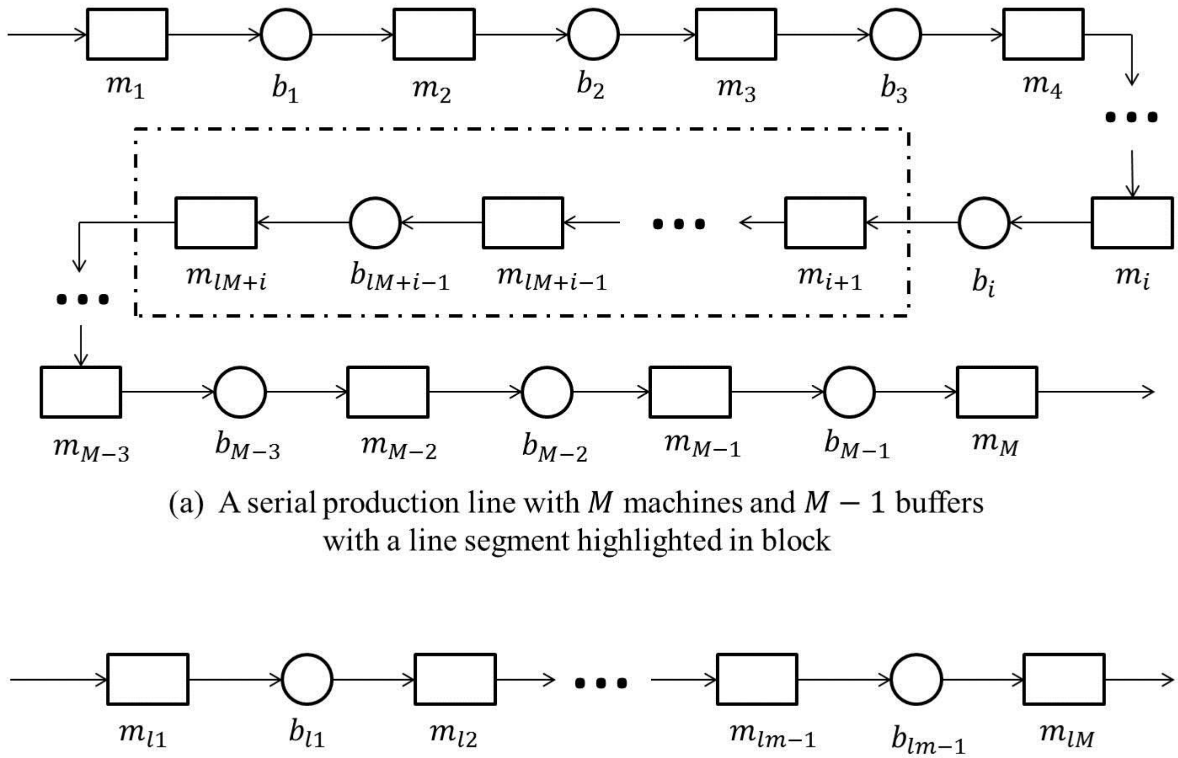
STANDALONE THROUGHPUT ANALYSIS ON THE PROPAGATION OF DISTURBANCES IN PRODUCTION SUB-SYSTEMS

6.1 Introduction

Standalone throughput (SAT) of a single machine is one of the most widely used performance indexes in industry due to its clear definition, ease of evaluation and the ability to provide a guidance for continuous improvement in production system. A complex multistage manufacturing system is typically segmented into several subsystems for efficient local management. It is important to evaluate performance of each subsystem to improve overall system productivity. However, the definition of standalone throughput of a production subsystem is not as clear as for a single machine in current literature or in practice, not to say an effective evaluation method. This chapter deals with the standalone throughput of a serial production line segment. The definition and implication of standalone throughput of a line segment is discussed. A data-driven method is developed based on online production data and is proved analytically under a practical assumption. In addition, the method is verified through simulation case studies to be an accurate and fast estimation of the standalone throughput of a production line segment.

6.2 System Descriptions and Assumptions

Continuous flow models is adopted in this chapter to analyze the dynamics of a serial production line consisting of M machines and $M-1$ buffers and a line segment with lM machines and



(a) A serial production line with M machines and $M-1$ buffers with a line segment highlighted in block

(b) A serial production line segment l within the serial production line with lM machines and $lM-1$ buffers

Figure 6.1 A serial production line with a line segment *l*

$lM-1$ buffers as shown in Figures 6.1 (a) and 6.1 (b), respectively. The following descriptions and assumptions are made.

1) Buffers $b_1, \dots, b_{M-1}$ have finite capacities, which are denoted as $B_1, \dots, B_{M-1}$. For ease of expression, we use $b_1(t), \dots, b_{M-1}(t)$ to denote the buffer levels of buffers $b_1, \dots, b_{M-1}$ at time t.
2) Machines $m_1, \dots, m_M$ has rated speeds, which are $\frac{1}{T_1}, \dots, \frac{1}{T_M}$. We use $s_1(t), \dots, s_M(t)$ to denote the speed of machines $m_1, \dots, m_M$ at time t.
3) Buffers $b_{l1}, \dots, b_{lM-1}$ in the line segment l as shown in Figure 1 have finite capacities, which are denoted as $B_{l1}, \dots, B_{lM-1}$. For ease of expression, we use $b_{l1}(t), \dots, b_{lM-1}(t)$ to denote the buffer levels of buffers $b_{l1}, \dots, b_{lM-1}$ at time t.

4) Machines $m_1, \dots, m_M$ in the line segment l as shown in Figure 1 has rated speeds, which are $\frac{1}{T_1}, \dots, \frac{1}{T_M}$. We use $s_1(t), \dots, s_M(t)$ to denote the speed of machines $m_1, \dots, m_M$ at time t.

5) The first machine m_1 can never be starved and the last machine m_M can never be blocked.

6) A machine cannot fail when it is idle.

7) m_{M^*} denotes the last slowest machine in the overall system, i.e. $M^* = \arg \min_{1 \le i \le M} \frac{1}{T_i}$.

8) $m_{M_l^*}$ denotes the last slowest machine in the line segment, i.e. $M^* = \arg \min_{1 \le i \le lM} \frac{1}{T_i}$.

9) Production line is subject to a sequence of random downtime events $E = [\vec{e}_1, \dots, \vec{e}_n]$, where $\vec{e}_i = (m_k, t_i, d_i)$ denotes a random downtime event that at time t_i, machine m_k is down for d_i.

10) Production line segment l is subject to a sequence of random downtime events $E_{ld} = [\vec{e}_{ld1}, \dots, \vec{e}_{ldn}]$, where $\vec{e}_{ldi} = (m_{lk}, t_i, d_i)$ denotes a random downtime event that at time t_i, machine m_{lk} is down for d_i.

11) The first machine m_{l1} in the line segment l is subject to a sequence of starvation events $E_{lfs} = [\vec{e}_{lfs1}, \dots, \vec{e}_{lfsn}]$, where $\vec{e}_{lfsi} = (m_{l1}, t_{lfsi}, d_{lfsi})$ denotes a starvation event at time t_{lfsi}, the first machine m_{l1} is starved for d_{lfsi}.

12) The last machine m_{lM} in the line segment l is subject to a sequence of starvation events $E_{llb} = [\vec{e}_{llb1}, \dots, \vec{e}_{llbn}]$, where $\vec{e}_{llbi} = (m_{lM}, t_{llbi}, d_{llbi})$ denotes a starvation event at time t_{llbi}, the first machine m_{lM} is starved for d_{llbi}.

6.3 Analysis of Current Disruption Events

We suppose that the serial production line, which is shown in Figure 6.1, is subjected to a sequence of disruption events $E = [\vec{e}_1, \dots, \vec{e}_n]$. Based on the analysis in Chapter 3, if an disruption event $\vec{e}_i \in E$ has $d_i > d_i^*$, the last slowest machine m_{M^*} is forced to be stopped during $(t_i + d_i^*, t_i + d_i]$. We can reconstruct the stoppage events of the last slowest machine from the downtime events with nontrivial permanent production time loss. The total time that the last slowest machine is stopped can be evaluated as

$$| \cup_{i \in n^s} (t_i + d_i^*, t_i + d_i] | \tag{6.1}$$

where $n^s = \{i = 1, \dots, n, \text{s.t.} L(\vec{e}_i; E) > 0\}$ denotes the number of disruption events that causes the last slowest machine being stopped.

During with the potential overlaps of the stoppage events constructed based on Equation 6.1 is cumbersome. It would be of interest to find a set of conditions under which the total production time loss due to a sequence of disruption events E can be expressed as the simple summation of $L(\vec{e}_i)$ and investigate how often this type of scenarios may occur. In other words, we would like to know the necessary and sufficient conditions, as relaxed as possible, for the following equality to hold.

$$|\cup_{i \in n^s} (t_i + d_i^*, t_i + d_i]| = \sum_{i=1}^{n} L(\vec{e}_i) \tag{6.2}$$

Proposition 6.1 *Given a realization of the production process subject to a sequence of downtime events* $E = [\vec{e}_1, \dots, \vec{e}_n]$ *within time interval* $(0, T]$, $(t_i + d_i^*, t_i + d_i] \cap \left(t_j + d_j^*, t_j + d_j\right] \neq \phi, i \neq j \in n^s$, *if and only if* $t_i + d_i^* = t_j + d_j^*$.

Proof: If $t_i + d_i^* = t_j + d_j^*$, then we have $(t_i + d_i^*, t_i + d_i] \cap \left(t_j + d_j^*, t_j + d_j\right] = \left(t_i + d_i^*, t_i + \min\{d_i, d_j\}\right) \neq \phi$, since $i, j \in n^s$. We will prove the sufficiency by contradiction. Suppose $t_i + d_i^* \neq t_j + d_j^*$. Let assume $t_i + d_i^* > t_j + d_j^*$ without loss of generality. Based on the definition of

d_j^*, at time $t_j + d_j^*$ the slowest machine m_{M^*} is forced to stop until at least time $t_j + d_j$. Therefore, we have

$$W(m_i) = \int_{t_i}^{t_i+d_i^*} s_{M^*}(t;E)dt = \int_{t_i}^{t_i+(t_j+d_i^*-t_i)} s_{M^*}(t;E)dt \tag{6.3}$$

This contradicts our definition of d_i^* since $t_j + d_j^* - t_i < d_i^*$.

End of the proof.

Together with Equation 6.2 and Proposition 6.1, we can easily establish the following corollary.

Corollary 6.1 *Given a realization of the production process subject to a sequence of disruption event* $E = [\vec{e}_1, \dots, \vec{e}_n]$, *within time interval* $(0, T]$, *if* $t_i + d_i^* = t_j + d_j^*, \forall i \neq j \in n^s$, *then the total stoppage time of the slowest time the last slowest machine* m_{M^*} *is* $|\bigcup_{i \in n^s}(t_i + d_i^*, t_i + d_i]| = \sum_{i=1}^{n} L(\vec{e}_i)$, *and so does the total production time loss.*

In Proposition 6.1, the condition $t_i + d_i^* = t_j + d_j^*, \forall\, i \neq j \in n^s$, is not as restrictive as it appears. First of all, two downtime events have to be overlapped, i.e., $(t_i, t_i + d_i] \cap (t_j, t_j + d_j] \neq \phi$. Moreover, if $i < M^*$ and $j > M^*$, the buffer content between machine m_i and m_{M^*} has to be exactly the same with the available buffer space between machine m_{M^*} and m_i at time $max(t_i, t_j)$; if both machines are upstream or downstream of the last slowest machine m_{M^*}, the buffers between machine m_{M^*} and m_i has to be either empty or full at time $max(t_i, t_j)$ when both machines are down. Our experience indicates that the occurrence frequency of these incidents is conceivably small and good approximation can still be achieved without considering the overlaps.

6.4 Definition and Estimation of Standalone Throughput of a Production Line Segment

For a single machine m_i, the SAT is $SAT_i = \frac{1}{T_i} A_i$, where A_i is the availability of machine m_i. The SAT of a single machine is actually the throughput of a machine when it is isolated from production line. Therefore, the SAT of a machine is not concerned with its blockage and starvation, which reflects the interaction with other machines.

Similarly, the SAT of a production line segment refers to its throughput when the line segment is isolated from its upstream and downstream. For a line segment l, as shown in Figure 6.1 (b), the interaction of l with its upstream and downstream comes from the starvation of the first machine and the blockage of the last machine. Now the problem is reduced to exclude the effects from first machine starvation and the last machine blockage of a line segment. However, the SAT of a line segment is not a simple extension from SAT_i. Given the production count PC_{lM} of the last machine in a production line segment over a time period T, one estimation used in practice is:

$$SAT_l = \frac{PC_{lM}}{T-(t^{lfs}+t^{llb})} \tag{6.4}$$

where t^{lfs} is the total amount of time for the first machine being starved and t^{llb} is the total amount of time for the last machine being blocked.

A careful examination finds that if even the starvation of the first machine is excluded from the calculation, the rest of the line segment can still process and output products through the last machine of the line segment. Similarly, the exclusion of the blockage of the last machine cannot prevent the production flow into the line segment. Therefore, simple exclusion of the total starvation from the first machine and the total blockage from the last machine does not truly isolate the production line segment. To capture the true standalone productivity of a line segment, we need to carefully evaluate the permanent production time loss caused by the starvation of the first machine and the blockage of the last machine in the line segment.

In order to analyze the impact from E_{lfs} and E_{llb} to the line segment l, we treat E_{lfs} and E_{llb} as downtime events as well. Therefore, the total downtime events on l can be expressed as

$$E_l = E_{lfs} + E_{llb} + E_{ld} \tag{6.5}$$

$E_l = [\vec{e}_{l1}, \dots, \vec{e}_{ln}]$, where we use $\vec{e}_{li} = (m_{li}, t_{li}, d_{li}), li = l1, \dots, ln$, to represent a downtime vent that at time t_{li}, machine m_{li} is down for d_{li}. To distinguish downtime events E_{lfs} and E_{llb} from E_{ld}, we call E_{lfs} and E_{llb} as generic downtime events and E_{ld} as random downtime events.

The SAT of l concerns the throughput of l in isolation from the whole system, i.e. the throughput with respect to only random downtime events $E_{ld} = [\vec{e}_{ld1}, \dots, \vec{e}_{ldn}]$. We denote $SAT_l(E_{ld})$ as standalone throughput of line segment l. Therefore, we have definition of SAT of a line segment.

Definition 6.1 *Given a realization of the production line segment l subjected to a sequence of random downtime events* $E_{ld} = [\vec{e}_{ld1}, \dots, \vec{e}_{ldn}]$ *during time period* $(0, T]$, *the SAT of the line segment l is:*

$$SAT_l(E_{ld}) = \frac{\int_0^T s_{M_l^*}(t; E_{ld}) dt}{T} \tag{6.6}$$

where $\int_0^T s_{M_l^*}(t; E_{ld})dt$ *is the production count of the slowest machine in the line segment with* E_{ld}.

However, production count $\int_0^T s_{M_l^*}(t; E_{ld})dt$ is impossible to obtain since the line segment l is integrated in the whole production system. It is advantageous to calculate SAT_l from available information such as production count $\int_0^T s_{M_l^*}(t; E_{ld})dt$, buffer levels, starvation and the blockage of machines and random downtime events.

According to conclusions in Chapter 3, the stoppage events on the last slowest machine $m_{M_l^*}$ can be reconstructed and total stoppage time is $t_l = |\cup_{i \in n^s} (t_{li} + d_{li}^*, t_{li} + d_{li})|$, where $n^s = \{li = l1, \dots, ln,\ s.t.\ d_{li} > d_{li}^*\}$. Furthermore, t_l can be attributed to $t_l = t_{lfs} + t_{llb} + t_{ld}$, where t_{lfs}, t_{llb} and t_{ld} denotes the stoppage of the last slowest machine $m_{M_l^*}$ due to E_{lfs}, E_{llb} and E_{ld}, respectively. Specifically, t_{lfs} can be expressed as $t_{lfs} = |\cup_{i \in n^{lfs}} (t_{lfsi} + d_{lfsi}^*, t_{lfsi} + d_{lfsi})|$, where $n^{lfs} = \{i = lfs1, \dots, lfsn, s.t.\ d_{lfsi} > d_{lfsi}^*\}$. Similarly, t_{llb} can be expressed as $t_{llb} = |\cup_{i \in n^{llbi}} (t_{llbi} + d_{llbi}^*, t_{llbi} + d_{llbi}]|$, where $n^{llb} = \{i = llb1, \dots, llbn, s.t.\ d_{llbi} > d_{llbi}^*\}$ and t_{ld} can be expressed as $t_{ld} = |\cup_{i \in n^{ld}} (t_{ldi} + d_{ldi}^*, t_{ldi} + d_{ldi})$, where $n^{ld} = \{i = ld1, \dots, ldn,\ s.t.\ d_{ldi} > d_{ldi}^*\}$.

Furthermore, if we assume that $t_i + d_i^* \neq t_j + d_j^*, \forall\, i \neq j$, according to Proposition 6.1, $t_{lfs} = \sum_{i \in n^{lfs}} d_{lfsi} - d_{lfsi}^*$. Similarly, $t_{llb} = \sum_{i \in n^{llb}} d_{llbi} - d_{llbi}^*$ and $t_{ld} = \sum_{i \in n^{ld}} d_{ldi} - d_{ldi}^*$. Note that t_{lfs} is the portion which causes the permanent production time loss from the total starvation time of the first machine, and t_{llb} is the portion of which caused the permanent production time loss from the total blockage time of the last machine.

Proposition 6.2 *For a production line segment* l *subject to a sequence of downtime events,* $E_l = E_{lfs} + E_{llb} + E_{ld} = \{\vec{e}_{l1}, \dots, \vec{e}_{ln}\}$ *during* $(0, T]$*, assuming* $t_i + d_i^* \neq t_j + d_j^*$*,* $\forall\, i \neq j$*, then for* $T' = T - t_{lfs} - t_{llb}$*,* $\exists\, T^*, \forall\, T' > T^*$*, the SAT of the line segment* l *is*

$$SAT_l = \frac{\int_0^T s_{M^*}(t; E_l)dt}{T'} \tag{6.7}$$

Proof: The line segment l is subject to $E_l = E_{lfs} + E_{llb} + E_{ld}$, where E_{lfs} and E_{llb} are generic downtime events, and E_{ld} are random downtime events. The production count of machine $m_{M_l^*}$ is

$$PC_{M_l^*} = \int_0^T s_{M_l^*}(t; E_l)dt = \left(T - t_{ld} - t_{lfs} - t_{llb}\right)\frac{1}{T_{M^*}} \tag{6.8}$$

For $T' > T^*$, the up time ratio of machine $m_{M_l^*}$ over T' is $U_{M_l^*} = \frac{T' - t_{ld}}{T'}$ and $U_{M_l^*}$ is a constant. Therefore, from Equation 6.8, we have

$$U_{M_l^*} = \frac{T' - t_{ld}}{T'} = \frac{(T' - t_{ld})\frac{1}{T_{M_l^*}}}{T'\frac{1}{T_{M_l^*}}} = \frac{\int_0^T s_{M_l^*}(t; E_l)dt}{T'} T_{M_l^*} \tag{6.9}$$

Assuming a virtual scenario that the line segment is subjected to only random downtime events E_{ld}^v during time period $(0, T^v]$, then the up time ratio of the machine $m_{M_l^*}$ over time T^v is still $U_{M_l^*}$, $\forall T^v > T^*$, we have $U_{M_l^*} = \frac{T^v - t_{ld}^v}{T^v} = \frac{T' - t_{ld}^v}{T'}$.

In this virtual scenario, the production count $PC_{M_l^*}^v$ of the machine $m_{M_l^*}$ is

$$PC_{M_l^*}^v = \int_0^T s_{M_l^*}(t; E_{ld}^v)dt = (T^v - t_{ld}^v)\frac{1}{T_{M_l^*}} \tag{6.10}$$

According to Definition 6.1, the SAT of the line segment l in this virtual scenario can be represented as

$$SAT_l = \frac{\int_0^T s_{M_l^*}(t; E_{ld}^v)dt}{T^v} = \frac{T^v - t_{ld}^v}{T^v} \frac{1}{T_{M_l^*}} \quad (6.11)$$

Therefore,

$$U_{M_l^*} = SAT_l \cdot T_{M_l^*} \quad (6.12)$$

Comparing Equations 6.9 and 6.12, SAT of a line segment l becomes

$$SAT_l = \frac{\int_0^T s_{M_l^*}(t; E_l)dt}{T'} \quad (6.13)$$

End of the proof.

Proposition 6.2 provides us a method to estimate the SAT of line segment l with online data. In Equation 6.7, production count $\int_0^T s_{M_l^*}(t; E_l)dt$ can be obtained directly from online information. T', on the other hand, can be calculated based on information of buffer levels and starvation and blockage of machines. According to the definition of d_l^*, we can define the permanent production time loss of the line segment l due to the starvation event $\vec{e}_{lfsi}$ of the first machine m_{l1} as

$$L_{lfs}(\vec{e}_{lfsi}; E_{lfs}) = \begin{cases} d_{lfsi} - d_{lfsi}^*, & d_{lfsi} > d_{lfsi}^* \\ 0, & d_{lfsi} \leq d_{lfsi}^* \end{cases} \quad (6.14)$$

where d_{lfsi}^* can be evaluated as the time for the buffers between the first machine and m_M^* to just become empty. Therefore, t_{lfs} can be represented as

$$t_{lfs} = \sum_{lfsi=lfs1}^{lfsn} L_{lfs}(\vec{e}_{lfsi}; E_{lfs}) \quad (6.15)$$

Similarly, we define the permanent production loss of the line segment l due to the blockage event $\vec{e}_{llbi}$ of the last machine as

$$L_{llb}(\vec{e}_{llbi}; E_{llb}) = \begin{cases} d_{llbi} - d^*_{llbi}, & d_{llbi} > d^*_{llbi} \\ 0, & d_{llbi} \leq d^*_{llbi} \end{cases} \tag{6.20}$$

where d^*_{llbi} can be evaluated as the time for the buffers between the last machine and m_{M^*} to just become full. Therefore, t_{llb} can be represented as

$$t_{llb} = \sum_{llbi=llb1}^{llbn} L_{llb}(\vec{e}_{llbi}; E_{llb}) \tag{6.21}$$

Then T' is calculated as $T' = T - t_{lfs} - t_{llb}$.

6.5 Application of Estimation Method and Simulation Results

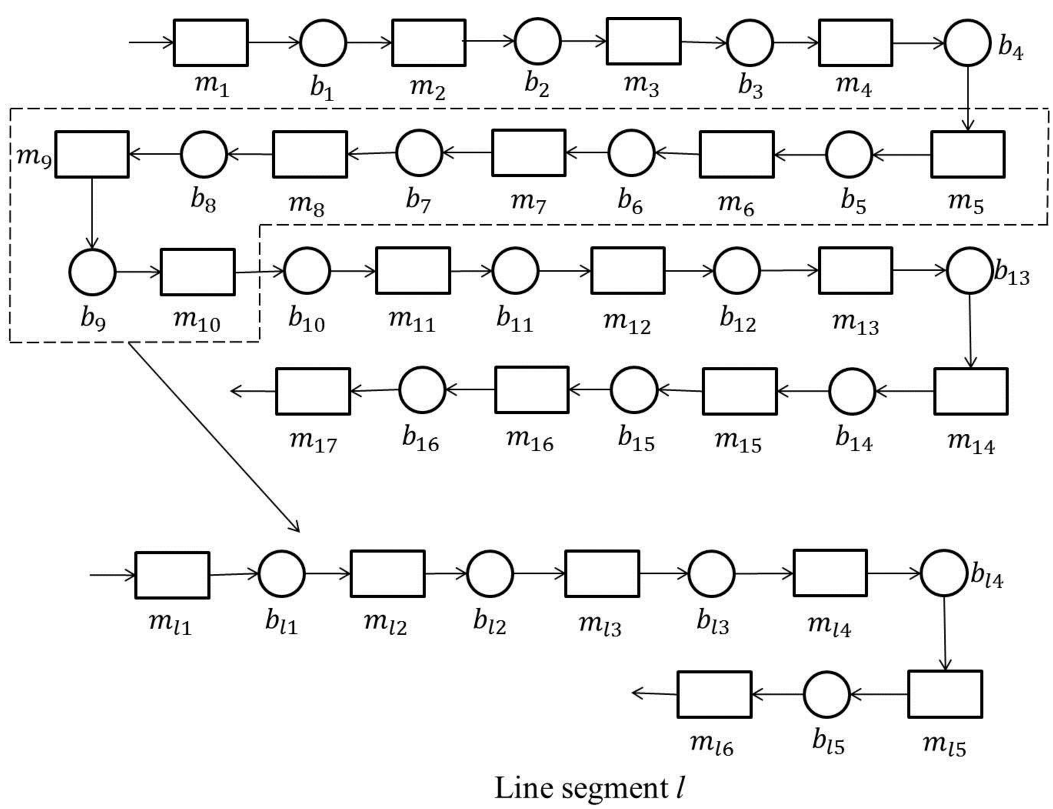


Figure 6.2 A serial production line with a line segment l

Simulation case studies are based on a true automotive machining line, as shown in Figure 6.2. We mock up the parameters of the production line for confidential consideration. In order to verify the SAT calculation in Equation 6.7, an isolated simulation model is generated only for line segment $\boldsymbol{l}$. The whole production line has seventeen machines and sixteen buffers. The line segment **l** includes six machines and five buffers. All machines are serially connected.

Table 6.1 Parameters of machines in the production system in case 1

Machines	Cycle time (Parts/min)	MTTR (Min)	MTBF (Min)
m_1	20	3	33.91
m_2	20	4	10
m_3	20	8	20
m_4	20	6	11
m_5	20	4	12
m_6	20	6	12
m_7	20	6	13
m_8	19	2.5	2
m_9	20	4.4	6.4
m_{10}	20	7.8	24
m_{11}	20	2.6	9
m_{12}	20	4.4	10
m_{13}	20	3.7	8
m_{14}	20	3	6.5
m_{15}	20	2.2	23.62
m_{16}	20	11.5	10
m_{17}	20	7.9	20

Table 6.2 Parameters of buffers in the production system in case 1

Buffers	Buffer capacity	Initial buffer level
b_1	10	5
b_2	35	18
b_3	25	13
b_4	10	5
b_5	10	5
b_6	10	5
b_7	10	5
b_8	10	5
b_9	10	5
b_{10}	10	5
b_{11}	115	58
b_{12}	10	5
b_{13}	100	50
b_{14}	10	5
b_{15}	10	5
b_{16}	10	5

Table 6.3 Parameters of machines in the line segment *l* in case 1

Machines	Cycle time (Parts/min)	MTTR (Min)	MTBF (Min)
m_{l1}	20	4	12
m_{l2}	20	6	12
m_{l3}	20	6	13
m_{l4}	19	2.5	2
m_{l5}	20	4.4	6.4
m_{l6}	20	4	24

Table 6.4 Parameters of buffers in the line segment *l* in case 1

Buffers	Buffer capacity	Initial buffer level
b_{l1}	10	5
b_{l2}	10	5
b_{l3}	10	5
b_{l4}	10	5
b_{l5}	10	5

The initial parameters of the production line in case 1 are shown in Table 6.1 and Table 6.2, and the parameters of the line segment $\boldsymbol{l}$ are shown in Table 6.3 and Table 6.4. For each scenario, we linearly increase the values of MTTR, buffer capacities, and initial buffer capacities. Note that the parameters in the first scenario as shown in Tables 6.1 to 6.4, are mocked up based on the true automotive machining line for confidential reason. The other 8 scenarios are based on simulation data to further demonstrate the effectiveness of the estimation method.

Specifically, in scenario j, the MTTR of machine m_i in the production line.

$$MTTR_i(k) = \begin{cases} MTTR_i(1) + (i-1), & m = 3,4,6,7,10,16,17 \\ MTTR_m(1) + 2(i-1), & m \neq 3,4,6,7,10,16,17 \end{cases}$$

The capacity of buffer B_m in the production line is

$$B_m(i) = \begin{cases} B_m(1) + (i-1), & m = 2,6,11,13 \\ B_m(1), & m \neq 2,6,11,13 \end{cases}$$

The MTTR of machine m_{lm} in the line segment is

$$MTTR_{lm}(i) = \begin{cases} MTTR_{lm}(1) + (i-1), & lm = 2,3,6 \\ MTTR_{lm}(1) + 2(i-1) & lm \neq 2,3,6 \end{cases}$$

The capacity of buffer B_{lm} in the line segment is

$$B_{lm}(i) = \begin{cases} B_{lm}(1) + (i-1), & lm = 2 \\ B_{lm}(1), & lm \neq 2 \end{cases}$$

Table 6.5 Comparison of 95% confidence interval of SAT for the line segment l using estimation method and simulation result

Scenario	SAT (Estimation)	SAT (Simulation)
1	[5.6161, 5.7557]	[5.6594, 5.8243]
2	[4.0662, 4.1684]	[4.0682, 4.2326]
3	[3.1951, 3.3134]	[3.1850, 3.3353]
4	[2.6288, 2.7497]	[2.6437, 2.7701]
5	[2.2166, 2.3173]	[2.2282, 2.3445]
6	[2.0259, 2.1093]	[2.0097, 2.1048]
7	[1.7057, 1.7694]	[1.6819, 1.7930]
8	[1.5119, 1.5803]	[1.5048, 1.5973]
9	[1.3737, 1.4379]	[1.3490, 1.4355]

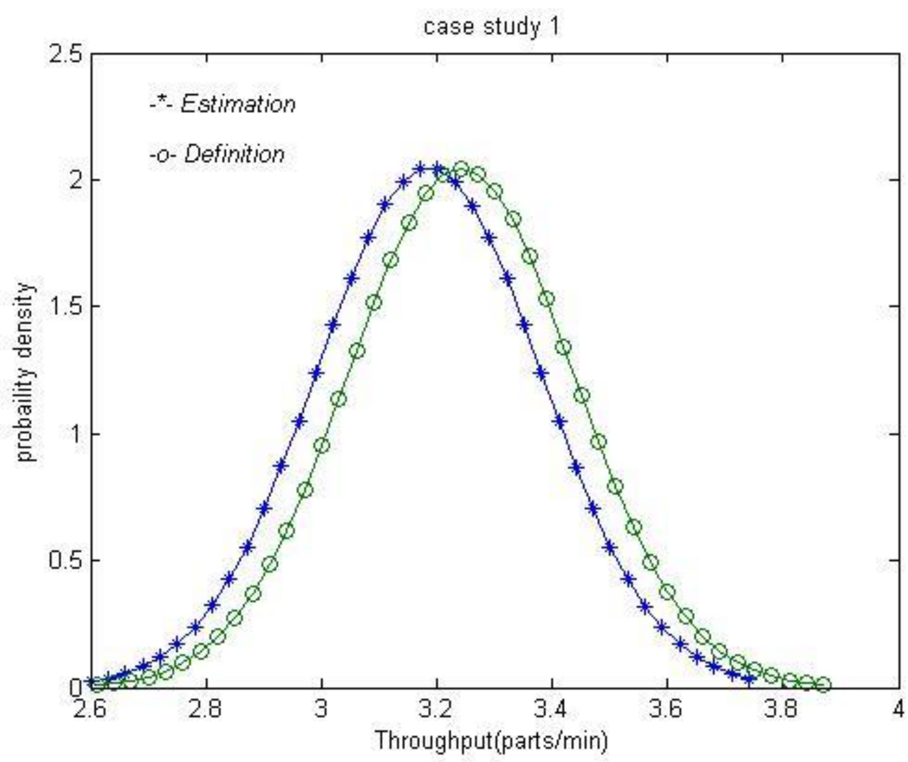


Figure 6.3 Estimation using Equation 6.7 vs. Definition 6.1 in case 1

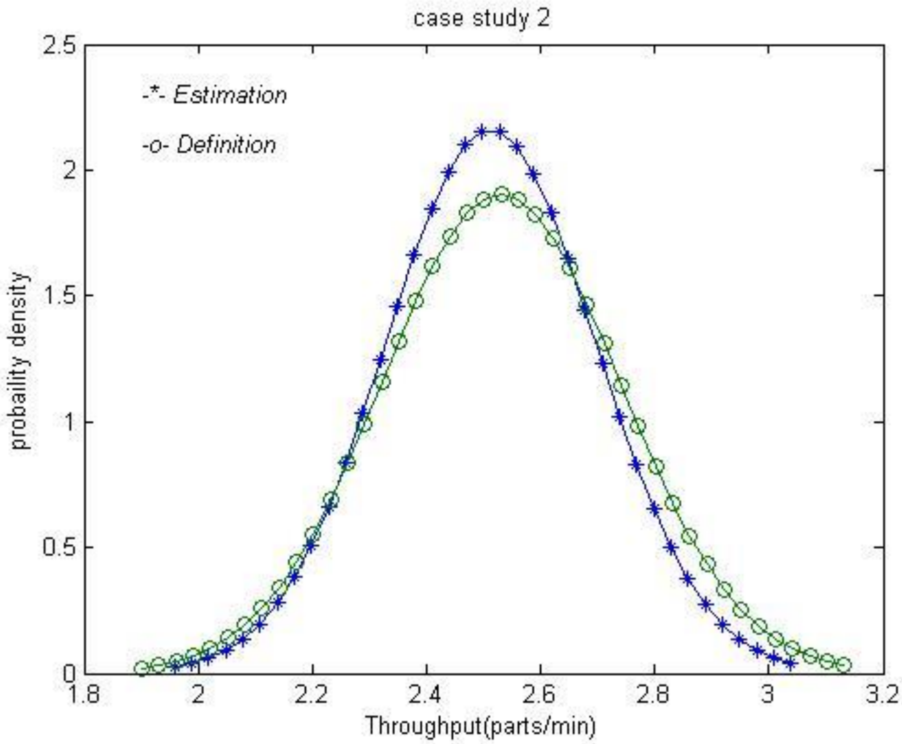


Figure 6.4 Estimation using Equation 6.7 vs. Definition 6.1 in case 2

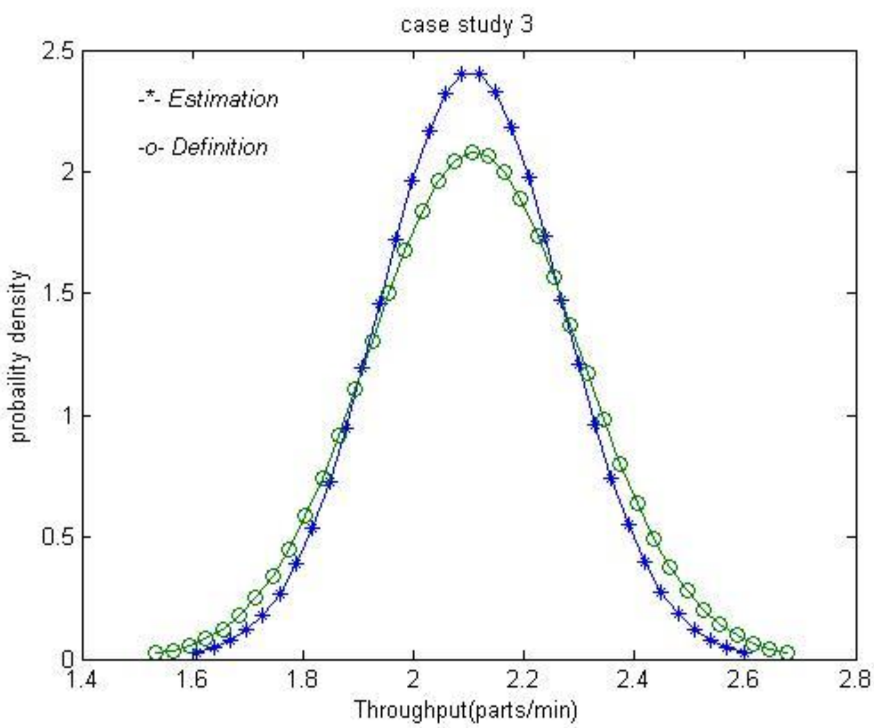


Figure 6.5 Estimation using Equation 6.7 vs. Definition 6.1 in case 3

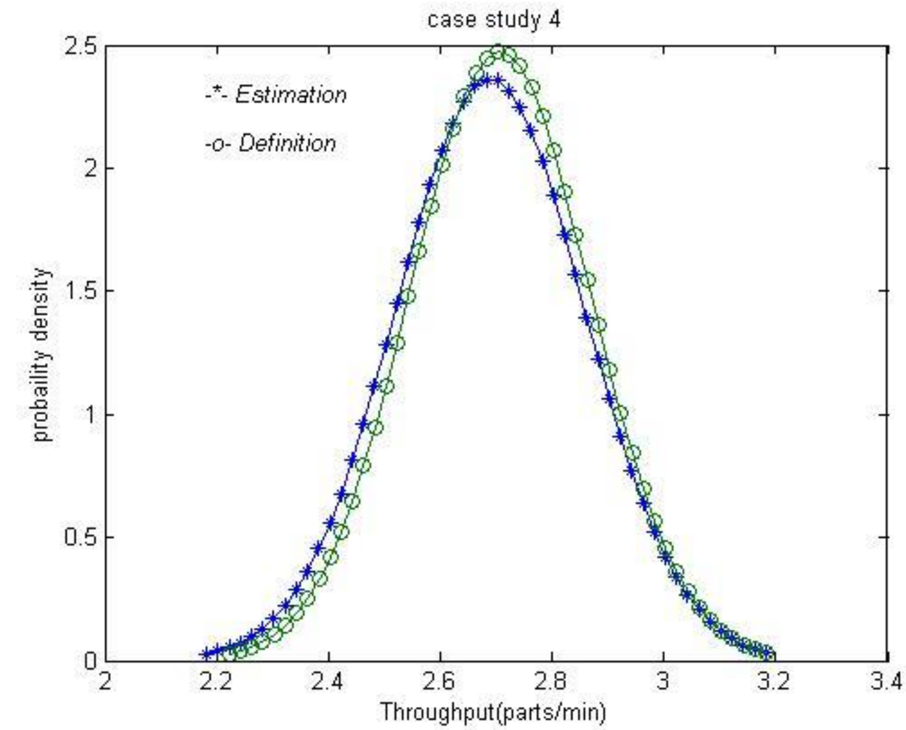


Figure 6.6 Estimation using Equation 6.7 vs. Definition 6.1 in case 4

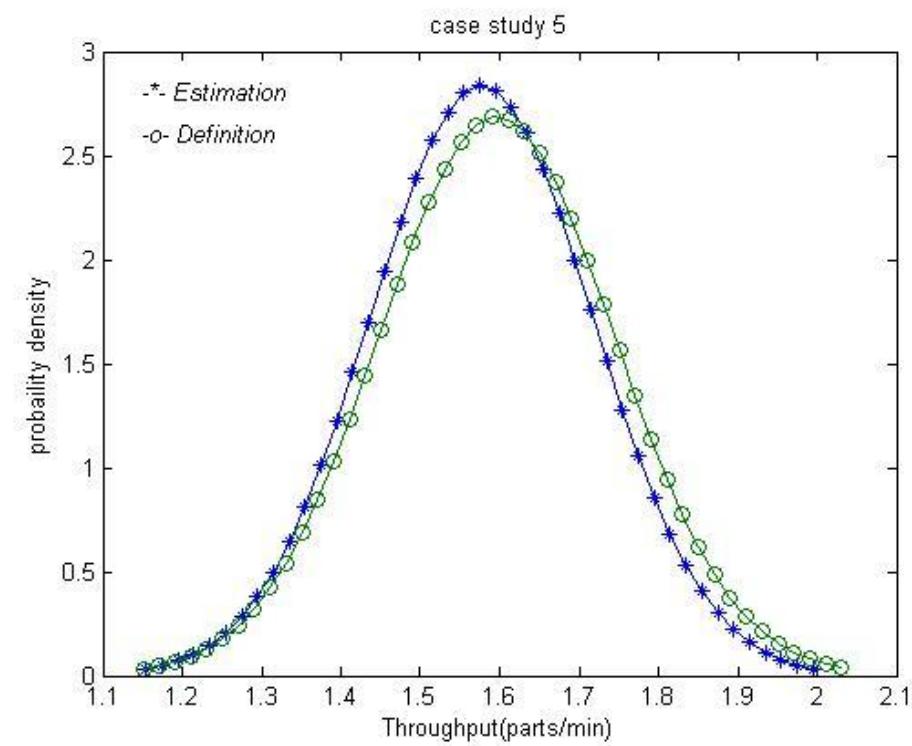


Figure 6.7 Estimation using Equation 6.7 vs. Definition 6.1 in case 5

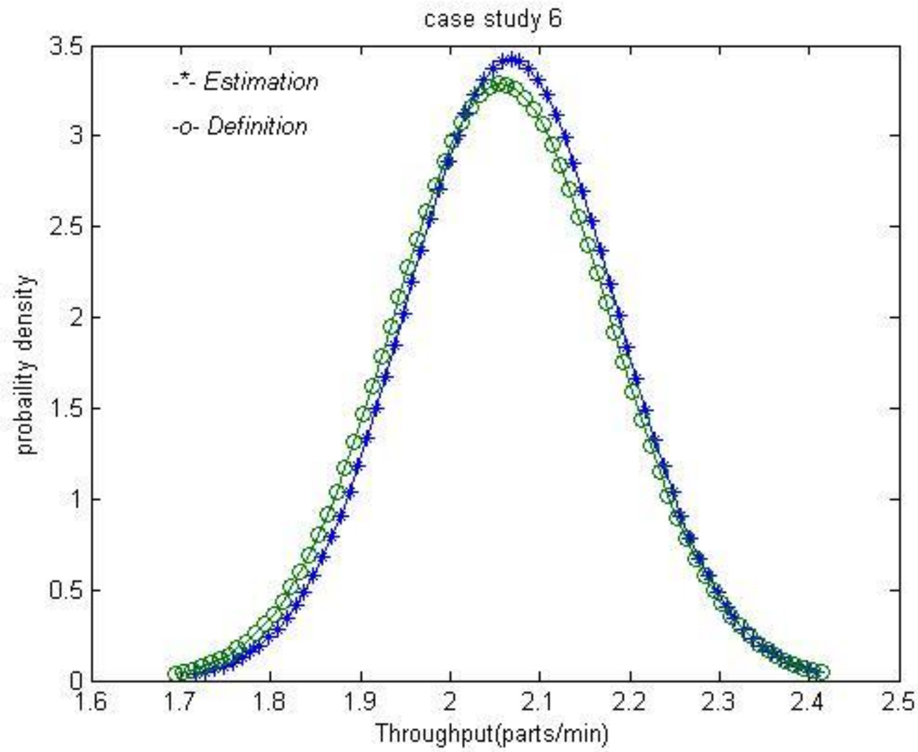


Figure 6.8 Estimation using Equation 6.7 vs. Definition 6.1 in case 6

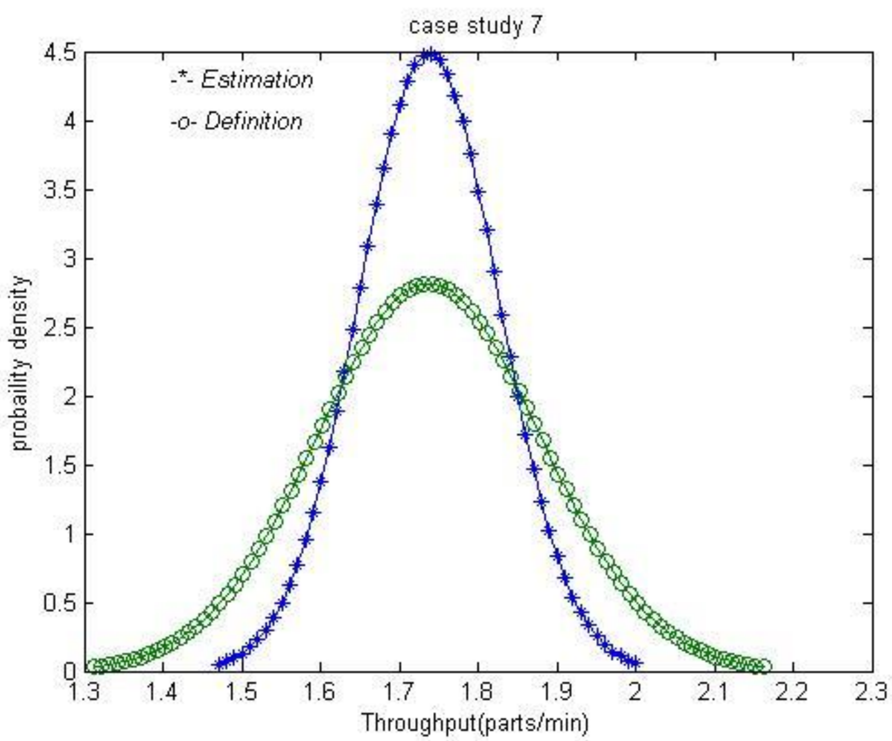


Figure 6.9 Estimation using Equation 6.7 vs. Definition 6.1 in case 7

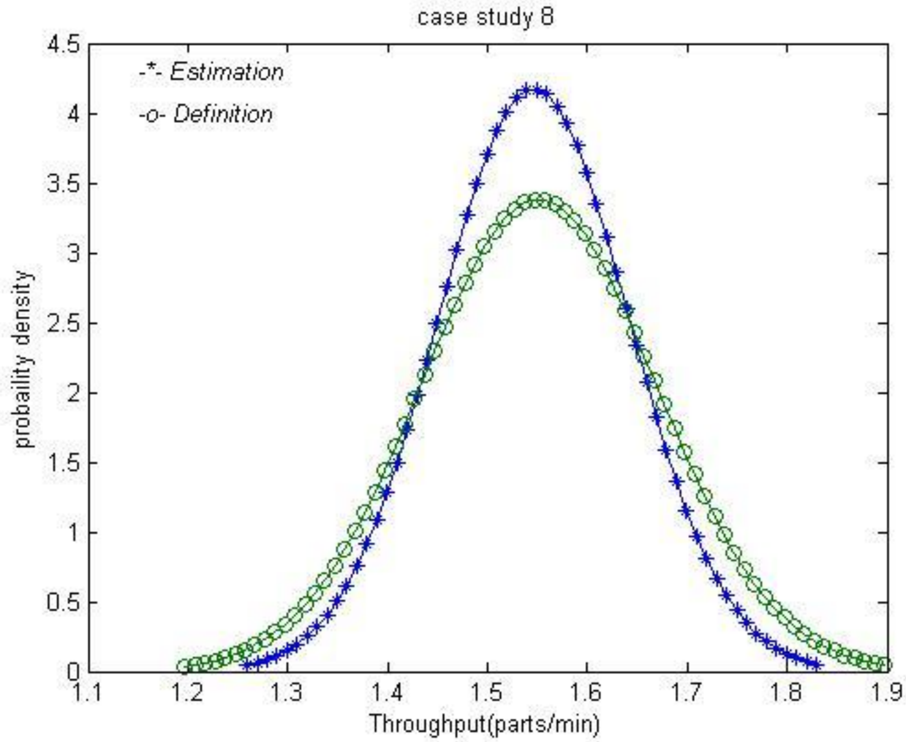


Figure 6.10 Estimation using Equation 6.7 vs. Definition 6.1 in case 8

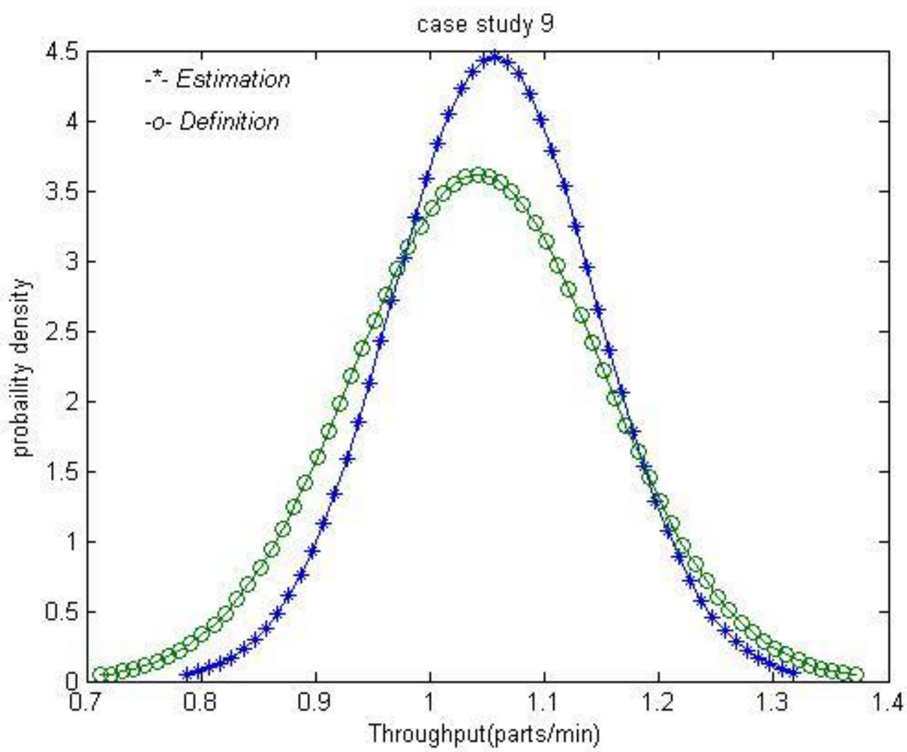


Figure 6.11 Estimation using Equation 6.7 vs. Definition 6.1 in case 9

In each scenario, we compare the estimation method in Equation 6.7 and the simulation results. The simulation setup is 50 hours warm-up period and 24 hours running time with 35 replications. Table 6.5 demonstrates the results with 95% confidence interval for nine scenario studies. Figure 6.3 to Figure 6.11 illustrates the comparison of the estimation method and simulation.

The comparisons show close agreement between the estimation method and simulation results. For scenarios from high availability of machines to low availability of machines (lower than 40%), the estimation method consistently evaluate the SAT of a line segment with high accuracy.

6.6 Conclusion

This paper provides the definition of and an estimation algorithm for SAT of a serial production line segment. The estimation method is analytically proven under the assumption of no concurrent downtime events. Based on the fact that concurrent downtime events are very rare, the estimation method can be used to evaluate the SAT of a line segment with high accuracy. Our case studies further confirm this conclusion. The estimation method provides a data-driven algorithm through utilizing production online data to quickly and accurately evaluate SAT of a serial line segment in a production system. It provides important insights in understanding production line dynamics. This quantitative tool can help production managers better identify problems and improve overall system performance. Our future work will extend the knowledge of serial production line to more complex production line segments, such as parallel line and loop back configuration.

Chapter 7

MARKET DEMAND-ORIENTED MODELING AND CONTROL OF MANUFACTURING SYSTEMS

7.1 Introduction

Advanced manufacturing system must quickly ramp up the newly developed technologies, tools, equipment, and resources if it is to meet production needs for a variety of machinery and motive applications. Such systems are subject to frequent new/improved technology insertions, dramatically fluctuating market demands, and engineering modification for new processes. These challenges greatly impact the productivity of the manufacturing [4]. Therefore, improving the production throughput to satisfy market demands have become a critical issue for many manufacturing industries.

Battery manufacturing systems could be typical examples of such systems. A typical battery production system consists of four functional areas: powder production, tube fabrication, cell assembly and battery assembly. In powder production, raw materials are mixed according to a certain proportion. Then, the powders are pressed and shaping into tubes through machining processes and the tubes are assembled into cells with covers, electrodes and other components. The finished cells are connected through welding (e.g., laser welding) and assembled into modules. Several modules are assembled into a battery pack and stored in finished-goods buffers. The batteries packs are delivered to customers based on market demand. Battery production lines typically include multistage manufacturing processes with combined manual and automatic operations. For example, loading/unloading operations are manual and welding or pressing operations are automatic.

There exhibit unique features in a multistage battery manufacturing system. Firstly, machines at each process stage may have different input and output rates. For example, the tube pressing machine could use 500g powder to produce one tube and the battery assembly machine could take nine cells to produce one battery pack during a machine cycle. Secondly, a battery production line can be highly unsynchronized with different process stages combining both manual and automatic operations and therefore having large variance in process rates. In this situation, capacity differences among machines as well as machine cycle time variance are major constrains that restrict battery production line throughput. Besides this, machine random downtime also impacts production efficiency. A new system modeling and in-depth analysis are needed to continuously improve customer demand satisfaction and system throughput with the above features. This chapter is devoted to this end.

7.2 Market Demand Driven System Descriptions, Assumptions and Notations

In this paper, we consider a continuous-flow production system consisting of M machines (represented as rectangles) and $M-1$ in-process buffers (represented as circles) as shown in Figure 7.1. Machine m_i, $1 \leq i \leq M$, denotes the i^{th} machines. Buffer b_i, $1 \leq i \leq M-1$, denotes the i^{th} in-process buffers. Buffer b_M represents the finished-goods buffer and virtual machine m_{M+1} represents a market demand. The following definitions and assumptions on a market demand driven system are adopted.

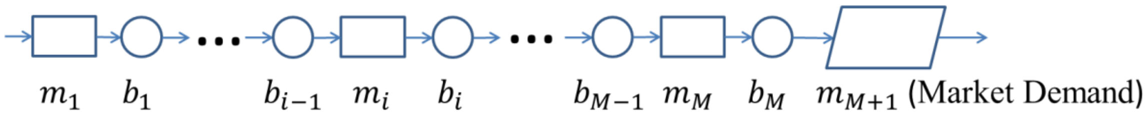


Figure 7.1 A production line and its market demand

1) Although the market demand of a production line varies with time, it is constant during a certain time period T, such as a quarter or a month. Therefore, in this research, market demand is assumed to be a sequence of fixed values described by $\{cd_1, \dots, cd_n\}$ (parts/unit time), which are denoted as market demand rates. In addition, it is assumed $\forall cd_i \in \{cd_1, \dots, cd_n\}$, $cd_i \cdot T > OP_{sys}(T)$, i.e. market demand is assumed to be higher than production count, where $OP_{sys}(T)$ is the production count (or productivity) of the system during $(0, T]$.Virtual machine m_{M+1} has identical rated input and output rates, i.e. $s_{M+1}^{in} = s_{M+1}^{out} = cd_i$ and it cannot be down.
2) Each buffer $b_i, i = 1, \dots, M$, has a finite buffer capacity, which is denoted as B_i. For simplicity, $b_i(t)$ is used to denote the buffer level of buffer b_i at time t.
3) A production line consists of both manual and automatic machines. Each automatic machine $m_i, 1 \leq i \leq M$, has a constant cycle time T_i and is characterized by one up state and one down state, which are denoted as $\alpha_i = 1$ and $\alpha_i = 0$. Automatic machine m_i obeys the exponential reliability model. When the machine is in its up state, it could transit into down state due to a machine random failure event with a transition rate p_i and when it is down, it could be repaired with a transition rate r_i. Thus, the mean time between failure (MTBF) and mean time to repair (MTTR) of the machine are $\frac{1}{p_i}$ and $\frac{1}{r_i}$.

   Manual machines do not have fixed cycle time. However, for each manual machine $m_i, 1 \leq i \leq M$, it is always possible to find the smallest cycle time T_i, which is referred to as the base cycle time. Any production cycle that lasts longer than the base cycle time will be treated as a special machine random failure event, referred to as an over-cycle event. Manual machine m_i has one up state and one down state, which are denoted as $\alpha_i = 1$ and $\alpha_i = 0$. It obeys the exponential reliability model. When the machine is up, it can transit into down state due to an over-cycle event with a transition rate p_i, and when the machine is down, the over-cycle event could finish with a transition rate r_i. The mean time between failure (MTBF) and mean time to repair (MTTR) of the machine are $\frac{1}{p_i}$ and $\frac{1}{r_i}$. For simplicity, we refer random failure events to as the events causing both automatic machine and manual machine down state for the rest of the paper.

4) Raw material A_0 enters the system from machine m_1. Each machine m_i, $i = 1, \dots, M$, receives λ_i parts of A_{i-1} from the previous buffer b_{i-1} to produce one part of A_i and enters the following buffer b_i, where A_i refers to the part produced by machine m_i.
5) If machine $m_i, i = 1, \dots, M$, is up and not starved or blocked, it has a rated input rate s_i^{in} and a rated output rate s_i^{out}, where $s_i^{in} = \frac{\lambda_i}{T_i}$ and $s_i^{out} = \frac{1}{T_i}$. Additionally, we use $v_i^{in}(t)$ and $v_i^{out}(t)$ to denote the instantaneous input and output rates of machine m_i at time t, where $v_i^{in}(t) \le s_i^{in}$ and $v_i^{out}(t) \le s_i^{out}$.
6) The first machine m_1 is never starved and the last virtual machine m_{M+1} is never blocked.
7) A machine $m_i, 1 < i \le M + 1$, is completely starved by machine $m_j, 1 \le j < i$, if the following conditions hold: machine m_i is up, machine m_j is down and all the buffers in between are empty, i.e. $b_j = \cdots = b_{i-1} = 0$. A machine cannot be down when it is completely starved.
8) A machine $m_i, 1 < i \le M + 1$, is partially starved by machine $m_j, 1 \le j < i$, if the following

conditions hold: machine m_i is up, machine m_j is up and not starved, the machines' output rates satisfy $\frac{s_j^{out}}{\prod_{r=j+1}^{i} \lambda_r} < s_i^{out}$ and all the buffers in between are empty, i.e. $b_j = \cdots = b_{i-1} = 0$. When machine m_i is partially starved by machine m_j, it produces at a reduced output rate $v_i^{out} = \frac{s_j^{out}}{\prod_{r=j+1}^{i} \lambda_r}$.

9) A machine $m_i, 1 \leq i < M+1$, is completely blocked by machine $m_j, i < j \leq M+1$, if the following conditions hold: machine m_i is up, machine m_j is down and all the buffers in between are full, i.e. $b_i = B_i, \ldots, b_{j-1} = B_{j-1}$. A machine cannot be down when it is completely blocked.

10) A machine $m_i, 1 \leq i < M+1$, is partially blocked by machine $m_j, i < j \leq M+1$, if the following conditions hold: machine m_i is up, machine m_j is up and not blocked, the machines' output rates satisfy $s_j^{out} \prod_{r=i+1}^{j} \lambda_r < s_i^{out}$ and all the buffers in between are full, i.e. $b_i = B_i, \ldots, b_{j-1} = B_{j-1}$. When machine m_i is partially blocked by machine m_j, it produces at a reduced output rate $v_i^{out} = s_j^{out} \prod_{r=i+1}^{j} \lambda_r$.

11) $\vec{e}^p = (m_i,\ t^p,\ d^p)$ denotes a disruption event that the end-of-line virtual machine m_{M+1} is partially starved by m_i at time t^p for d^p. Since such disruption event is caused by machine capacity differences between m_i and m_{M+1} (asynchronous), it is defined as a disruption event caused by machine capacity (DEMC).

12) $\vec{e}^c = (m_i,\ t^c,\ d^c)$ denotes a disruption event that the end-of-line virtual machine m_{M+1} is completed starved by m_i at time t^c for d^c. Since such disruption event is caused by machine random failure event, it is defined as a disruption event caused by machine failure (DEMF).

Remark 2.1 By modeling the market demand as the end-of-line virtual machine m_{M+1}, the market demand (MD) and the system output (OP_{sys}) during $(0, T]$ can be measured as the output of the end-of-line virtual machine without and with disruption events, i.e. $MD = T \cdot cd_i$ and $OP_{sys} = \int_0^T v_{M+1}^{out}(t)dt$. Market demand dissatisfaction (MDD) is defined as the number of parts that system fails to satisfy customer demand, i.e., $MDD = MD - OP_{sys}$, which is the production loss at the end-of-line virtual machine m_{M+1}. The problem of market demand satisfaction is transferred into the problem of system productivity analysis.

Remark 2.2 Note that assumption 2 assumes a machine m_i uses λ_i parts of A_{i-1} to produce a part A_i. This implies that the input and output rates of machine m_i follow a relationship of $\frac{v_i^{in}(t)}{v_i^{out}(t)} = \frac{s_i^{in}}{s_i^{out}} = \lambda_i$.

Remark 2.3 When machine m_i is partially starved by machine m_j, machine m_i has a reduced speed $\frac{s_j^{out}}{\prod_{r=j+1}^{i} \lambda_r}$, where $\prod_{r=j+1}^{i} \lambda_r$ denotes the number of A_j required to produce a A_i. Similarly, when machine m_i is partially blocked by machine m_j, machine m_i has a reduced speed $s_j^{out} \prod_{r=i+1}^{j} \lambda_r$, where $\prod_{r=i+1}^{j} \lambda_r$ denotes the number of A_i required to produce a A_j.

7.3 Stochastic System Model

Since the exact analytical solution only exists for a two-machine one-buffer system, approximate method is necessary to estimate the performance of general complex production systems. In this Section, a decomposition technique presented in [67] is utilized to decompose an $M + 1$-machine system into a set of M two-machine one-buffer subsystems $l_i, i = 1, \dots, M$, as shown in Figure 7.2. Each subsystem l_i is characterized by one upstream pseudo-machine m_i^u, one downstream

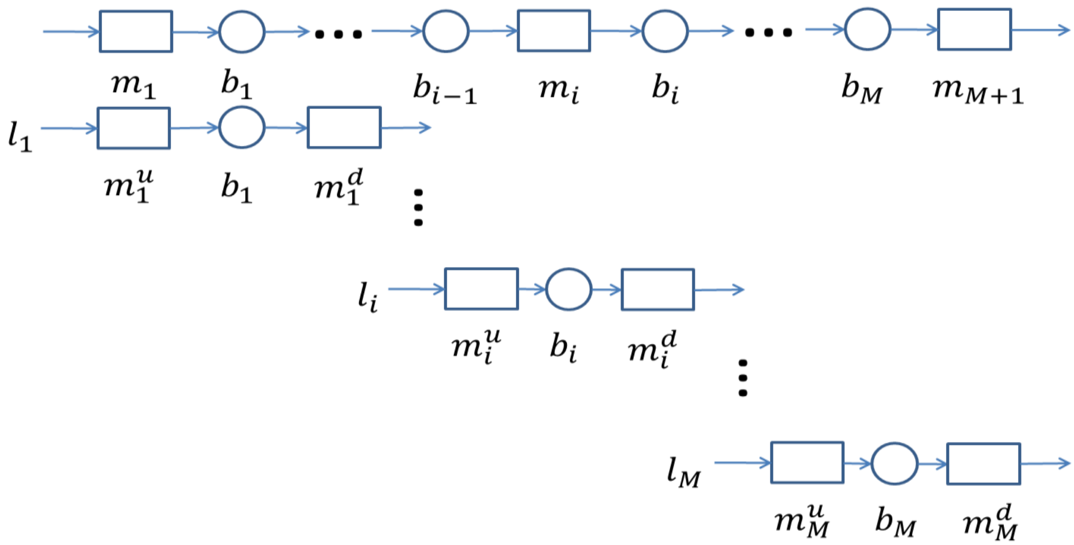


Figure 7.2 Decomposition of a production line with M+1 machines

pseudo-machine m_i^d and a buffer b_i in between, where m_i^u and m_i^d model the upstream and downstream systems of buffer b_i. Each pseudo-machine m_i^u can have local states LS_i^udenoting machine m_i is not starved, and remote states RS_i^u denoting the machine m_i is starved. Similarly, pseudo-machine m_i^d can have local states LS_i^d and remote states RS_i^d, which denote the states when machine m_i is not blocked and blocked. To evaluate the performance measure of the system such as throughput, unknown parameters need to be determined, which include the output rate of pseudo-machine m_i^u and input rate of pseudo-machine m_i^d at each state and the transition rate matrices of pseudo-machines.

In each subsystem l_i, we use α_i^u and α_i^d to denote the state of pseudo-machines m_i^u and m_i^d. When pseudo-machine m_i^u is in a local state, i.e. $\alpha_i^u \in LS_i^u$, state α_i^u is determined by the states of machine m_i, i.e. $\alpha_i^u = \alpha_i$. The rated output rate of pseudo-machine m_i^u is denoted as $\mu_i^{out}(\alpha_i)$, where $\mu_i^{out} = 0$ if $\alpha_i = 0$, and $\mu_i^{out} = s_i^{out}$ if $\alpha_i = 1$. When pseudo-machine m_i^u is in a remote state, i.e. $\alpha_i^u \in RS_i^u$, machine m_i is partially or completely starved by machine $m_j, 1 \leq$

$j < i$. The remote state α_i^u can be described by the local machine state α_i, the state of pseudo-machine m_{i-1}^u in the upstream subsystem l_{i-1} and buffer b_{i-1}, i.e. $\alpha_i^u = \{b_{i-1}, \alpha_{i-1}^u, \alpha_i\}$. The rated output rate μ_i^{out} is determined by the local states of machine m_j. If machine m_i is up and partially starved by machine m_j, then $\mu_i^{out} = \frac{s_j^{out}}{\prod_{r=j+1}^{i} \lambda_r}$. If machine m_i is up and is completely starved by machine m_j, then $\mu_i^{out} = 0$.

The pseudo-machine m_i^d can be similarly defined. When m_i^d is in a local state, i.e. $\alpha_i^d \in LS_i^d$, state α_i^d is determined by the local states of machine m_{i+1}, i.e. $\alpha_i^d = \alpha_{i+1}$. The rated input rate of pseudo-machine m_i^d is denoted as $\omega_i^{in}(\alpha_{i+1})$, where $\omega_i^{in} = 0$ if $\alpha_{i+1} = 0$, and $\omega_i^{in} = s_{i+1}^{in}$ if $\alpha_{i+1} = 1$. When pseudo-machine m_i^d is in a remote state, i.e. $\alpha_i^d \in RS_i^d$, machine m_{i+1} is partially or completely blocked by machine $m_j, i < j \leq M+1$. The remote state α_i^d can be described by the local machine state α_{i+1}, the state α_{i+1}^d of pseudo-machine m_{i+1}^d in the downstream subsystem l_{i+1} and buffer b_{i+1}, i.e. $\alpha_i^d = \{b_{i+1}, \alpha_{i+1}, \alpha_{i+1}^d\}$. The rated input rate ω_i^{in} is determined by the local states of machine m_j. If machine m_i is up and partially blocked by machine m_j, then $\omega_i^{in} = s_j^{in} \prod_{r=i}^{j-1} \lambda_r$. If machine m_i is up and is completely blocked by machine m_j, then $\omega_i^{in} = 0$.

The transition rate matrices of each pseudo-machine can be determined by solving the decomposition equations proposed in literature [67] (shown in Appendix A). However, in this paper, we assume that each machine m_i may have different input and output rates. Therefore, a new model is developed with the major assumption of identical input and output rate of each machine released from Gershwin's original model. This released assumption will have two major impacts of the original model. The first one is the buffer level change equation which is determined as the difference of parts flowing in and out from the buffer, i.e.

$$\dot{b}_i = \mu_i^{out}(\alpha_i^u) - \omega_i^{in}(\alpha_i^d) \tag{7.1}$$

Secondly, the conservation of mass equations need to take into account the different input and output rates of each machine. To address this relationship, $\omega_{i-1}^{in} = \lambda_i \mu_i^{out}$ is imposed to represent the inflow rate and outflow rate of pseudo machines m_i^u and m_{i-1}^d with a factor λ_i. Therefore, the conservation of mass equations for machine m_i in state α_i can be expressed as

$$\sum_{\{b_i,\alpha_i,\alpha_i^d\}\in RS_{i-1}^d} \text{prob}\left[\{b_i,\ \alpha_i,\alpha_i^d\}\right]\omega_{i-1}^{in}\left(\{b_i,\ \alpha_i,\alpha_i^d\}\right) + \sum_{\{\alpha_i\}\in LS_{i-1}^d} \text{prob}[\{\alpha_i\}]\omega_{i-1}^{in}(\{\alpha_i\}) =$$
$$\lambda_i \left[\sum_{\{b_{i-1},\alpha_{i-1}^u,\alpha_i\}\in RS_i^u} \text{prob}[\{b_{i-1},\alpha_{i-1}^u,\alpha_i\}]\mu_i^{out}(\{b_{i-1},\alpha_{i-1}^u,\alpha_i\}) + \right.$$
$$\left. \sum_{\{\alpha_i\}\in LS_i^u} \text{prob}[\{\alpha_i\}]\mu_i^{out}(\{\alpha_i\})\right] \tag{7.2}$$

With these two modifications, the probability density functions $f[\{b_i,\alpha_i^u,\alpha_i^d\}]$ and boundary conditions $\text{prob}[\{0,\alpha_i^u,\alpha_i^d\}]$ and $\text{prob}[\{B_i,\alpha_i^u,\alpha_i^d\}]$ (i.e. the probability functions when buffer b_i is empty or full) can be determined based on the solution algorithm in Appendix B for each subsystem l_i. Therefore, all system performance measures can be evaluated. To demonstrate the evaluation process with Equations 1 and 2, we discuss below the evaluations of the frequency that a subsystem l_i entering a state $\{b_i,\alpha_i^u,\alpha_i^d\}$, the average system output rate and the average buffer levels.

For a subsystem $l_i, 1 \le i \le M$, the probability density function $f\left(b_i,\alpha_i^u,\alpha_i^d\right)$ that the subsystem l_i is in state $\{b_i,\alpha_i^u,\alpha_i^d\}$ can be expressed as the fraction of time that the system stays in the state, i.e.,

$$f\left(b_i, \alpha_i^u, \alpha_i^d\right)\delta b_i = \frac{N(b_i,\alpha_i^u,\alpha_i^d)}{T}\frac{\delta b_i}{|\mu_i^{out}(\alpha_i^u)-\omega_i^{in}(\alpha_i^d)|} + o(\delta t) \tag{7.3}$$

where $N(b_i, \alpha_i^u, \alpha_i^d)$ is the number of times that the subsystem l_i enters state $\{b_i, \alpha_i^u, \alpha_i^d\}$ during $(0, T]$. Therefore, the frequency that subsystem l_M enters state $\{b_i, \alpha_i^u, \alpha_i^d\}$ can be determined as

$$\frac{N(b_i,\alpha_i^u,\alpha_i^d)}{T} = |\mu_i^{out}(\alpha_i^u) - \omega_i^{in}(\alpha_i^d)| f\left(b_i, \alpha_i^u, \alpha_i^d\right) \tag{7.4}$$

System output rate OPR_{sys} is defined as the average number of products produced by the end-of-line machine m_{M+1} per unit time. Note that the pseudo-machine m_M^d in subsystem l_M consists of a single machine m_{M+1}. System output rate OPR_{sys} can be expressed as the average output rate of pseudo-machine m_M^d as

$$OPR_{sys} = \sum_{\alpha_M^u \in I_M^u} \sum_{\alpha_M^d \in I_M^d} \frac{\omega_M^{in}(\alpha_M^d)}{\lambda_{M+1}} \text{prob}\left[\left\{0, \alpha_M^u, \alpha_M^d\right\}\right] +$$

$$\sum_{\alpha_M^u \in I_M^u} \sum_{\alpha_M^d \in I_M^d} \frac{\omega_M^{in}(\alpha_M^d)}{\lambda_{M+1}} \text{prob}\left[\left\{B_M, \alpha_M^u, \alpha_M^d\right\}\right] +$$

$$\sum_{\alpha_M^u \in I_M^u} \sum_{\alpha_M^d \in I_M^d} \int_0^{B_{M-1}} \frac{\omega_M^{in}(\alpha_M^d)}{\lambda_{M+1}} f\left[\left\{b, \alpha_M^u, \alpha_M^d\right\}\right] \mathrm{d}b \tag{7.5}$$

where $\frac{\omega_M^{in}(\alpha_M^d)}{\lambda_{M+1}} = v_{M+1}^{out}\left(\alpha_M^d\right)$ denotes the output rate of the end-of-line virtual machine m_{M+1}, and $I_M^u = LS_M^u \cup RS_M^u$ and $I_M^d = LS_M^d \cup RS_M^d$ denotes all the local states and remote states of pseudo-machines m_M^u and m_M^d. In Equation 5, the first two terms denote the average output rate of the pseudo-machine m_{M-1}^d when buffer b_{M-1} is empty and full and the last term calculates the average output rate of the pseudo-machine m_{M-1}^d in general states. The expected buffer level $E[b_i]$ is determined as

$$E[b_i] = \sum_{\alpha_i^u \in I_i^u} \sum_{\alpha_i^d \in I_i^d} \int_0^{B_i} b f\left[\left\{b, \alpha_i^u, \alpha_i^d\right\}\right] \mathrm{d}b + B_i \text{prob}\left[\left\{B_i, \alpha_i^u, \alpha_i^d\right\}\right] \tag{7.6}$$

where $I_i^u = LS_i^u \cup RS_i^u$ and $I_i^d = LS_i^d \cup RS_i^d$.

7.4 Event-based Modeling

7.4.1 Mathematical System Dynamic Description

MDD is measured as the production loss at the end-of-line virtual machine m_{M+1}. Since virtual machine m_{M+1} can never be down or blocked, the production loss at the virtual machine can only be caused by partial starvation or complete starvation of all upstream machines i.e., DEMCs and DEMFs of all machines. DEMCs are caused by machine capacity asynchronization and DEMFs are caused by machine random failures. In this situation, identification and mitigation of bottlenecks caused by DEMCs (denoted as MC-BN) and DEMFs (denoted as MF-BN) are an effective way to reduce MDD. MC-BN and MF-BN are defined as the machines whose machine capacities and machine random failures impede MDD in the strongest manner.

MDD can be expressed as a function of each individual machine and buffer's parameters:

$$MDD = f(s_1^{in}, s_1^{out}, p_1, r_1, \dots, s_M^{in}, s_M^{out}, p_M, r_M, s_{M+1}^{in}, s_{M+1}^{out}, B_1, \dots, B_M) \qquad (7.7)$$

The definitions of MC-BN and MF-BN are defined based on Equation 7.

Definition 7.1 *A machine* $m_i, i = 1, \dots, M$ *is a MC-BN if*

$$\left|\frac{\partial MDD}{\partial s_i^{out}}\right| > \left|\frac{\partial MDD}{\partial s_j^{out}}\right|, \forall j \neq i \qquad (7.8)$$

Definition 7.2 *A machine* $m_i, i = 1, \dots, M$ *is a MF-BN if*

$$\frac{\partial MDD}{\partial MTTR_i} > \frac{\partial MDD}{\partial MTTR_j}, \forall j \neq i \qquad (7.9)$$

Remark 3.1 The absolute values of $\frac{\partial MDD}{\partial s_i^{out}}$ is used in Definition 1 because the value is negative, i.e. MDD decreases as s_i^{out} increases. □

Definitions 7.1 and 7.2 indicate that a machine is an MC-BN or MF-BN if a perturbation of its rated output rate or MTTR leads to the largest decrease or increase of the MDD. Since there is no closed-form expression of MDD when system becomes complex and it is very difficult to evaluate the derivatives of MDD, identification methods of MC-BN and MF-BN will be developed based on the measurable data or simulation results.

Assume that the virtual machine m_{M+1} is subject to a set of DEMCs $E^c = \{\vec{e}_1^c, \ldots, \vec{e}_{n^c}^c\}$, $n^c \geq 1$, and DEMFs $E^F = \{\vec{e}_1^F, \ldots, \vec{e}_{n^F}^F\}$, $n^F \geq 1$ during $(0, T]$, where $\vec{e}_i^c = (m_j, t_i, d_i)$ and $\vec{e}_i^F = (m_k, t_i, d_i)$ are the i^{th} DEMC and DEMF. Then MDD can be calculated as

$$MDD = \sum_{i=1}^{n^c} d_i [s_{M+1}^{out} - v_{M+1}^{out}(\vec{e}_i^c)] + \sum_{i=1}^{n^F} d_i s_{M+1}^{out} \tag{7.10}$$

where n^c and n^F are the number of DEMCs and DEMFs and $v_{M+1}^{out}(\vec{e}_i^c) = \frac{s_j^{out}}{\prod_{r=j+1}^{M+1} \lambda_r}$ is the processing rate of the virtual machine m_{M+1} if it is partially starved due to a DEMC $\vec{e}_i^c = (m_j, t_i, d_i)$. Based on Equation 10, indicators for MC-BN and MF-BN identification can be developed.

Proposition 7.1 *$\left|\frac{\partial MDD}{\partial s_i^{out}}\right| > \left|\frac{\partial MDD}{\partial s_j^{out}}\right|, i \neq j$, if $\frac{PT_i}{\prod_{r=i+1}^{M+1} \lambda_r} > \frac{PT_j}{\prod_{r=j+1}^{M+1} \lambda_r}$, where $PT_i = \sum_{l=1}^{n_i^c} d_{i,l}$ is the summation of the duration of DEMCs caused by machine m_i.*

Proof: $E_k^c = \left\{\vec{e}_{k,1}^c,\ \ldots,\ \vec{e}_{k,n_k^c}^c\right\}$, $n_k^c \geq 1$, denotes the DEMCs caused by machine m_k, where $\vec{e}_{k,l}^c = \left(m_k,\ t_{k,l}, d_{k,l}\right)$, $1 \leq l \leq n_k^c$, is the l^{th} DEMC in E_k^c. Based on Equation 7.10, market demand dissatisfaction MDD can be calculated with E_k^c as

$$MDD = \sum_{k=1}^{M} \sum_{l=1}^{n_k^c} d_{k,l} \left[s_{M+1}^{out} - \frac{s_k^{out}}{\prod_{r=k+1}^{M+1} \lambda_r}\right] + \sum_{l=1}^{n^F} d_l s_{M+1}^{out} \tag{7.11}$$

Suppose we reduce s_i^{out} (i.e. the rated output rate of machine m_i) in a small amount δs. Then the market demand dissatisfaction MDD becomes

$$MDD' = \sum_{1\leq k\leq M}^{k\neq i} \sum_{l=1}^{n_k^c} d_{k,l} \left[s_{M+1}^{out} - \frac{s_k^{out}}{\prod_{r=k+1}^{M+1} \lambda_r}\right] + \sum_{l=1}^{n_i^c} d_{i,l} [s_{M+1}^{out} - \frac{s_i^{out} - \delta s}{\prod_{r=i+1}^{M+1} \lambda_r}] + \sum_{l=1}^{n^F} d_l s_{M+1}^{out} \tag{7.12}$$

The derivative of customer satisfaction in Definition 1 can be calculated with Equations 7.11 and 7.12 as

$$\left|\frac{\partial MDD}{\partial s_i}\right| = \lim_{\delta s \to 0} \left|\frac{MDD - MDD'}{\delta s}\right| = \frac{\sum_{l=1}^{n_i^c} d_{i,l}}{\prod_{r=i+1}^{M+1} \lambda_r} \tag{7.13}$$

Therefore, partial derivative in Equation 7.8 is determined by $\frac{PT_i}{\prod_{r=i+1}^{M+1} \lambda_r}$, where $PT_i = \sum_{l=1}^{n_i^c} d_{i,l}$. □

Proposition 7.2 $\frac{\partial MDD}{\partial MTTR_i} > \frac{\partial MDD}{\partial MTTR_j}, i \neq j$, *if* $n_i^F > n_j^F$, *where* n_i^F *denotes the number of DEMFs caused by machine* m_i.

Proof: $E_k^F = \left\{\vec{e}_{k,1}^F,\ \ldots,\ \vec{e}_{k,n_k^F}^F\right\}$, $n_k^F \geq 1$, denotes the DEMFs caused by machine m_k, where $\vec{e}_{k,l}^F = \left(m_k,\ t_{k,l}, d_{k,l}\right)$, $1 \leq l \leq n_k^F$, is the l^{th} DEMF in E_k^F. Based on Equation 7.10, market demand dissatisfaction MDD can be calculated with E_k^F as

$$MDD = \sum_{l=1}^{n^c} d_l [s_{M+1}^{out} - v_{M+1}^{out}(\vec{e}_l^c)] + \sum_{k=1}^{M} \sum_{l=1}^{n_k^F} d_{k,l} s_{M+1}^{out} \tag{7.14}$$

Suppose we reduce the duration of each downtime event happened at machine m_i (i.e. MTTR of machine m_i) in a small amount δd. Then the duration of DEMFs at end-of-line virtual machine m_{M+1} caused by machine m_i is also reduced by δd. The market demand dissatisfaction MDD becomes

$$MDD' = \sum_{l=1}^{n^c} d_i [s_{M+1}^{out} - v_{M+1}^{out}(\vec{e}_l^c)] + \sum_{1 \le k \le M}^{k \ne i} \sum_{l=1}^{n_k^F} d_{k,l} s_{M+1}^{out} + \sum_{l=1}^{n_i^F} \left(d_{i,l} - \delta d\right) s_{M+1}^{out} \tag{7.15}$$

The derivative of customer satisfaction in Definition 2 can be calculated with Equations 7.14 and 7.15 as

$$\frac{\partial MDD}{\partial MTTR_i} = \lim_{\delta d \to 0} \frac{MDD - MDD'}{\delta d} = s_{M+1}^{out} n_i^F \tag{7.16}$$

Since s_{M+1}^{out} is a constant, n_i^F determines the derivative value in Equation 7.9. □

PT_i and n_i^F can be obtained by real-time information or simulation. With real-time information or simulation result, PT_i denotes the summation of the duration of DEMCs caused by machine m_i and n_i^F denotes the number of DEMFs caused by machine $m_i, 1 \le i \le M$. Alternatively, based on the stochastic analysis in Section 7.3, PT_i can be estimated as the probability that machine m_i causes DEMCs (in other words, partially starves virtual machine m_{M+1}) and n_i^F can be estimated as the frequency that machine m_i causes DEMFs (in other words, completely starves virtual machine m_{M+1}), which are denoted as $\widetilde{PT}_i$ and $\tilde{n}_i^F$ respectively.

Note that in the last subsystem l_M (shown in Figure 7.2), the pseudo-machine m_M^d consists of single end-of-line virtual machine m_{M+1}. To find $\widetilde{PT}_i$ and $\tilde{n}_i^F$, it is equivalent to estimate the probability that machine m_i partially starves pseudo-machine m_M^d (i.e. causes DEMCs) and the frequency that machine m_i completely starves pseudo-machine m_M^d (i.e. causes DEMF). For the ease of expression, G_i^p and G_i^c are used to denote the states that pseudo-machine machine m_M^d is partially and completely starved by a machine $m_i, 1 \leq i \leq M$.

When pseudo-machine m_M^d is partially starved by machine m_i, the subsystem l_M is in a state $\{0, \alpha_M^u, \alpha_M^d\}$, where $\alpha_M^d \in G_i^p$. $\widetilde{PT}_i$ can be estimated as the probability that the subsystem is in such states, i.e.

$$\widetilde{PT}_i = \sum_{\alpha_M^d \in G_i^p} \sum_{\alpha_M^u \in I_M^u} P[\{0, \alpha_M^u, \alpha_M^d\}] \tag{7.17}$$

where $I_M^u = LS_M^u \cup RS_M^u$. When the last virtual machine m_{M+1} is completely starved by machine m_i, the subsystem l_M is in a state $\{0, \alpha_M^u, \alpha_M^d\}$, where $\alpha_M^d \in G_i^c$. The frequency that virtual machine m_{M+1} is completely starved by machine m_i (i.e. $\tilde{n}_i^F$) can be calculated based on Equation 7.4 as

$$\tilde{n}_i^F = \sum_{\alpha_M^u \in I_M^u} \frac{N(0, \alpha_M^u, \alpha_M^d)}{T} = \sum_{\alpha_M^u \in I_M^u} \left|\mu_M^{out}(\alpha_M^u) - \omega_M^{in}(\alpha_M^d)\right| f(0, \alpha_M^u, \alpha_M^d), \alpha_M^d \in G_i^c \tag{7.18}$$

7.4.2 Independent MC-BNs and MF-BNs Identification

Propositions 7.1 and 7.2 provide indicators for MC-BN and MF-BN, i.e. the machine m_i with the highest value of $\frac{PT_i}{\prod_{r=i+1}^{M+1} \lambda_r}$ (or $\frac{\widetilde{PT}_i}{\prod_{r=i+1}^{M+1} \lambda_r}$) is MC-BN and the machine with the highest value of n_i^F (or $\tilde{n}_i^F$) is MF-BN. However, the machines with the second largest indicators are not necessary the next MC-BN and MF-BN when current MC-BN and MF-BN are mitigated.

Therefore, to find the second or third independent MC-BNs and MF-BNs, sensitivity analysis must be repeated when the previous MC-BNs and MF-BNs are mitigated. The following iterative procedure is used to identify independent MF-BNs with severity ranking. MC-BNs can be similarly identified. For the ease of discussion, we use MF-BN(*j*) to denote the $j^{th}, 1 \leq j \leq M,$ significant independent MF-BN.

1) Let $j = 1$ and identify MF-BN(*j*) using MF-BN indicator.
2) Decrease the MTTR of MF-BN(*j*) by 10% and identify the MF-BN in the updated system. The new MF-BN is denoted as MF-BN*.
3) If MF-BN* does not indicate any MF-BN(*k*), $1 \leq k \leq j$, MF-BN(*j*+1) = MF-BN* and update $j = j + 1$. Otherwise, return to Step 2.
4) If $j > M$ or $MDD = 0$, terminate the procedure; otherwise, return to Step 2.

Note that the procedure uses simulation or stochastic analysis to find MF-BN and MC-BN indicators. To identify independent MC-BNs and MF-BNs, the simulation or stochastic analysis must be repeated. Although the method is very useful for independent MC-BNs and MF-BNs identification through off-line calculation, it is a computational expensive procedure. It is difficult to apply the method to identify MC-BNs and MF-BNs in real-time.

To identify MF-BNs in real-time, a data-driven method is developed through utilizing online data. Numerical studies indicate that most of the major MF-BNs (e.g. the first two or three MF-BNs) identified based on the iterative procedure are turning points, where a turning point refers to a machine that is more likely to cause its upstream machines to be completely starved and downstream machines to be completely blocked and is less likely to be completely starved or blocked compared with its adjacent machines [10]. Therefore turning points can be defined as local MF-BNs and can be found based on a data-driven method developed in literature [10]. An MF-BN can be identified as a machine m_i if

System 1

	m_1	b_1	m_2	b_2	m_3	b_3	m_4	b_4	m_5	b_5	m_6
Rated input rate	16		8.1		5.2		3.2		1.1		1
Rated output rate	8.5		4.6		3.5		0.8		1		1
MTTR	15		18		10		8		13		N/A
MTBF	80		75		70		75		70		N/A
Buffer Capacity		30		25		20		10		10	

Figure 7.3 Simulation verification example 1

System 2

	m_1	b_1	m_2	b_2	m_3	b_3	m_4	b_4	m_5	b_5	m_6	b_6	m_7
Rated input rate	10		7.9		5.1		3.2		2.3		1.2		1
Rated output rate	8		5		3		2.8		1		1		1
MTTR	10		20		10		15		10		10		N/A
MTBF	60		65		65		70		70		75		N/A
Buffer Capacity		16		21		16		15		15		10	

Figure 7.4 Simulation verification example 2

System 3

	m_1	b_1	m_2	b_2	m_3	b_3	m_4	b_4	m_5	b_5	m_6	b_6	m_7	b_7	m_8
Rated input rate	16		7.9		5.1		3.2		2.3		1.2		1.1		1
Rated output rate	8		4.8		3.2		2.2		1		1		1		1
MTTR	8		13		9		10		10		9		6		N/A
MTBF	50		55		55		60		60		65		65		N/A
Buffer Capacity		16		21		16		15		15		10		10	

Figure 7.5 Simulation verification example 3

$$TB_i^c - TS_i^c > 0, \qquad i \neq 1, M+1$$

$$TB_i^c - TS_i^c < 0, \qquad i \neq 1, M+1$$

$$TB_i^c + TS_i^c < TB_{i-1}^c + TS_{i-1}^c, \qquad i \neq 1, M+1$$

$$TB_i^c + TS_i^c < TB_{i+1}^c + TS_{i+1}^c, \qquad i \neq 1, M+1$$

$$\text{if } i = 1: TB_1^c - TS_1^c > 0, TB_2^c - TS_2^c < 0, TB_1^c + TS_1^c < TB_2^c + TS_2^c$$

$$\text{if } i = M+1: TB_M^c - TS_M^c > 0, TB_{M+1}^c - TS_{M+1}^c < 0, TB_{M+1}^c + TS_{M+1}^c < TB_M^c + TS_M^c$$

where TB_i^c and TS_i^c are the accumulated complete blockage and starvation time of machine m_i. The severity of the local MF-BNs can be determined based on MF-BN indicator values, i.e. the natural value ranking of n_i^F.

Numerical experiments are performed to verify the major MF-BNs and their severity rankings based on the data-driven method and MF-BN indicator ranking. Extensive simulation studies have been performed, and three examples are shown in Figures 7.3-7.5. MF-BNs identified by data-driven method with indicator severity rankings are shown in the first row of Tables 7.1-7.3. It can be observed that in all three systems, the local MF-BNs with indicator severity rankings match the results based on the simulation iterative procedure.

Table 7.1 Results for MC-BNs and OC-BNs identification in system 1

Data-driven method	MF-BN	m_2				
Iterative simulation	MF-BN	m_2	m_3	m_4	m_1	m_5

Table 7.2 Results for MC-BNs and OC-BNs identification in system 2

Data-driven method	MF-BN	m_5	m_2				
Iterative simulation	MF-BN	m_5	m_2	m_4	m_6	m_3	m_1

Table 7.3 Results for MC-BNs and OC-BNs identification in system 3

Data-driven method	MF-BN	m_5	m_7					
Iterative simulation	MF-BN	m_5	m_7	m_2	m_4	m_6	m_3	m_1

7.4.3 Supervisory Control Scheme

To reduce MDD in a market demand driven system, a supervisory control algorithm is introduced. The control is a feedback process which involves two steps:

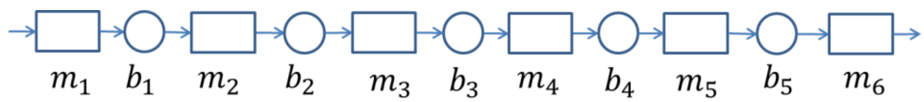
1) Collecting information about critical performance measures over time;

2) Taking appropriate control actions based on the collected information.

It means periodically monitoring MDD, comparing it to a desired level which obviously can be presented as $MDD^* = 0$, identifying the main causes of $MDD > 0$, and taking corrective actions to eliminate the causes. It is similar to the plan-do-check-act cycle for continuous improvement.

Since MC-BN and MF-BN are the main reasons causing system failing to satisfy market demand (i.e. $MDD > 0$), the control action focuses on identifying and mitigating the MC-BNs and MF-BNs. Specifically, when MC-BNs and MF-BNs are identified based on the aforementioned methods, limited resources and budget can be prioritized and focused on the selected MC-BNs and MF-BNs to most effectively improve the system performance. For example, extra workers can be assigned to MC-BNs to increase their rated output rates and maintenance priority can be scheduled for MF-BNs to decrease their MTTRs.

7.5 Case Study



	m_1	b_1	m_2	b_2	m_3	b_3	m_4	b_4	m_5	b_5	m_6
Rated input rate	16		8.1		5.2		3.2		1.1		1
Rated output rate	8.0		4.9		3.1		1		1		1
MTTR	16		15		15		15		10		N/A
MTBF	60		65		60		60		65		N/A
Buffer Capacity		16		11		6		5		5	

Figure 7.6 A battery production line and its market demand

Extensive numerical experiments are performed to verify the methods to identify MC-BNs and MF-BNs and the effectiveness of control policies focusing on MC-BNs and MF-BNs. As an illustration, a market demand driven system consisting of 6 machines is selected, which is shown in Figure 7.6.

Table 7.4 MC-BNs and MF-BNs identified based on iterative procedure and indicators

Iterative simulation	MF-BN	m_3	m_1	m_4	m_2	m_5
	MC-BN	m_4	m_3	m_2	m_1	
	Computational time	15 mins				
Indicators with simulation	MF-BN	m_3	m_1	m_4	m_2	m_5
	MC-BN	m_4	m_3	m_2	m_1	
	Computational time	8 mins				
Indicators with stochastic	MF-BN	m_3	m_1	m_4	m_2	m_5
	MC-BN	m_4	m_3	m_2	m_1	
	Computational time	3 mins				

Table 7.4 shows the MC-BNs and MF-BNs results based on three methods. Sensitivities $\frac{\Delta MDD}{\Delta s_i^{out}}$ and $\frac{\Delta MDD}{\Delta MTTR_i}$ are estimated numerically with steps $\partial s_i^{out} = 0.05 s_i^{out}$ and $\partial MTTR_i = 0.05 MTTR_i$ and the iterative procedure is adopted to identify all the independent bottlenecks. These results are shown in the "Iterative simulation" row. Bottlenecks based on the indicators $\frac{PT_i}{\prod_{r=i+1}^{M+1} \lambda_r}$ and n_i^F through simulation evaluation are shown in the "Indicators with simulation"

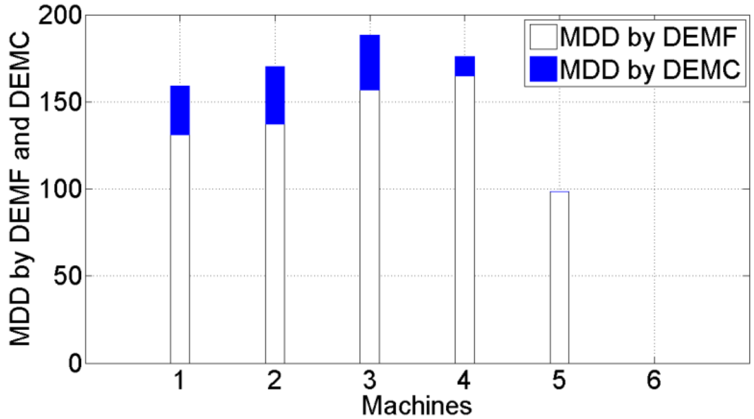


Figure 7.7 $\boldsymbol{MDD^{c}}$ and $\boldsymbol{MDD^{F}}$ caused by each machine

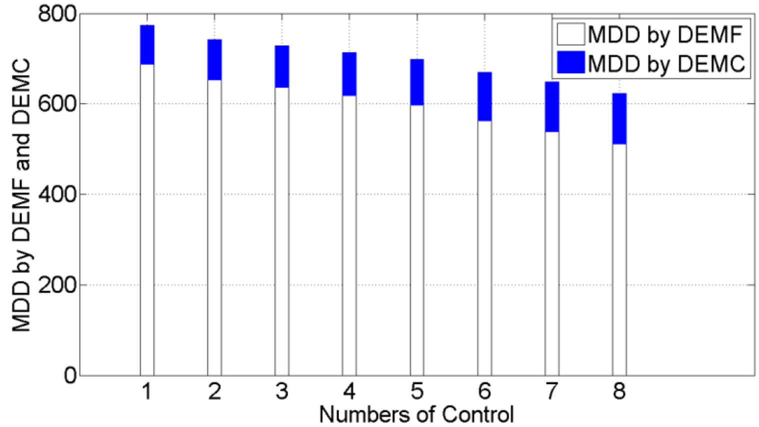


Figure 7.8 $\boldsymbol{MDD}$ changes with supervisory control focusing on MF-BNs

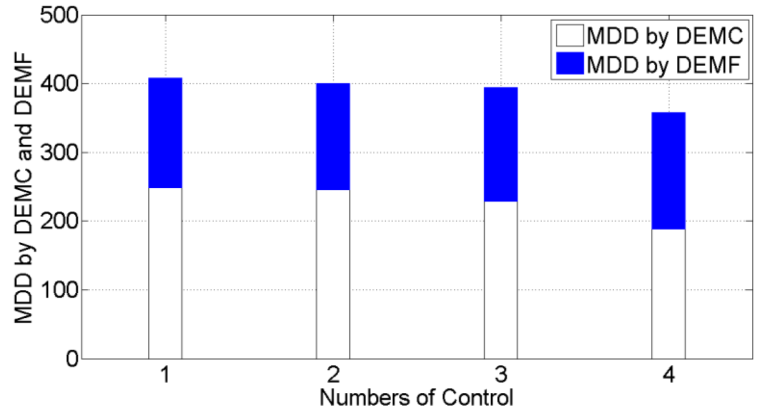


Figure 7.9 MDD changes with supervisory control focusing on MC-BNs

row. Bottlenecks based on the indicators $\frac{\widetilde{PT}_i}{\prod_{r=i+1}^{M+1}\lambda_r}$ and $\tilde{n}_i^F$ through stochastic evaluation is shown in the "Indicators with stochastic" row.

In addition, the effects of improving MC-BNs and MF-BNs are analyzed. For convenience, MDD^c and MDD^F are used to denote the MDD caused by DEMCs and DEMFs. Figure 7.7 shows the portions of MDD^c and MDD^F caused by each machine. It can be observed that DEMFs cause the most of MDD. Therefore, the supervisory control is focused on MF-BNs first. MF-BNs are identified and improved with a control interval of 24 hours and MDD^c and MDD^F after each improvement are shown in Figure 7.8. According to the result, the control gradually decreases the MDD^F portion and the overall MDD. This is due to the fact that the improvements of MF-BNs reduce the overall downtime, which naturally decreases MDD^F. When the continuous improvements on MF-BNs cannot further decrease the overall MDD, the supervisory control switches to MC-BNs mitigation. Similarly, the control gradually reduces the MDD^c portion and the overall MDD, and changes the ratio of the MDD^F portion and the MDD^c portion as shown in Figure 7.9.

Table 7.5 The three control policies

		m_1	m_2	m_3	m_4	m_5
Policy 1	Output rate (increase by %)	0	5%	15%	30%	0
	MTTR (decrease by min)	2	0	3	1	0
Policy 2	Output rate (increase by %)	0	5%	15%	30%	0
	MTTR (decrease by min)	0	1	3	2	0
Policy 3	Output rate (increase by %)	0	0	5%	15%	30%
	MTTR (decrease by min)	3	2	1	0	0

Table 7.6 MDD changes caused by the three policies

	Policy 1	Policy 2	Policy 3
Throughput increase	36	19	8
MDD decrease	36	19	8

Lastly, the effectiveness and the benefits of using the proposed supervisory control methodology are demonstrated through comparing with two other control policies in a simulation time of 24 hours. The three policies are listed in Table 7.5. In policy 1, the major MF-BNs are identified as m_3, m_1 and m_4 and MC-BNs are identified as m_4, m_3 and m_2 based on the iterative procedure. The output rates of the three major MC-BNs are increased by 30%, 15% and 5% and the MTTRs of the major three MF-BNs are decreased by 3, 2 and 1 minute. In policy 2, the top three machines are selected based on the highest sensitivity values without using iterative procedure in Section 7.4.2. The output rates of the three machines with the highest sensitivity values $\frac{\partial MDD}{\partial s_i^{out}}$ are increased by 30%, 15% and 5% and the MTTRs of the three machines with the highest sensitivity values $\frac{\partial MDD}{\partial MTTR_i}$ are decreased by 3, 2 and 1 minute. In policy 3, the output rates of the three machines with the lowest output rate are increased by 30%, 15% and 5% and the MTTRs of the three machines with the highest MTTR are decreased by 3, 2 and 1 minute. Table 7.6 shows the MDD decrease and throughput increase when the three control policies are applied. The result indicates that the proposed supervisory control brings the most MDD reduction (throughput increase). This validates that the most effective way to improve system throughput and reduce MDD is through identification and mitigation of MC-BNs and MF-BNs.

7.6 Conclusion

This chapter addresses multistage production system performance using a Markovian continuous-flow model with a decomposition method. The model captures the key features of machines with different input and output rates as well as large variance in machine cycle time, which were not normally described in the existing analytical models. A market demand driven system is developed by modeling market demand as the end-of-line virtual machine. In such a system, any market demand dissatisfaction, i.e. MDD, can be measured as the production loss at the end-of-line virtual machine. An event-based methodology is developed to categorize the MDD caused by machine capacity bottlenecks (MC-BNs) and machine failure bottlenecks (MF-BNs). Indicators for MC-BN and MF-BN identification are proposed. It is shown that the MC-

BN and MF-BN identification indicators are related to DEMCs and DEMFs. Based on the identification indicators, an iterative procedure and a data-driven method are developed to identify all the independent MC-BNs and MF-BNs. A supervisory control algorithm to identify and mitigate MC-BNs and MF-BNs is introduced. The case study confirms that the supervisory control focusing on MC-BNs and MF-BNs can most efficiently increase system throughput and reduce MDD.

In the future, optimal control algorithms will be developed to minimize MDD through improving MC-BNs and MF-BNs with constrains (e.g. limited resources and budget).

7.7 Appendix

7.7.1 Transaction Matrices of Pseudo-machines

The states of a pseudo-machine $m_i^u, 1 \leq i \leq M$, can be categorized into two: local states LS_i^u and remote states RS_i^u. The size of local states LS_i^u and remote states RS_i^u are assumed to be LN_i^u and RN_i^u. The transition rate matrix of pseudo-machine is a $(LN_i^u + RN_i^u) \times (LN_i^u + RN_i^u)$ square matrix. Depending on the transition links, the transition rate matrix τ_i^u consists of four sub-matrices as:

$$\tau_i^u = \begin{bmatrix} \tau_i^u(\alpha_i^{u*}, \alpha_i^{u*}) & \tau_i^u(\alpha_i^{u*}, \alpha_i^u) \\ \tau_i^u(\alpha_i^u, \alpha_i^{u*}) & \tau_i^u(\alpha_i^u, \alpha_i^u) \end{bmatrix}$$

where $\tau_i^u(k, j)$ denotes the transition rate sub-matrix from k to j and α_i^{u*} and α_i^u denote the local states and remote states of pseudo-machine m_i^u.

The sub-matrix $\tau_i^u(\alpha_i^{u*}, \alpha_i^{u*})$ is a $LN_i^u \times LN_i^u$ square matrix. It can be determined based on the transition rate of machine m_i, i.e.

$$\tau_i^u(\alpha_i^{u*}, \alpha_i^{u*}) = \tau_i(\alpha_i, \alpha_i') \tag{7.19}$$

where $\tau_i(\alpha_i, \alpha_i')$ denotes the transition rate of machine m_i from state α_i to α_i'.

The sub-matrix $\tau_i^u(\alpha_i^u, \alpha_i^{u*})$ is a $RN_i^u \times LN_i^u$ matrix. It links remote states α_i^u with local states α_i^{u*}. The transition can be triggered for two reasons: the machine m_j causing DEMFs is repaired or machine m_i is down. Therefore, the transition rate matrix consists of two contributions $\tau_i^{u,1}(\alpha_i^u, \alpha_i^{u*})$ and $\tau_i^{u,2}(\alpha_i^u, \alpha_i^{u*})$, which are found as

$$\tau_i^{u,1}(\alpha_i^u, \alpha_i^{u*}) = r_j,\ \ \alpha_i^u = \{0, \alpha_{i-1}^u, \alpha_i\} \in RS_i^u,\ \ \alpha_i^{u*} = \{\alpha_i\} \in LS_i^u \tag{7.20}$$

$$\tau_i^{u,2}(\alpha_i^u, \alpha_i^{u*}) = p_i,\ \ \alpha_i^u = \{0, \alpha_{i-1}^u, \alpha_i\} \in RS_i^u,\ \ \alpha_i^{u*} = \{\alpha_i\} \in LS_i^u \tag{7.21}$$

The transition rate matrix $\tau_i^u(\alpha_i^u, \alpha_i^{u*})$ can be calculated as

$$\tau_i^u(\alpha_i^u, \alpha_i^{u*}) = \tau_i^{u,1}(\alpha_i^u, \alpha_i^{u*}) + \tau_i^{u,2}(\alpha_i^u, \alpha_i^{u*}) \tag{7.22}$$

The sub-matrix $\tau_i^u(\alpha_i^u, \alpha_i^u)$ is a $RN_i^u \times RN_i^u$ square matrix. It links the remote states α_i^u with remote states α_i^u. The transition occurs when a machine $m_j, j \neq i$, transfers from state α_j to state α_j' such that the pseudo-machine m_i^u transfers from a remote state $\alpha_i^u = \{0, \alpha_{i-1}^u, \alpha_i\}$ to another remotes state $\alpha_i^{u\prime} = \{0, \alpha_{i-1}^{u\prime}, \alpha_i\}$. The transition rate matrix can be expressed as

$$\tau_i^u(\alpha_i^u, \alpha_i^{u'}) = \tau\big(\alpha_j, \alpha_j'\big),\ \ \alpha_i^u = \{0, \alpha_{i-1}^u, \alpha_i\} \in RS_i^u,\ \ \alpha_i^{u\prime} = \{0, \alpha_{i-1}^{u\prime}, \alpha_i\} \in RS_i^u \tag{7.23}$$

The sub-matrix $\tau_i^u(\alpha_i^{u*}, \alpha_i^u)$ is a $LN_i^u \times RN_i^u$matrix. It links local states α_i^{u*} with remote states α_i^u. According to the analysis in [67], the transition rate matrix can be calculated as

$$\tau_i^u(\alpha_i^{u*}, \alpha_i^u) = \frac{\text{prob}[\{\alpha_i^{u*}\}] \sum_{q \in I_i^u} \tau_i^u(\alpha_i^{u*}, q) - \sum_{q \in RS_i^u} \tau_i^u(\alpha_i^{u*}, q) \text{prob}[\{q\}]}{\text{prob}[\{\alpha_i^{u*}\}]} \tag{7.24}$$

where $I_i^u = RS_i^u \cup LS_i^u$.

Similarly, the transition rate matrix of a pseudo-machine $m_i^d, 1 \le i \le M-1$, can be determined. LN_i^d and RN_i^d denote the size of local states LS_i^d and remote states RS_i^d of pseudo-machine m_i^d. The transition rate matrix of pseudo-machine is a $(LN_i^d + RN_i^d) \times \left(LN_i^d + RN_i^d\right)$ square matrix, which is expressed as

$$\tau_i^d = \begin{bmatrix} \tau_i^d(\alpha_i^{d*}, \alpha_i^{d*}) & \tau_i^d(\alpha_i^{d*}, \alpha_i^d) \\ \tau_i^d(\alpha_i^d, \alpha_i^{d*}) & \tau_i^d(\alpha_i^d, \alpha_i^d) \end{bmatrix}$$

$\tau_i^d(\alpha_i^{d*}, \alpha_i^{d*})$ is a square matrix with a size of $LN_i^d \times LN_i^d$ and can be determined as

$$\tau_i^d(\alpha_i^{d*}, \alpha_i^{d*}) = \tau_{i+1}(\alpha_{i+1}, \alpha'_{i+1}) \tag{7.25}$$

$\tau_i^d(\alpha_i^d, \alpha_i^{d*})$ is a matrix with a size of $RN_i^d \times LN_i^d$ and can be determined as

$$\tau_i^{d,1}\left(\alpha_i^d, \alpha_i^{d*}\right) = r_j,\ \alpha_i^d = \left\{B_i,\ \alpha_{i+1}, \alpha_{i+1}^d\right\} \in RS_i^d,\ \alpha_i^{d*} = \{\alpha_{i+1}\} \in LS_i^d \tag{7.26}$$

$$\tau_i^{d,2}\left(\alpha_i^d, \alpha_i^{d*}\right) = p_i,\ \alpha_i^d = \left\{B_i, \alpha_{i+1}, \alpha_{i+1}^d\right\} \in RS_i^d,\ \alpha_i^{d*} = \{\alpha'_{i+1}\} \in LS_i^d \tag{7.27}$$

$$\tau_i^d\left(\alpha_i^d, \alpha_i^{d*}\right) = \tau_i^{d,1}\left(\alpha_i^d, \alpha_i^{d*}\right) + \tau_i^{d,2}\left(\alpha_i^d, \alpha_i^{d*}\right) \tag{7.28}$$

$\tau_i^d(\alpha_i^d, \alpha_i^d)$ is a $RN_i^d \times RN_i^d$ square matrix and is determined as

$$\tau_i^d(\alpha_i^d, \alpha_i^{d'}) = \tau\left(\alpha_j, \alpha'_j\right),\ \alpha_i^d = \left\{B_i,\ \alpha_{i+1}, \alpha_{i+1}^d\right\} \in RS_i^d,\ \alpha_i^{d\prime} = \left\{B_i,\ \alpha'_{i+1}, \alpha_{i+1}^{d\prime}\right\} \in RS_i^d \tag{7.29}$$

$\tau_i^d(\alpha_i^{d*}, \alpha_i^d)$ is a $LN_i^d \times RN_i^d$ square matrix and is determined as

$$\tau_i^d(\alpha_i^{d*}, \alpha_i^d) = \frac{\mathrm{prob}[\{\alpha_i^{d*}\}]\sum_{q\in I_i^d}\tau_i^d(\alpha_i^{d*}, q) - \sum_{q\in RS_i^d}\tau_i^d(\alpha_i^{d*}, q)\mathrm{prob}[\{q\}]}{\mathrm{prob}[\{\alpha_i^{d*}\}]} \tag{7.30}$$

where $I_i^d = RS_i^d \cup LS_i^d$.

7.7.2 Solution Algorithm

According to the analysis in [67], the computational algorithm for decomposition equations is shown below.

Step 1:

1) Determine the state space of each pseudo-machine $m_i^u, 1 \leq i \leq M$, described in Section 7.3.
2) Determine the sub-matrices of transition rate $\tau_i^u(\alpha_i^{u*}, \alpha_i^{u*})$, $\tau_i^u(\alpha_i^u, \alpha_i^{u*})$ and $\tau_i^u(\alpha_i^u, \alpha_i^u)$ based on Equations 7.19-7.23.
3) Estimate the sub-matrices of transition rate $\tau_i^u(\alpha_i^{u*}, \alpha_i^u)$ with a low starting value.
4) Determine the state space of each pseudo-machine $m_i^d, 1 \leq i \leq M$.
5) Determine the sub-matrices of transition rate $\tau_i^d(\alpha_i^{d*}, \alpha_i^{d*})$, $\tau_i^d\left(\alpha_i^d, \alpha_i^{d*}\right)$ and $\tau_i^d(\alpha_i^d, \alpha_i^d)$ based on Equations 7.25-7.29.
6) Estimate the sub-matrices of transition rate $\tau_i^d(\alpha_i^{d*}, \alpha_i^d)$ with a low starting value.
7) Analyze each subsystem l_i using the method proposed by Tan and Gershwin [68] and find the initial values for the performance measures and state probabilities.

Step 2:

1) Calculate the probabilities that pseudo-machine m_i^u is in a remote state $\alpha_i^u \in RS_i^u$ using

   $\text{prob}[\{0, \alpha_{i-1}^u, \alpha_i\}] = \text{prob}\left[\{0, \alpha_{i-1}^u, \alpha_{i-1}^d = \alpha_i\}\right] + \sum_{\alpha_{i-1}^d \in RS_i^d} \text{prob}\left[\{0, \alpha_{i-1}^u, \alpha_{i-1}^d\}\right]$

   And calculate the probabilities that pseudo-machine m_i^d is in a remote state $\alpha_i^d \in RS_i^d$ using

   $\text{prob}\left[\{B_i,\ \alpha_{i+1}, \alpha_{i+1}^d\}\right] = \text{prob}\left[\{B_i, \alpha_{i+1}^u = \alpha_{i+1}, \alpha_{i+1}^d\}\right] + \sum_{\alpha_{i+1}^u \in RS_{i+1}^u} \text{prob}\left[\{0, \alpha_{i+1}^u, \alpha_{i+1}^d\}\right]$
2) Calculate the probabilities that pseudo-machine m_i^u is in a local state $\alpha_i^u \in LS_i^u$ and the probabilities that pseudo-machine m_i^d is in a local state $\alpha_i^d \in LS_i^d$ using Equation 7.2.
3) Update the sub-matrix of transition rate $\tau_i^u(\alpha_i^{u*}, \alpha_i^u)$ with Equation 7.24 and the sub-matrix of transition rate $\tau_i^d(\alpha_i^{d*}, \alpha_i^d)$ with Equation 7.30.

4) Analyze each subsystem l_i using the method proposed by Tan and Gershwin [68] and update the values of performance measures and state probabilities.

Step 3:

1) Given a small tolerance value δ, check the following equation

$$\left|\tilde{\lambda}_{i+1} - \lambda_{i+1}\right| \leq \delta, 1 \leq i \leq M - 1$$

where $\tilde{\lambda}_{i+1} = \frac{\mu_i^{out}}{\mu_{i+1}^{out}}$. If the equation holds, finish the iteration. Otherwise, go back to Step 2.

Chapter 8

AN IMPROVED SUPERVISORY CONTROL SCHEME

8.1 Introduction

In Chapter 7, we have introduced a supervisory control scheme to improve system performance and market demand satisfaction. In the scheme, both real-time EBM based and Stochastic EBM based bottleneck identification and mitigation rules can be used. Both rules have their advantages and limitations. For example, real-time EBM considers realized production trajectory, but may not be accurate in predicting future production status in face of uncertainties. The real-time EBM based rule is efficient only within a relatively small time window. Stochastic EBM, on the other hand, is developed to estimate the long-term steady state system performance. Stochastic EBM based rule may not be as efficient as real-time EBM based rule in improving system performance in small time windows. Nevertheless, it has higher control efficiency when time windows become large. Therefore, a question arises naturally as when real-time EBM based and stochastic EBM based rules will be used for system control. Additionally, it remains a problem as how frequency to update the EBM based rules. If the time interval control is too short, it may result in chasing noise, but too long of time interval control will miss the opportunity.

Motivated by the questions, in this chapter, we use Markov decision model to find the optimal control policies for bottleneck identification and mitigation rules' selection and their updating. Based on the optimal policies, an improved supervisory control scheme is further developed.

8.2 Problem Formulation

8.2.1 System Descriptions

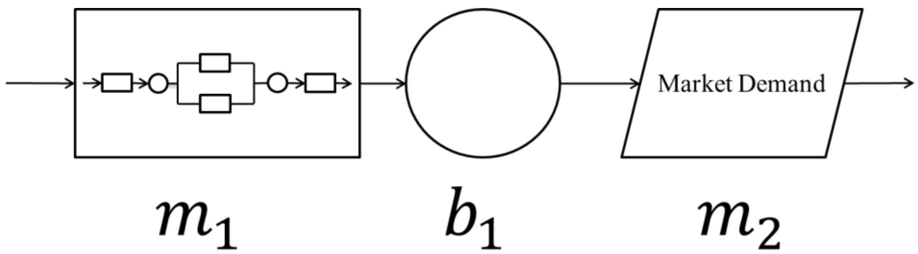


Figure 8.1 An advanced manufacturing system

The system that we consider consists of a production line, a finished-goods buffer and a market demand virtual machine, which is shown in Figure 8.1. The existing literature [15-17] has enabled us to model a manufacturing system as a single virtual machine with aggregation method or decomposition method. Therefore, in this chapter, we use virtual machine m_1 to model the manufacturing system. The system has following characteristics:

1) Real-time EBM based and stochastic EBM based rules are used to identify bottlenecks. For ease of expression, we use f_0 to represent real-time EBM based rule and f_1 to represent stochastic EBM based rule. Both rules are updated simultaneously when a certain criteria is satisfied. a is denoted as the policy age, which is defined as:

$$a(t) = \begin{cases} g(t), & T_n < t < T_{n+1} \\ 0, & t = T_{n+1} \end{cases} \tag{8.1}$$

where $g(t)$ is a non-decreasing function of time t and T_n and T_{n+1} are the nth and $n+1$th time when the EBM policies are updated. For ease of discussion, we assume that $g(t) = a(t-1)+1$. This assumption can be relaxed without any problem.

2) Virtual machine m_1 has a constant rated speed, denoted by s_1 (units/time unit). Virtual machine m_1 has one up state ($\alpha_1 = 1$) and one down state ($\alpha_1 = 0$). It obeys Bernoulli reliability model [Liang Zhang, Meerkov]. At time t, virtual machine m_1 has a probability of $p_1(f_i, a)$ being up and has a probability of $1 - p_1(f_i, a)$ being down, where $i = 1, 2$ and $a \geq 0$. $p_1(f_i, a)$ is a function of rules $\{f_0, f_1\}$ and policy age a with the following properties: (a) $p_1(f_i, a)$ is an decreasing function of policy age a, (b) there exists a time t_0, such that $\begin{cases} p_1(f_0, a) < p_1(f_1, a), & a \leq t_0 \\ p_1(f_0, a) > p_1(f_1, a), & a > t_0 \end{cases}$. All the machine state changes occur at the end of a time unit.
3) Virtual machine m_2 has a constant rated speed, denoted as d (units/time unit). It can never break down.
4) Finished-goods buffer b_1 has a finite buffer capacity, which is denoted as B_1. For ease of expression, we still use $b_1, 1 \leq b_1 \leq B_1$, to denote the buffer level of the finished-goods buffer b_1.
5) Backordering is not allowed. Any product demand that is not satisfied in a time unit will be lost permanently. It cannot be made up at latter production.
6) System state cam be expressed as $\{a, b_1\} \in S$. The state space is the set $S = \{0,1,2, \dots\} \times \{0, 1, \dots, B_1\}$.

8.2.2 Decision Variables

There are two decision variables: variable f indicating the bottleneck identification rule that will be selected, i.e. real-time EBM based rule f_0 or stochastic EBM based rule f_1, and variable β determining whether the rules f_0 and f_1 are updated. Specifically, $\beta = 0$ denotes that the rules are not updated and $\beta = 1$ denotes that the rules are updated. The aim is to find an optimal control policy $\{\pi_f, \pi_\beta\}$, such that the expected system cost V_{t+1} is minimized at each state $\{a, b_1\} \in S$.

8.2.3 Cost Function

The expected system cost function consists of the average operation cost, fixed cost to update f_0 and f_1, and lost sale penalty. The following notations are used:

c_0, average operational cost per time unit,

c_u, fixed cost for updating f_0 and f_1,

c^-, unit lost sale penalty per time unit.

Let λ $(0 < \lambda < 1)$ denote a discount factor. When system is in state $\{a, b_1\}, a \geq 0, 0 \leq b_1 \leq B_1$, and control actions $\{f, \beta\}$ are used, the expected discounted system cost $V_{t+1}^{\{f,\beta\}}(a, b_1)$ through an infinite horizon can be represented with the following recursive equations:

$$V_{t+1}^{\{f,\beta\}}(a, b_1) = c_0 + \beta c_u + c^-[\big(1 - p_1(f_i, a)\big)\max(d - b_1, 0)] + \lambda[p_1(f_i, a)V_t(a + 1, \min(b_1 + s_1 - d, B_1)) + \big(1 - p_1(f_i, a)\big)V_t(a + 1, \min(b_1 - d, 0))] \quad (8.2)$$

where $V_t(a, b_1)$ is the minimum expected discounted system cost when system is in state $\{a, b_1\}$. This minimum cost and the optimal policy can be found by solving the following recursive optimality equations:

$$V_{t+1}^{\beta=0}(a, b_1) = V_{t+1}^{\{\pi_f,\beta=0\}}(a, b_1) = \min_{i=0,1}\big\{c_0 + c^-[\big(1 - p_1(f_i, a)\big)\max(d - b_1, 0)] + \lambda[p_1(f_i, a)V_t(a + 1, \min(b_1 + s_1 - d, B_1)) + \big(1 - p_1(f_i, a)\big)V_t(a + 1, \min(b_1 - d, 0))]\big\},$$

(8.3)

$$V_{t+1}^{\beta=1}(a,b_1) = V_{t+1}^{\{\pi_f,\beta=1\}}(a,b_1) = \min_{i=0,1}\{c_0 + c_u + c^-\left[\left(1-p_1(f_i,0)\right)\max(d-b_1,0)\right] + \lambda\left[p_1(f_i,0)V_t(0,\min(b_1+s_1-d,B_1)) + \left(1-p_1(f_i,0)\right)V_t(0,\min(b_1-d,0))\right]\}, \quad (8.4)$$

$$V_{t+1}(a,b_1) = \min\left\{V_{t+1}^{\beta=0}(a,b_1), V_{t+1}^{\beta=1}(a,b_1)\right\}, \quad (8.5)$$

$$V_0(a,b_1) = 0. \quad (8.6)$$

where π_f is the optimal policy for bottleneck identification and mitigation rules' selection and π_β is the optimal policy for rules f_0 and f_1's updating. Equations 8.2 and 8.3 determine the optimal policy π_f when $\beta = 0$ and $\beta = 1$ respectively. Equation 8.5 determines the optimal value of β at state $\{a, b_1\}$, i.e. π_β. Equation 8.6 gives the initial condition.

8.3 The Structure of the Optimal Policies

In order to identify the structure of the optimal control policies, we introduce the following properties of the optimal expected discounted cost functions.

Lemma 8.1 *The cost function $V_t(a,b_1)$satisfies the following relations:*

a. $V_t(a,b_1) \geq V_t(a,b_1')$, $a \geq 0$ *and* $0 \leq b_1' \leq b_1 \leq B_1$,

b. $V_t(a,b_1) \leq V_t(a',b_1)$, $a' \geq a \geq 0$ *and* $0 \leq b_1 \leq B_1$.

Proof: We will prove the Lemma by induction. The Lemma holds for $t = 0$ based on the definition of $V_0(a,b_1) = 0$. Assume that the Lemma holds for $t-1, t > 0$. We will show that the Lemma also holds for t.

Part a: Based on Equations 8.3 and 8.4, we have:

$$V_{t+1}^{\beta=0}(a,b_1) = \min_{i=0,1}\{c_0 + c^-\left(1-p_1(f_i,a)\right)\max(d-b_1,0) + \lambda\left[p_1(f_i,a)V_t(a+1,\min(b_1+s_1-d,B_1)) + \left(1-p_1(f_i,a)\right)V_t(a+1,\min(b_1-d,0))\right]\}$$

$$\geq \min_{i=0,1}\{c_0 + c^-\left(1 - p_1(f_i, a)\right)\max(d - b_1', 0) + \lambda\left[p_1(f_i, a)V_t(a + 1, \min(b_1' + s_1 - d, B_1)) + \left(1 - p_1(f_i, a)\right)V_t(a + 1, \min(b_1' - d, 0))\right]\}$$

$$= V_{t+1}^{\beta=0}(a, b_1')$$

$$V_{t+1}^{\beta=1}(a, b_1) = \min_{i=0,1}\{c_0 + c^-\left(1 - p_1(f_i, 0)\right)\max(d - b_1, 0) + \lambda\left[p_1(f_i, 0)V_t(0, \min(b_1 + s_1 - d, B_1)) + \left(1 - p_1(f_i, 0)\right)V_t(0, \min(b_1 - d, 0))\right]\}$$

$$\geq \min_{i=0,1}\{c_0 + c^-\left(1 - p_1(f_i, 0)\right)\max(d - b_1', 0) + \lambda\left[p_1(f_i, 0)V_t(0, \min(b_1' + s_1 - d, B_1)) + \left(1 - p_1(f_i, 0)\right)V_t(0, \min(b_1' - d, 0))\right]\}$$

$$= V_{t+1}^{\beta=1}(a, b_1')$$

where $b_1' \geq b_1$. Therefore, based on Equation 8.6, we can prove:

$$V_t(a, b_1) = \min\left\{V_{t+1}^{\beta=0}(a, b_1), V_{t+1}^{\beta=1}(a, b_1)\right\}$$

$$\geq \min\left\{V_{t+1}^{\beta=0}(a, b_1'), V_{t+1}^{\beta=1}(a, b_1')\right\} = V_t(a, b_1'), b_1 \leq b_1'.$$

Part b: Based on Equation 8.3, we have:

$$V_{t+1}^{\beta=0}(a, b_1) = \min_{i=0,1}\{c_0 + c^-\left(1 - p_1(f_i, a)\right)\max(d - b_1, 0) + \lambda\left[p_1(f_i, a)V_t(a + 1, \min(b_1 + s_1 - d, B_1)) + \left(1 - p_1(f_i, a)\right)V_t(a + 1, \min(b_1 - d, 0))\right]\}$$

$$\leq \min_{i=0,1}\{c_0 + c^-\left[\left(1 - p_1(f_i, a')\right)\max(d - b_1, 0)\right] + \lambda\left[p_1(f_i, a)V_t(a' + 1, \min(b_1 + s_1 - d, B_1)) + \left(1 - p_1(f_i, a)\right)V_t(a' + 1, \min(b_1 - d, 0))\right]\}$$

$$= \min_{i=0,1}\left\{c_0 + c^-\left(1 - p_1(f_i, a')\right)\max(d - b_1, 0) + \lambda[V_t(a' + 1, \min(b_1 - d, 0)) - p_1(f_i, a)(V_t(a' + 1, \min(b_1 - d, 0)) - V_t(a' + 1, \min(b_1 + s_1 - d, B_1)))]\right\}$$

$$\leq \min_{i=0,1}\left\{c_0 + c^-\left(1 - p_1(f_i, a')\right)\max(d - b_1, 0) + \lambda[V_t(a' + 1, \min(b_1 - d, 0)) - p_1(f_i, a')(V_t(a' + 1, \min(b_1 - d, 0)) - V_t(a' + 1, \min(b_1 + s_1 - d, B_1)))]\right\}$$

$$= V_{t+1}^{\beta=0}(a', b_1)$$

where $a \leq a'$. The first inequality holds based on the assumption that $p_1(f_i, a)$ decreases when a increases, i.e. $p_1(f_i, a) \geq p_1(f_i, a')$ and the assumption that $V_t(a, b_1)$ increases when a increases. The second inequality holds based on Part a, i.e. $V_t(a' + 1, \min(b_1 - d, 0)) - V_t(a' + 1, \min(b_1 + s_1 - d, B_1)) > 0$, and the assumption that $p_1(f_i, a)$ decreases when a increases, i.e. $p_1(f_i, a) \geq p_1(f_i, a')$.

It is easy to prove that

$$V_{t+1}^{\beta=1}(a, b_1) = V_{t+1}^{\beta=1}(a', b_1), a \leq a'.$$

Therefore, based on Equation 8.6, we can prove:

$$V_t(a, b_1) = \min\left\{V_{t+1}^{\beta=0}(a, b_1), V_{t+1}^{\beta=1}(a, b_1)\right\}$$

$$\leq \min\left\{V_{t+1}^{\beta=0}(a + 1, b_1), V_{t+1}^{\beta=1}(a + 1, b_1)\right\} = V_t(a + 1, b_1), a \leq a'.$$

End of the proof.

Let $V(a, b_1)$ denote the minimum expected discounted cost with initial state being $\{a, b_1\} \in S$. From Proposition 3.1 in [69], it follows that $V(a, b_1)$ is $V(a, b_1) = \lim_{t\to\infty} V_t(a, b_1)$. Hence the Lemma 8.1 can be expressed with the following Lemma:

Lemma 8.2 *The cost function $V(a, b_1)$satisfies the following relations:*

a. $V(a, b_1) \geq V(a, b_1 + 1)$, $a \geq 0$ *and* $0 \leq b_1 \leq B_1 - 1$,

b. $V(a, b_1) \leq V(a + 1, b_1)$, $a \geq 0$ *and* $0 \leq b_1 \leq B_1 - 1$.

Lemma 8.2.a indicates that the expected discounted cost function $V(a, b_1)$ is non-increasing with respect to finished-goods buffer level b_1. Lemma 8.2.b indicates that the expected discounted cost function $V(a, b_1)$ is non-decreasing with respect to policy age a. As a consequence of Lemma 8.2, we can obtain the following Lemmas:

Lemma 8.3

a. $V^{\{f_0,\beta\}}(a, b_1) \leq V^{\{f_1,\beta\}}(a, b_1), 0 < a \leq t_0 \text{ or } \beta = 1$,

b. $V^{\{f_0,\beta\}}(a, b_1) \geq V^{\{f_1,\beta\}}(a, b_1), a > t_0 \text{ and } \beta = 0$.

Proof: We will prove Part a. Suppose policy age is less than the time threshold t_0, i.e. $0 < a \leq t_0$. Then based on the assumption $p_1(f_0, a) \geq p_1(f_1, a)$, Lemma 8.2.a and Equation 8.3 and 8.4, we have:

$$V^{\{f_0,\beta=0\}}(a, b_1) = \min_{i=0,1}\{c_0 + c^-\big(1 - p_1(f_0, a)\big)\max(d - b_1, 0) + \lambda\big[p_1(f_0, a)V(a + 1, \min(b_1 + s_1 - d, B_1)) + \big(1 - p_1(f_0, a)\big)V(a + 1, \min(b_1 - d, 0))\big]\}$$

$$\leq \min_{i=0,1}\{c_0 + c^-\big[\big(1 - p_1(f_1, a)\big)\max(d - b_1, 0)\big] + \lambda[V(a + 1, \min(b_1 - d, 0)) - p_1(f_1, a)(V_t(a + 1, \min(b_1 - d, 0)) - V_t(a + 1, \min(b_1 + s_1 - d, B_1)))]\}$$

$$= V^{\{f_1,\beta=0\}}(a, b_1).$$

$$V^{\{f_0,\beta=1\}}(a, b_1) = \min_{i=0,1}\{c_0 + c^-[(1 - p_1(f_0, 0)) \max(d - b_1, 0)] + \lambda[p_1(f_0, 0)V(0, \min(b_1 + s_1 - d, B_1)) + (1 - p_1(f_0, 0))V(0, \min(b_1 - d, 0))]\}$$

$$\leq \min_{i=0,1}\{c_0 + c^-[(1 - p_1(f_1, 0)) \max(d - b_1, 0)] + \lambda[V(0, \min(b_1 - d, 0)) - p_1(f_1, 0)(V_t(0, \min(b_1 - d, 0)) - V_t(0, \min(b_1 + s_1 - d, B_1)))]\}$$

$$= V^{\{f_1,\beta=1\}}(a, b_1).$$

The case when $\beta = 1$ and Part b can be proved similarly.

End of the proof.

Lemma 8.3 shows that if policy age a is less or equal to the threshold t_0, i.e. $0 \leq a \leq t_0$, or rules f_0 and f_1 are updated, i.e. $\beta = 1$, real-time EBM based rule f_0 is preferred. If policy age a is larger than the threshold t_0, i.e. $a > t_0$, and $\beta = 0$, stochastic EBM based rule f_1 is preferred. Therefore, the optimal control policy π_f is a control limit rule:

Proposition 8.1 *The optimal control policy π_f, as a function of policy age a and decision variable β, is a control limit rule. That is* $\pi_f = \begin{cases} f_0, & if\ 0 \leq a \leq t_0\ or\ \beta = 1 \\ f_1, & if\ a \geq t_0\ and\ \beta = 0 \end{cases}$.

Then we will prove that the optimal control policy π_β is also a control limit rule.

Lemma 8.4 If $V^{\beta=0}(a, b_1) \geq V^{\beta=1}(a, b_1)$, $0 \leq b_1 \leq B_1$, then $\forall a' \geq a$, $V^{\beta=0}(a', b_1) \geq V^{\beta=1}(a', b_1)$, $0 \leq b_1 \leq B_1$.

Proof: If $V^{\beta=0}(a, b_1) \geq V^{\beta=1}(a, b_1)$, then based on Equations 8.3 and 8.4, we have

$$V^{\beta=0}(a,b_1) = V^{\{\pi_f,\beta=0\}} = c_0 + c^-[\left(1-p_1\left(\pi_f,a\right)\right)\max(d-b_1,0)] + \lambda[p_1\left(\pi_f,a\right)V_t(a+1,\min(b_1+s_1-d,B_1)) + \left(1-p_1\left(\pi_f,a\right)\right)V_t(a+1,\min(b_1-d,0))]$$

$$\geq V^{\beta=1}(a,b_1) = V^{\{\pi_f,\beta=1\}} = c_0 + c_u + c^-\left[\left(1-p_1\left(\pi_f,0\right)\right)\max(d-b_1,0)\right] + \lambda\left[p_1\left(\pi_f,0\right)V(0,\min(b_1+s_1-d,B_1)) + \left(1-p_1\left(\pi_f,0\right)\right)V(0,\min(b_1-d,0))\right]$$

Then for $V^{\beta=0}(a',b_1), a' \geq a$, we have:

$$V^{\beta=0}(a',b_1) = V^{\{\pi_f,\beta=0\}} = c_0 + c^-[\left(1-p_1\left(\pi_f,a'\right)\right)\max(d-b_1,0)] + \lambda[p_1\left(\pi_f,a'\right)V_t(a'+1,\min(b_1+s_1-d,B_1)) + \left(1-p_1\left(\pi_f,a'\right)\right)V_t(a'+1,\min(b_1-d,0))]$$

$$\geq V^{\beta=0}(a,b_1) \geq V^{\beta=1}(a,b_1) = c_0 + c_u + c^-\left[\left(1-p_1\left(\pi_{f_i},0\right)\right)\max(d-b_1,0)\right] + \lambda\left[p_1\left(\pi_f,0\right)V(0,\min(b_1+s_1-d,B_1)) + \left(1-p_1\left(\pi_f,0\right)\right)V(0,\min(b_1-d,0))\right] = V^{\beta=1}(a',b_1).$$

The first inequality has been proved in Lemma 8.1 and the second inequality holds according to the assumption that $V^{\beta=0}(a,b_1) \geq V^{\beta=1}(a,b_1)$.

End of the proof.

According to Lemma 8.4, optimal policy π_β is a control limit rule. The following Proposition provides the mathematical expression of the optimal policy π_β.

Proposition 8.2 *For fixed buffer level b_1, optimal control policy π_β is a control limit rule, i.e.*

$$\pi_\beta = \begin{cases} 0, & 0 \leq a \leq a^*(\pi_f,b_1) \\ 1, & a > a^*(\pi_f,b_1) \end{cases}.$$

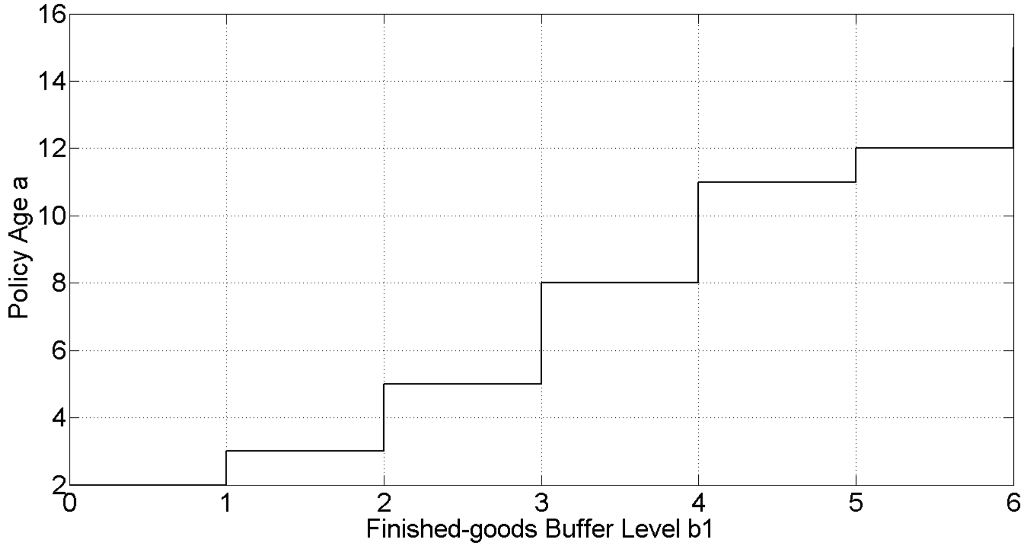


Figure 8.2 The graphic structure of optimal control policy π_β

It means for fixed buffer level b_1, there exists a critical policy age $a^*(\pi_f, b_1)$ such that if $a > a^*$, the optimal policy π_β requires an update, i.e. $\pi_\beta = 1$, and if $0 \leq a \leq a^*$, the optimal policy π_β does not require an update, i.e. $\pi_\beta = 0$. The critical policy age $a^*(\pi_f, b_1)$ is the least policy age a such that $\forall a' \geq a$, $V^{\beta=0}(a, b_1) \geq V^{\beta=1}(a, b_1)$. Mathematically, it can be expressed as $a^*(\pi_f, b_1) = \inf\{a | V^{\beta=0}(a, b_1) \geq V^{\beta=1}(a, b_1), a \geq 0\}$ and determined based on the policy iterative algorithm in [70].

Figure 8.2 shows the structure of the optimal control policy π_β graphically. The figure shows all the possible states with the buffer capacity being $B_1 = 6$. The critical switching line divides the whole state space into two regions. The upper part is the “updating region” and the lower part is the “not updating region”.

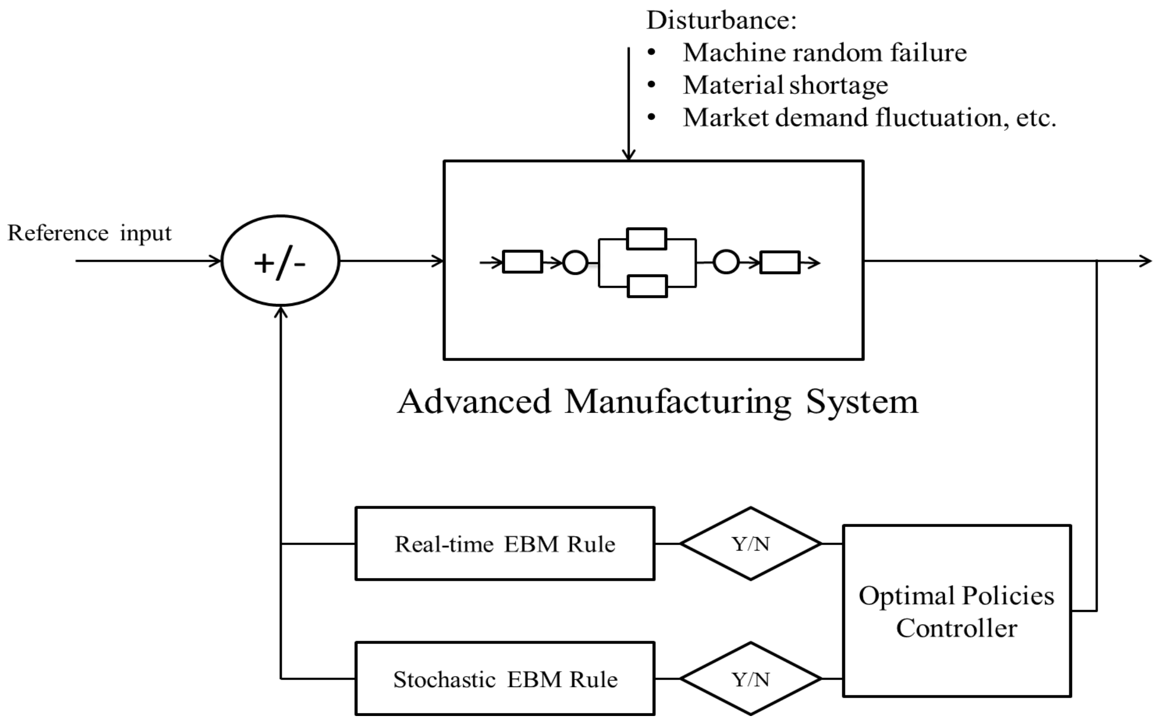


Figure 8.3 An improved supervisory control algorithm

8.4 An Improved Supervisory Control Scheme

We integrate the optimal control policies into the supervisory control scheme (introduced in Chapter 7) and show the improved supervisory control scheme in figure 8.3. Different from the control scheme in Chapter 7, we introduce a controller into the improved supervisory control scheme, which selects bottleneck identification and mitigation rules and decides whether to update the rules based on the optimal policies π_f and π_β.

Extensive numerical analysis has been performed to verify the effectiveness of the improved supervisory control scheme. In the analysis, we compare the market demand satisfaction and system cost changes when different control schemes (including the old supervisory control scheme, improved supervisory control scheme, first-come, first-serve policy, etc.) are applied. The results show that the improved supervisory control scheme can lead to the highest market demand satisfaction while keeping the lowest system cost.

8.5 Conclusion

In this chapter, we use Markov decision model to study the optimal selection of bottleneck identification and mitigation rules and the optimal control of their updating in an advanced manufacturing system. The system consists of two virtual machines, which model a production line and market demand and a finished-goods buffer. Using Markov decision technique, we have shown that for a fixed finished-goods buffer level, the optimal policies for bottleneck identification and mitigation rules' selection and their updating are control limit rules regarding to policy age.

The analysis answers the questions: "if real-time EBM and stochastic EBM based bottleneck identification and mitigation rules are different, which rules should be used," and "when should we update EBM based rules". We integrate the optimal control policies into the supervisory control algorithm in Chapter 7. Extensive numerical analysis shows that with the new control algorithm, system performance is further improved in terms of market demand satisfaction while system cost is significantly decreased.

Chapter 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

This dissertation focuses on real-time modeling and control of multistage manufacturing systems. Five related problems in the domain of system modeling and control are addressed. The major achievements of the dissertation can be summarized as follows:

1) The impact of disruption events in both general serial production lines and general parallel production lines has been addressed, where multiple slowest machines exists in both situations. The analysis unifies the general serial production lines and general parallel production lines and provides a quantifiable way to evaluate the impact of disruption events to system productivity. It indicates that not all the disruption events can cause permanent production loss in the system. Only the disruption events that force the last slowest machine to stop (through starvation, blockage or breakdown) contribute to the permanent production loss at a production line. And the impact of a disruption event can be quantified with the permanent production loss caused by the disruption event, which is defined as the production loss at the last slowest machine caused by the event.

2) A sensor system with three virtual layers is developed for multi-stage manufacturing systems. The first two virtual layers convert the material flow into information flow and send the information to the third virtual layer. The sensor information is further transferred into true knowledge in the third layer through an integrated system modeling approach, i.e. EBM. Dynamic EBM naturally integrates two important system considerations, i.e. system capacity

and production loss. A unified system performance index, i.e. permanent production loss caused by disruption events, stations and supporting activities provides a natural severity ranking of all the stations and supporting activities.

3) The definition of SAT and an estimation method are developed to evaluate the performance of a subsystem. The SAT of a subsystem is defined as the productivity of the line segment while not being affected by its upstream and downstream machines. The estimation method provides a data-driven algorithm through utilizing production online data to quickly and accurately evaluate SAT of a subsystem. The analysis provides important insights in understanding production line dynamics. This quantitative tool can help production managers better identify problems and improve overall system performance.

4) The market demand satisfaction has been analyzed in the dissertation. A market demand driven system is developed by modeling market demand as the end-of-line virtual machine. In such a system, any market demand dissatisfaction, i.e. MDD, can be measured as the production loss at the end-of-line virtual machine. This unifies the analysis of market demand satisfaction and system productivity. An event-based methodology is developed to categorize the MDD caused by machine capacity bottlenecks (MC-BNs) and machine failure bottlenecks (MF-BNs). Indicators for MC-BN and MF-BN identification are proposed. It is shown that the MC-BN and MF-BN identification indicators are related to DEMCs and DEMFs. Based on the identification indicators, an iterative procedure and a data-driven method are developed to identify all the independent MC-BNs and MF-BNs. A supervisory control algorithm to identify and mitigate MC-BNs and MF-BNs is introduced. The case study confirms that the supervisory control focusing on MC-BNs and MF-BNs can most efficiently increase system throughput and reduce MDD.

5) A Markov decision model has been developed to find the optimal policies for bottleneck identification and mitigation rules' selection and their updating in an advanced manufacturing system. The analysis answers the questions in the supervisory control scheme in Chapter 7 as: "if real-time EBM and stochastic EBM based bottleneck identification and mitigation rules are different, which rules should be used," and "when should we update EBM based rules". Based on the analysis, an improved supervisory control scheme is developed by integrating the optimal control policies integrated into previous supervisory control algorithm. Extensive numerical analysis shows that with the new control algorithm, system performance is further improved in terms of market demand satisfaction while system cost is significantly decreased.

The original contribution of the dissertation is summarized as follows:

1) An event-based modeling (EBM) approach has been developed as an integrated real-time system modeling method. It is a comprehensive system modeling method, which takes into considerations of every aspects of a manufacturing system, including machines, supporting activities, etc. EBM approach is novel because it is a data driven method which transfers the sensor information into useful managerial knowledge in real-time. It provides a step forward in real-time modeling of dynamic systems.

2) A supervisory control algorithm is developed to integrate EBM, the integrated real-time system modeling tool, into system control. The algorithm analyzes the most up-to-date system dynamics with EBM and makes proper corrective control decisions based on this analysis. Comparing with the control methods based on long-term steady state analysis, the

supervisory control algorithm provides a more comprehensive control strategy especially for dynamic systems with unpredicted changes, random disturbances and economy oscillations.

9.2 Future Work

Future work could be conducted in the following areas:

1) Extend the EBM method to the "integrated manufacturing system", which includes production systems and supply chains. The success of an advanced manufacturing system not only depends on the smooth operation within the production system, but also relies on the close cooperation between the production system and supply chains. However, in most existing literature, the modeling and control of supply chain system and production system are analyzed separately. The analysis can be useful in improving individual production system and supply chain system's performance. However, it is difficult to extend the methods to analyze the integrated manufacturing system. To achieve the overall system efficiency, it is necessary to extend the EBM method to the integrated manufacturing system.
2) In Chapter 8, we use stationary Markov decision model to find the optimal control policies. The analysis is useful for system control in any large time scales, but may have a poor control of the system during small time segments. To overcome the problem, in the future, we will use more advanced decision theories, such as weighted Markov decision theory, to provide more comprehensive control strategies.

REFERENCE

[1] K. Efthymiou, P. A., P. N., M. D., and G. Chryssolouris, "Manufacturing Systems Complexity Review: Challenges and Outlook," in *Procedia CIRP 3*, 2012.

[2] Q. Chang, J. Ni, P. Bandyopadhyay, S. Biller, and G. Xiao, "Maintenance Opportunity Planning System," *ASME Transaction, Journal of Manufacturing Science and Engineering,* vol. 129, pp. 661-668, 2007.

[3] Q. Chang, J. Ni, P. Bandyopadhyay, S. Biller, and G. X. Xiao, "Supervisory factory control based on real-time production feedback," *Journal of Manufacturing Science and Engineering-Transactions of the Asme,* vol. 129, pp. 653-660, Jun 2007.

[4] Y. Li, Q. Chang, G. Xiao, and S. Biller, "Event-based modelling of distributed sensor networks in battery manufacturing," *International Journal of Production Research, ahead-of-print (2014)*, pp. 1-14, 2014.

[5] Y. Ding, D. Ceglarek, and J. Shi, "Fault Diagnosis of Multistage Manufacturing Processes by Using State Space Approach," *ASME Transactions, Journal of Manufacturing Science and Engineering,* vol. 124, pp. 313-322, 2002.

[6] Y. Ding, E. A. Elsayed, S. Kumara, J.-C. Lu, F. Niu, and J. J. Shi, "Distributed Sensing for Quality and Productivity Improvements," *IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING,* vol. 3, pp. 344-359, 2006.

[7] J. Liu, Q. Chang, G. Xiao, and S. Biller, "The Costs of Downtime Incidents in Serial Multi-Stage Manufacturing Systems," *ASME Transaction, Journal of Manufacturing Science and Engineering,* vol. 134, pp. 02-16, 2012.

[8] R. Langer, J. Li, S. Biller, Q. Chang, N. Huang, and G. Xiao, "Simulation Study of a Bottleneck-Based Dispatching Policy for a Maintenance Workforce," *International Journal of Production Research,* vol. 48, pp. 1745-1763, 2010.

[9] A. C. A. Cauvin, A. F. A. Ferrarini, and E. T. E. Tranvouez, "Disruption management in distributed enterprises: A multi-agent modelling and simulation of cooperative recovery behaviours," *International Journal of Production Economics,* vol. 122, pp. 429-439, Nov 2009.

[10] Q. Chang, S. Biller, and G. X. Xiao, "Transient Analysis of Downtimes and Bottleneck Dynamics in Serial Manufacturing Systems," *Journal of Manufacturing Science and Engineering-Transactions of the Asme,* vol. 132, Oct 2010.

[11] X. Gu, X. Jin, and J. ni, "Resilience Measures of Manufacturing Systems Under Disruptions," in *Proceedings of the 2014 International Conference on Manufacturing Science and Engineering*, Detroit, Michigan, USA, 2014.

[12] H. Hishamuddin, R. A. Sarker, and D. Essam, "A disruption recovery model for a single stage production-inventory system," *European Journal of Operational Research,* vol. 222, pp. 464-473, Nov 1 2012.

[13] T. Wu, J. Blackhurst, and P. O'grady., "Methodology for supply chain disruption analysis," *International Journal of Production Research,* vol. 45, pp. 1665-1682, 2007.

[14] J. W. Fowler and O. Rose, "Grand challenges in modeling and simulation of complex manufacturing systems," *Simulation,* vol. 80, pp. 469-476, 2004.

[15] S. B. Gershwin, *Manufacturing systems engineering*. Engelwood Cliffs, NJ: PTR Prentice Hall, 1994.

[16] J. Li, E. B. Dennis, N. Huang, and M. A. Jeffrey, "Throughput analysis of production systems: recent advances and future topics," *International Journal of Production Research,* vol. 47, pp. 3823-2851, 2009.
[17] J. Li and S. M. Meerkov, *Production systems engineering*. Livemore, CA: Wingspan Press, 2007.
[18] J. Heilala, H. Helaakoski, and I. Peltomaa, "Smart assembly C data and model driven," presented at the IFIP Proceedings of International Federation for Information Processing, 2008.
[19] L. Schruben, "Mathematical Programming Models of Discrete Event System Dynamics," in *Proceedings of the 2000 Winter Simulation Conference,* Piscataway, NJ, USA, 2000, pp. 381-385.
[20] B. Tjahjono and R. F. ndez, "Practical approach to experimentation in a simulation study," in *Proceedings of the Winter Simulation Conference*, 2008, pp. 1981-1988.
[21] S. Lee, H. Cho, and M. Jung, "A conceptual framework for the generation of simulation models from process plans and resource configuration," *International Journal of Production Research,* vol. 38, pp. 811–828, 2000.
[22] S. R. Das, "Adaptive Protocols for Parallel Discrete Event Simulation," *Journal of the Operational Research Society,* vol. 51, pp. 385-394, 2000.
[23] R. M. Fujimoto, *Parallel and Distributed Simulation Systems*: Wiley-Interscience, 2000.
[24] L. S. Franz, "Data diven modeling: an application in scheduling," *Decision Sciences,* vol. 20, pp. 359-377, 1989.
[25] S. B. Gershwin, "An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking," *Operations Research,* vol. 35, pp. 291-305, 1987a.
[26] S. B. Gershwin, "Representation and analysis of transfer lines with machines that have different processing rates," *Annals of Operations Research,* vol. 9, pp. 511-530, 1987b.
[27] Y. Dallery, R. David, and X. L. Xie, "An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers," *IIE Transaction,* vol. 20, pp. 280-283, 1988.
[28] Y. Dallery, R. David, and X. L. Xie, "Approximate analysis of transfer lines with unreliable machines and finite buffers," *Ieee Transactions on Automatic Control,* vol. 34, pp. 943-953, 1989.
[29] M. H. Burman, "New results in flow line analysis," PhD, OR Center, MIT, 1995.
[30] J. Li and S. M. Meerkov, "Customer demand satisfaction in production systems: a due-time performance approach," *IEEE Transactions on Robotics and Automation,* vol. 17, pp. 472-482, 2001.
[31] J. Li, "Production variability in manufacturing systems: a system approach," PhD, University of Michigan, 2000.
[32] T. Altiok, *Performance analysis of manufacturing systems*. Berlin: Springer, 1997.
[33] M. Di Mascolo, R. David, and Y. Dallery, "Modeling and analysis of assembly systems with unreliable machines and finite buffers," *IIE Transaction,* vol. 23, pp. 315-330, 1991.
[34] S. B. Gershwin and M. H. Burman, "A decomposition method for analyzing inhomogeneous assembly/disassembly systems," *Annals of Operations Research,* vol. 93, pp. 91-115, 2000.
[35] J. Li, "Performance analysis of manufacturing systems with rework loops," *IIE Transaction,* vol. 36, pp. 755-765, 2004a.

[36] J. Li, "Modeling and analysis of manufacturing systems with parallel lines," *IIE Transaction*, vol. 49, pp. 1824-1829, 2004b.
[37] J. Li, "Overlapping decomposition: a system-theoretic method for modeling and analysis of complex production systems," *IEEE Transactions on Automation Sciences and Engineering*, vol. 2, pp. 40-53, 2005.
[38] W. J. Hopp and J. T. Simon, "Estimating throughput in an unbalanced assembly-like flow system," *INT.J.PROD.RES.*, vol. 31, pp. 4851-4868, 1993.
[39] Y. Arzi and A. Kalir, "Approximating throughput of non-homogeneous multiple-parallel machines' production lines," Department of Industrial Engineering, Tel Aviv University, Tel Aviv, Israel1996.
[40] A. Patchong and D. Willaeys, "Modeling and analysis of an unreliable flow line composed of parallel-machine stages," *IIE Transaction*, vol. 33, pp. 559-568, 2001.
[41] S. M. Meerkov and L. Zhang, "Transient Behavior of Serial Production Lines With Bernoulli Machines," *IIE Trans.*, vol. 40, pp. 297–312, 2008.
[42] L. Zhang, C. Wang, J. Arinez, and S. Biller., "Transient analysis of Bernoulli serial lines: performance evaluation and system-theoretic properties," *IIE Transactions*, vol. 45, pp. 528-543, 2013.
[43] S. Ahn and V. Ramaswami, "Transient analysis of fluid flow models via stochastic coupling to a queue," *Stochastic Models*, vol. 20, pp. 71-101, 2004.
[44] H. Kobayashi and Q. Ren, "A mathematical theory for transient analysis of communication networks," *IEICE Transactions on Communication*, vol. E75-B, pp. 1266-1276, 1992.
[45] B. Sericola, "Transient analysis of stochastic fluid models," *Performance Evaluation*, vol. 32, pp. 245-263, 1998.
[46] S. M. Meerkov and L. Zhang, "Transient behavior of serial production lines with Bernoulli machines," *IIE Transactions*, vol. 40, pp. 297-312, 2008.
[47] S. M. Meerkov, N. Shimkin, and L. Zhang, "Transient Behavior of Two-Machine Geometric Production Lines," *IEEE Transactions on Automatic Control*, vol. 55, pp. 453-458, 2010.
[48] S. M. Meerkov and L. Zhang, "Product quality inspection in Bernoulli lines: Analysis, bottlenecks, and design," *Int. J. Prod. Res.*, vol. 48, pp. 4745-4766, 2010.
[49] J. S. Fenner, M. K. Jeong, and J.-C. Lu, "Multi-Stage optimal automatic control," *School Ind. Syst. Eng., Georgia Inst. Technol., Atlanta, GA*, 2004.
[50] M. L. Pinedo, *Planning and Scheduling in Manufacturing and Services, Springer*. New York: Springer, 2005.
[51] J. Shi and S. Zhou, "Quality Control and Improvement for Multistage Systems: A Survey," *IIE Transactions*, vol. 41, pp. 744-753, 2009.
[52] D. A. Jacobs and S. M. Meerkov, "A system-theoretic property of serial production lines: improvability," *International Journal of System Science*, vol. 26, pp. 95–137, 1995a.
[53] D. A. Jacobs and S. M. Meerkov, "Mathematical theory of improvability for production systems," *Mathematical Problems in Engineering*, vol. 1, pp. 95–137, 1995b.
[54] G. J. Beaujon and V. R. Singhal, "Representing activities in activities based costing systems.," General Motors1989.
[55] S. Y. Chiang, "Bottlenecks in production systems with Markovian machines: theory and applications," PhD, University of Michigan, 1999.

[56] S. Y. Chiang, C. T. Kuo, J. T. Lim, and S. M. Meerkov, "Improvability theory of assembly systems. I: Problem formulation and performance evaluation," *Mathematical Problems in Engineering,* vol. 6, pp. 321-357, 2000b.
[57] S. Y. Chiang, C. T. Kuo, and S. M. Meerkov, "Bottlenecks in Markovian production lines: A systems approach," *Ieee Transactions on Robotics and Automation,* vol. 14, pp. 352-359, Apr 1998.
[58] S. Y. Chiang, C. T. Kuo, and S. M. Meerkov, "DT-bottlenecks in serial production lines: Theory and application," *Ieee Transactions on Robotics and Automation,* vol. 16, pp. 567-580, Oct 2000.
[59] S. Y. Chiang, C. T. Kuo, and S. M. Meerkov, "c-bottlenecks in serial production lines: Identification and application," *Mathematical Problems in Engineering,* vol. 7, pp. 543-578, 2001.
[60] A. M. Bonvik, C. E. Couch, and S. B. Gershwin, "A comparison of production-line control mechanisms," *International journal of production research,* vol. 35, pp. 789-804, 1996.
[61] L. Zhang, "Feedback release control of Bernoulli production lines," presented at the Intelligent Control and Automation (WCICA), 2011 9th World Congress on. IEEE, 2011.
[62] R. Akella and P. R. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *Automatic Control, IEEE Transactions on,* vol. 31, pp. 116-126, 1986.
[63] S. C. Kutzner and G. P. Kiesmüller, "Optimal control of an inventory-production system with state-dependent random yield," *European Journal of Operational Research,* vol. 227, pp. 444-452, 2013.
[64] F. A. V. d. D. Schouten and S. G. Vanneste, "Maintenance optimization of a production system with buffer capacity," *European journal of operational research,* vol. 82, pp. 323-338, 1995.
[65] S. M. Iravani and I. Duenyas, "Integrated maintenance and production control of a deteriorating production system," *IIE Transactions,* vol. 34, pp. 423-435, 2002.
[66] X. Jin, S. J. Hu, J. Ni, and G. Xia, "Assembly Strategies for Remanufacturing Systems With Variable Quality Returns," *Automation Science and Engineering, IEEE Transactions o,* vol. 10, pp. 76-85, 2013.
[67] M. Colledani and S. B. Gershwin, "A decomposition method for approximate evaluation of continuous flow multi-stage lines with general Markovian machines," *Annals of Operations Research,* vol. 209, pp. 5-40, 2013.
[68] B. Tan and S. B. Gershwin, "Analysis of a general Markovian two-stage continuous-flow production system with a finite buffer," *International Journal of Production Economics,* vol. 120, pp. 327-339, 2009.
[69] S. M. Ross, *Introduction to Stochastic Dynamic Programming*. New York: Academic Press, 1983.
[70] E. G. Kyriakidis and T. D. Dimitrakos, "Optimal preventive maintenance of a production system with an intermediate buffer," *European Journal of Operational Research,* vol. 168, pp. 86-99, 2006.