4-19-2017

Transient Performance Evaluation, Bottleneck Analysis and Control of Production Systems

Zhiyang Jia
University of Connecticut - Storrs, zhiyang.jia@uconn.edu

Transient Performance Evaluation, Bottleneck Analysis and Control of Production Systems

Zhiyang Jia, PhD

University of Connecticut, 2017

The focus of production system is to analyze, improve, and control the flow of products in the manufacturing process. In the field of study, the major difficulties are unreliable machines and finite buffers capacities, which lead to nonlinear and stochastic mathematical models. Extensive results on production systems have been derived for steady state operations, while their transient performance and properties are also of practical importance but paid with significantly less attention.

In this dissertation, we study the problems of transient performance evaluation, bottleneck analysis, and production control of serial lines, closed lines and assembly systems. Specifically, in the framework of finite production run-based serial lines with Bernoulli/ geometric machine reliability model, we derive mathematical model and analytical formulas to evaluate the performance measures of small systems. Then, we propose computationally efficient algorithms based on decomposition and aggregation for large systems, to approximate the systems performance measures with high accuracy. System-theoretic properties and bottleneck problems are also discussed. For closed lines and assembly systems, based on Markovian analysis, we develop the mathematical models and propose approximation methods for transient performance evaluation. For serial lines with Bernoulli machines and with operation control, mathematical models for the system under consideration are derived and analytical methods are developed for calculating the system transient performance.

One effective and efficient approach of analyzing transient performance of serial lines, closed lines and assembly systems is present in this dissertation. Bottlenecks, theoretic properties and control of the systems are studied under the transient analysis. Extension of the results to systems with exponential or non-Markovian models, adaptive control of machines, continuous improvement of systems, etc. can be further studied in future work.

Transient Performance Evaluation, Bottleneck Analysis and Control of Production Systems

Zhiyang Jia

B.S., Northwestern Polytechnical Universtiy, 2010

M.S., Beijing University of Technology, 2013

M.S., University of Connecticut, 2016

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2017

APPROVAL PAGE

Doctor of Philosophy Dissertation

Transient Performance Evaluation, Bottleneck Analysis and Control of Production Systems

Presented by

Zhiyang Jia, B.S., M.S.

Major Advisor_____

Liang Zhang

Associate Advisor_____

Peter B. Luh

Associate Advisor_____

Ashwin Dani

University of Connecticut

2017

To my family

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Professor Liang Zhang for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research during my four years Ph.D study at University of Connecticut. I am highly indebted and thoroughly grateful to him for providing me with materials and insights that I could not possibly have discovered on my own, for his helpful words and beneficial suggestions. It was a great pleasure working with him and I could not have imagined having a better advisor and mentor.

Besides my advisor, profound gratitude goes to Professor Peter B. Luh and Professor Ashwin Dani, who have been truly dedicated mentors. I would like to express the deepest appreciation to them for their insightful comments and guidance in my research, courses, and Ph.D general exam. Without their guidance and persistent help my Ph.D career would not have been successfully.

I would like to express my gratitude to all professors in the Department of Electrical and Computer Engineering at University of Connecticut who imparted knowledge from all professional fields and put their faith on me and urged me to do better. Specially cherish the memory of Dr. Robert S. Lynch Jr., who was my instructor on courses Linear System Theory and Digital Signal Processing, but passed away in 2015 after long illness. He was such a respected and energetic instructor and gave me invaluable approval and confidence during my first year studying in the United States, which was the hardest time in a foreign country to me.

My sincere thanks also goes to Professor Jinshan Li, Dr. Feng Ju, and Dr. Xiang Zhong,

who exchanged experiences, ideas and discussed with me during several conferences. I am also very appreciative to Professor Semyon Meerkov, who is one of the distinguished experts in my research area, especially for giving me his suggestions and expectations on researching and studying. I would also like to thank to National Science Foundation for their financial support granted through doctoral fellowship.

Finally I would like to thank all my friends and colleagues at University of Connecticut. Countless happy laughter and cheerful voices with all of you in the past four years, as well as indispensable assistance and help on learning and researching. Last but not the least, I would like to thank my family: my mother Zunzhi Bao, my father Jianjun Jia and my wife Tuoxi Wu, for supporting me spiritually throughout pursuing my Ph.D degree at University of Connecticut and my life in general. They are the most important people in my world and I dedicate this thesis to them.

Zhiyang Jia

April 2017

University of Connecticut

Contents

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

Production systems are machines and material handling devices arranged so as to produce desired products. The machines can be a group of processing units such as human operators, machines, cells, etc., and the material handling devices can be boxes, shelves, carts, conveyors, automated guided vehicles, etc. Each machine is characterized by its technological operation(s), capacity to produce parts, and reliability and quality characteristics; each material handling device is characterized by its technology employed for storing and moving parts and buffer capacity, i.e., capacity to store work-in-process between each pair of consecutive operations.

Production system is one of the major parts of manufacturing research and practice. The focus of this field is to analyze, improve, and control the flow of products in the manufacturing process. In the field of study, if all machines are perfectly reliable and all buffers are of infinite capacity, then the significance of the research will be trivial. Unfortunately, the facts and also the main difficulties of production systems research are that, unreliable machines are typically in practice and buffers capacities are generally finite. These considerations make the problem of parts flow much more complicated. In more specific terms, the difficulties in investigating production systems arise due to mutual interferences of the machines because

of breakdowns. Indeed, the breakdown of one machine may affect other machines in the system, by blocking those upstream and starving those downstream. Buffers are used to alleviate these perturbations. However, having the buffers "infinite" and, hence, efficient for alleviating the perturbations, creates economic problems. Therefore, the buffers must be finite and the machines, obviously cannot be made absolutely reliable. These features lead to mathematical models of production systems nonlinear and stochastic which are difficult to analyze.

During the past 60 years, extensive results have been derived for modeling, analysis, improvement, design, and control of production systems (see, for instance, monographs [1–9]). While the majority of the studies were carried out for steady state operations. Although it is usually difficult to claim that a production system is in steady state from the practical perspective, the steady state analysis approach is sufficiently effective and accurate for manufacturing systems with relatively large production volume. The large production volume allows the system transient to decay in a period negligible compared to the overall production run-time, and, thus, renders it valid to use steady state approach.

It should be noted that the transient performance and properties of production systems are also of practical importance. For example, if the steady state is reached after a relatively long period of time, the system may suffer substantial throughput losses compared to the steady state level. In this case, the steady state analysis can not provide an accurate analysis due to the long period of transient process. This scenario is commonly observed in production systems with perishable products (e.g., automotive paint shops), where some of the buffers must be depleted at the end of each shift to avoid quality deterioration. This results in some of the buffers being empty at the beginning of the next shift. It has been reported that the throughput loss due to empty initial buffers may be as high as 10% of its production in a shift [9]. In another example of production system transients, consider a manufacturing process where the production is carried out based on production runs (one production run is sometimes referred to as a batch, or a lot). Typically, a certain number of identical or

similar goods are grouped together based on customer orders or demand forecast to form a production run. Then, production runs are entered into the manufacturing systems to be completed. In many cases, process changeover and transitions are necessary between two consecutive production runs. As a result, while the work-in-process starts to accumulate at the beginning of a production run, the manufacturing system may operate in a transient regime that is qualitatively different from its steady state behavior. Similar transients may be observed towards the end of a production run, when the work-in-porcess is purged.

But unfortunately, compared to the large amount of results obtained in steady state analysis of production systems, research on production system transients is still largely unexplored. Indeed, transient behavior of production systems remains largely unexplored and it is deemed as one of the most important directions in production systems research [10]. Transient behavior of single-server and single-stage queueing systems has been studied in a number of publications (see, for instance, [11–16]). The results on multi-stage production systems, however, have been relatively limited. Among the available literature on this topic, papers [17, 18] study the transient behavior of two-stage tandem queues with no in-process buffer. It is assumed in both papers that the machines are reliable and have exponentially distributed processing times. Paper [19] studies the transient evolution of a buffer with Markov-modulated input and output flows and derives an algorithm for solving the partial differential equation of the buffer occupancy's probability density function. In addition to these analytical studies, computer simulation has been used in the investigation of production system transients (see, for instance, [20–23]). Moreover, time series analysis (ARMA models) is used in the research [24] to approximate the transient performance of production systems. However, this method requires a given simulation model for the system under consideration in order to "train" the time series model.

Therefore, transient-based performance evaluation, bottleneck analysis, and control of production systems are investigated in-depth in our research.

1.2 Problems Addressed

In this research, we will study the problems of performance evaluation, bottleneck analysis, and production control of the production systems. Each of them is detailed as follows.

1.2.1 Performance evaluation

The first and most fundamental aspect of studying a production system is to develop methods to calculate its performance measures as functions of machine and buffer parameters. In terms of production system transients, this amounts to "predicting" the future performance of the system based on the current initial condition. This problem is not trivial due to the stochastic and nonlinear features of the systems at hand. The definitions of a few most important performance measures of interest for a production system during transients are summarized as follows:

Throughput, $TP(t)$: Average number of parts produced by the last machine of a production system per unit of time at (future) time instant t.

Work-in-process of the i-th buffer, $WIP_i(t)$: Average number of parts contained in the i-th in-process buffer of a production system at (future) time instant t.

Blockage of machine i, $BL_i(t)$: Probability that machine i is up, buffer i is full, and machine $i+1$ does not take a part from the buffer at (future) time instant t.

Starvation of machine i, $ST_i(t)$: Probability that machine i is up and buffer i-1 is empty at (future) time instant t.

Completion time of machine i, CT_i: Average amount of time for machine i to finish one production run.

1.2.2 Bottleneck analysis

In a production system, there often exists a certain operation that is the limiting factor to achieve a greater overall systems performance. Such operation is usually referred to as

the bottleneck of the system. In this research, we define the bottleneck as the machine such that changing its parameters α leads to greater effect on system performance index Ψ (e.g., steady state production rate, throughput, energy consumption, etc.) compared to changing the corresponding parameter of other machines in the system. This implies that machine m_i is the bottleneck if

$$\left|\frac{\partial \Psi}{\partial \alpha_i}\right| > \left|\frac{\partial \Psi}{\partial \alpha_j}\right|, \quad \forall j \neq i. \tag{1.1}$$

Clearly, improving the bottleneck machine defined above can provide the highest return in desired system performance. Indeed, bottleneck identification and mitigation is one of the most important task in production management and operation.

1.2.3 Production control

Production control are crucial for meeting increasingly high customer demands and expectations in the present, highly competitive, manufacturing areas. The goals include maximizing production rate, minimizing the completion time, reducing work-in-process, improving responsiveness to changes in demand, etc. To carry out effective and efficient production control, the system states should be continuously monitored so that the control actions can be made according to the real time feedback information of the system.

In practice, production control can be implemented in various forms. For example, in production systems with energy-intensive operations, the startup and shutdown of these operations should be carried out based on the system status to avoid energy waste while maintaining desired production level. Effective control in this regard has the potential to reduce equipment idle running time, increase the life cycle of the machines, and improve system energy efficiency. Indeed, production control-based shop floor continuous improvement is recognized as one of the most cost-effective ways to achieve energy-efficient production [25].

1.3 Literature Review

1.3.1 Performance evaluation

Performance evaluation of production systems during steady state have been extensively investigated. In addition to the countless results obtained on serial lines (see, for instance, [1–5]), the problem of steady state performance evaluation in assembly systems with unreliable machines and finite buffers has been studied in [26–31]. As for the closed lines, Lim and Meerkov analyse an asymptotic reliable two-machine, two-buffer closed serial line [32]. A case study at an automotive paint shop is described. Frien *et al.* present a decomposition approach to approximate the system production rate for homogeneous production lines [33]. Gershwin *et al.* [34] extend the decomposition approach to study closed-loop systems with geometric machines. Meanwhile, the majority of the results reported in the literature on analysis of re-entrant lines are obtained by simulations (e.g., [35–37]).

Compared to the large amount of results obtained in steady state performance evaluation of production systems, research on the transients behavior is still largely unexplored. Available results include analytical studies on transients of single-server or single-stage queueing systems (see, for instance, [11,12]) and bufferless two-stage production systems (see [17,19]). For multi-stage production systems, numerical and simulation investigations are reported in [20, 38], while analytical investigations have been carried out in [39–43]. Case studies based on production system transients have been reported in [25, 44]. In addition, it should be noted that, to the best of our knowledge, analytical study of assembly system transients only appears in paper [45], which studies the transient throughput of a class of one-server Markovian assembly-like queue with infinite queueing capacity. Also, despite these valuable results, the systems considered are usually assumed to have infinite supply of raw parts. Systems with the finite production run has rarely been addressed yet.

1.3.2 Bottleneck analysis

Bottleneck detection is extensively studied in the literature and also in practice. Methods to detect the bottleneck in a system by measuring either the waiting time in front of a machine or the workload represented by the percentage of the time a machine is active was described in [46]. An approach to determine the likelihood of multiple bottlenecks using a bottleneck probability matrix was describe in [47]. The bottleneck detection method from [47] also investigates all possible combinations of bottlenecks, which rapidly becomes more complicated for larger systems. There is also the possibility to detect the bottleneck by analyzing the structure of the system, [48] for example. Besides the methods described above, the active period method developed in [49] at Toyota Central Research and Development Laboratories, is based on the analysis of machine status information determining periods during which a machine is active without interruption. The approach described in [50] is based on the evaluation of the so-called criticality indicator for each workplace and comparison of the indicator values to detect the critical place.

Rigorous study of bottleneck in production systems with unreliable machines and finite buffers is initiated in [51], which developed an indirect method of bottlenecks identification for open serial lines. Specifically, using the largest sensitivity of the system performance index with respect to the isolation production rate of each machine as the definition of a bottleneck, it is shown in this work that the location of the bottleneck in a serial production line can be determined by analyzing the frequencies of blockages and starvations of each machine. Since these frequencies could be either measured (real-time operation environment) or calculated, this offers a simple but accurate tool for bottleneck identification. Built on the concept of [51], paper [52] develops a method for identification of bottleneck with respect to machine capacity. The method is applied in a case study of a camshaft production line at an automotive engine plant. The method is later extended to assembly systems in [53] and to serial lines with rework in [54]. It should be noted that, all the bottleneck analysis

of the systems are carried out under steady state. To the best of our knowledge, there is no investigation on any bottleneck during systems transients.

1.3.3 Production control

There has been an increasing interest in devising optimal production control policies that manage production in uncertain environments since the 1980s. An optimal flow rate control problem for a failure-prone machine subject to a constant demand is introduced in [55, 56]. The single-part type, single-machine problem is analyzed in detail in [57]. This work is extended to the case in which no backlog is allowed [58] and to the case of limited backlog [59]. In addition, this optimal production control problems with random demand is studied in [60, 61]. Moreover, the performance of the kanban, minimal blocking, basestock, CONWIP, and hybrid kanban-CONWIP control policies in a four-machine tandem production line making parts for an automobile assembly line is studied in [62]. A framework for scheduling discrete events in manufacturing systems is described in [63]. It should be noted that, nowadays effective control of production operations is also considered as one of the most economical methods to improve energy efficiency in manufacturing systems, while most manufacturing execution systems currently used in practice have no module or function to deal with energy management during operation [64]. Among limited results addressing this issue, an analytical model by combining an M/M/1 model with an energy control policy is developed by [65]. For the problem of scheduling startup and shutdown of machines in Bernoulli serial lines, a constrained optimization problem is formulated and studied by [66]. In [67], several switch-off dispatching policies for a non-bottleneck machine in a job shop to minimize its energy consumption was studied.

Note that the majority of the analysis of production systems with particular control are also carried out under the assumption, that the system is in steady state, while it is still lack of investigations on transient performance measures and system properties of the systems with control.

Chapter 2

SYSTEM MODELS AND APPROACHES OF THE RESEARCH

2.1 Mathematical Modeling

2.1.1 System structures

- **Serial Lines**

The serial line structure is the most commonly used one seen in production systems (see Figure 2.1, where circles represent machines and rectangles represent buffers). It should be noted that the "straight line" structure of Figure 2.1 is purely conceptual abstraction. In practice, the physics layout of a serial line may follow other shapes (e.g., L-shape, U-shape, S-shapes).

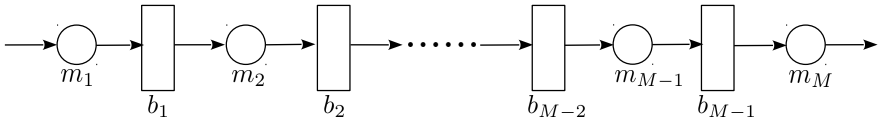
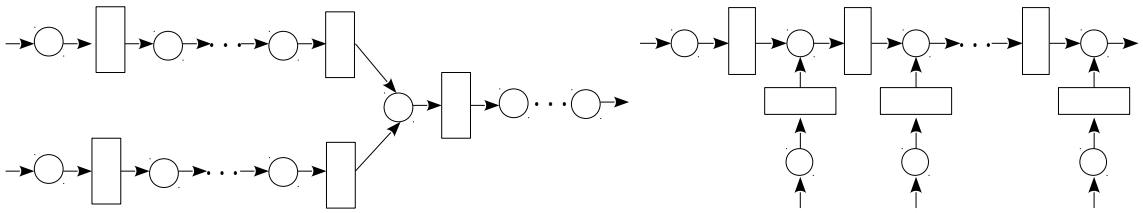


Figure 2.1: Serial production line

- **Assembly Systems**

An assembly system consists of two or more serial lines, referred to as component lines, one or more merge operations, where the components are assembled, and several subsequent

machines to perform additional processing on the assembled parts. Figure 2.2 shows the



(a) Single merge operation (b) Multiple merge operation

Figure 2.2: Assembly systems

block diagrams of typical assembly systems, where (a) demonstrates a simple structure of assembly systems with only one merge operation and two component lines and (b) illustrates an assembly system with multiple merge operations. Note that systems similar to that in (b) are commonly seen in automotive engine plants where the horizontal line represents the general engine assembly (with engine blocks as "raw materials"), while the vertical lines are various departments producing engine parts, such as crank shaft, camshaft, etc.

- **Closed Lines with respect to Carriers**

Production lines in manufacturing environment sometimes have parts transported from one operation to another on carriers (referred to as pallets, skids, etc.). In such systems, parts are loaded on and attached to the pallets at the first machine to undergo all the operations, upon completion of the operations, the finished parts are unloaded and the pallets are released and sent back to the first machine through an empty carrier buffer (see Figure 2.3). Since in this situation the number of parts in the system is bounded by the number of available carriers, additional starvation/blockage may be incurred due to empty/full carrier return buffer b_0. These lines are called closed with respect to carriers (or just closed).

2.1.2 Machine models

- **Timing Issues**

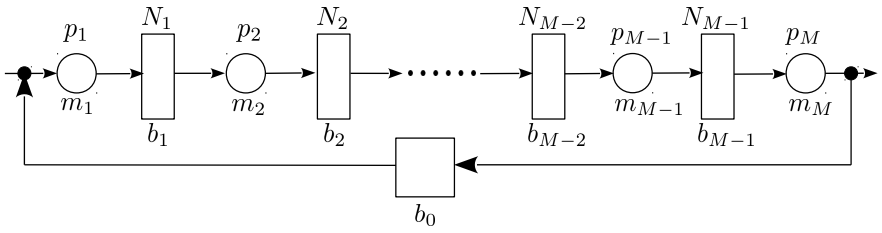


Figure 2.3: Closed line with respect to carriers

Cycle Time (τ): the time necessary to process a part by a machine. The cycle time may be constant, variable, or random. In large volume production systems, τ is practically always constant or close to being constant. This is the case in most production systems of the automotive, electronics, appliance, and other industries [9].

- **Reliability Models**

As mentioned before, the reliability of the machines poses one of the main challenges in analyzing a production system. In this research, we assume that each machine in a production system has two basic status: *up* and *down*. In addition, assume that one machine can process materials only when it is in *up* status, and cannot when it is in *down* status (due to failures, tool replacement, periodic maintenance, etc). Moreover, we define machine *uptime* as a continuous time interval, during which a machine is in *up* status, while *downtime* as the continuous time interval during which the machine is *down*. Clearly, machine *uptime* and *downtime* can be modeled as either constants or random variables. In this research, the latter is adopted, which is closer to production practice.

Machine reliability model refers to the probability mass functions (pmf's) or the probability density functions (pdf's) of the machine's *uptime* and *downtime*. In the discrete time case, for example, *geometric reliability model* will be addressed. In this model, machine up- and downtime are assumed to be geometric random variables with parameters P (breakdown probability) and R (repair probability), respectively. It should be noted that, *Bernoulli reliability model* is a special case of *geometric reliability model* where the sum of the breakdown rate and repair rate is 1. In the continuous time case, the *exponential reliability model* is

one of the most important models. In this model, the *uptime* and *downtime* of the machine are assumed to be exponential random variables, with parameters λ (breakdown rate) and μ (repair rate). Note both geometrical and exponential reliability models are characterized by Markov chains due to constant breakdown and repair probabilities/rates. Clearly, production systems with machines having these models can be analyzed using Markovian-based approaches.

In practice, however, the up- and downtime of machines may follow non-Markovian or even arbitrary distributions [9]. Therefore, in this research, we will also study systems with machines having *Rayleigh reliability model, Weibull reliability model, Gamma reliability model, log-normal reliability model*, etc. These models allow modeling of cases where a machine has time-varying breakdown rates and repair rates. In all the models discussed above, the mean and variance of their *uptime* and *downtime* in the machine models can be calculated based on their probability distributions. It has also been studied that the system performances are not sensitive to the specific machine models, but just to the moments (mean and variance) of its *uptime* and *downtime* probablity distributions [9]. In addition to those two moments, the coefficient of variation (CV) furthermore characterizes its level of "randomness". It is defined as the ratio of the standard variation and mean value. To determine the parameters of these models, note that long-tailed pmf's and pdf's have CV of the distribution greater than 1; for short-tailed pmf's, CV is smaller than 1. There is empirical evidence that in many production systems the coefficients of variation of *up*- and *downtime* of the machines on the factory floor are less than 1. Theoretically, it can be proved that machines with increasing (decreasing) breakdown and repair rates have $CV<1$ ($CV>1$). Since in most practical situations, the breakdown and repair rates are increasing in time, with accumulated depreciation of machinery and maintenance experiences, we assume that the distributions of the machine up- and downtimes must have the CVs less than 1 [68]. Among the above reliability, *exponential reliability model* and *Rayleigh reliability model* have fixed CVs (1 for exponential, $\sqrt{5}/2$ for Reyleigh). For *Weibull reliability model, gamma reliability*

model, and log-normal reliability model, the mean and variance of the random variables can be arbitrarily set by rationally selecting the parameters, and, as a result, the value of CV can be placed arbitrarily as well.

2.2 Approaches Used on the Addressed Problems

In this research, we study the problems of performance evaluation, bottleneck analysis, and production control in the modeling framework above. Each of them is detailed as follows.

2.2.1 Performance evaluation

In this research,

1. We develop mathematical models for transient analysis and derive analytical formulas to calculate the performance measures. For those systems that can be modeled as discrete-time-discrete-states (DTDS) or continuous-time-discrete-states (CTDS) Markovian models, analytical approaches are pursued. Specifically, system are characterized by Markov chains, the transition probabilities are derived according to the dynamic models. Then the performance measures are calculated based on the system states probability distributions. For those systems with non-Markovian machine reliability models, simulation approach will be used in future work.

2. For Markovian systems, closed-form formulas are derived for small-size cases (e.g., one- and two-machine lines). With the increase of the system size, however, the implementation of Markovian analysis is almost impossible due to the large number of system states. Thus, we also investigate computationally efficient algorithms for transient performance evaluation in larger scale of systems. Specifically, various decomposition-based approaches are explored. The idea is to decompose the original complex system into a group of virtual smaller lines (usually one-, two-, or three-machine lines), which are easier to analyze. Then, Markovian approach can be applied again. We derive

analytical methods to calculate the parameters of these virtual lines such that they can provide accurate approximation of the original system.

3. Finally, we study the structural properties of production systems during transients. Properties such as reversibility, monotonicity, effects of up- and downtimes, have been discussed in the framework of steady state operations [9, 69]. The results provide important insights to system design and improvement. While some initial efforts have been carried out for transient studies, but the results are quite limited [42]. In this research, studies on these system-theoretic properties are carried out based on the performance evaluation methods to be developed above.

2.2.2 Bottleneck analysis

In this research,

1. We first explore appropriate definitions of bottlenecks during production transient. Note that the performance index used in defining transient bottlenecks may be either a terminal one (e.g., completion time of a production run) or a temporal one (e.g., accumulated throughput within a time interval). In this research, we study bottlenecks with respect to both types of performance indices. In fact, the bottleneck during transients has not been rigorously studied in the existing research.

2. We derive mathematical methods to identify the transient bottlenecks of production systems. To accomplish this, the performance evaluation methods developed above are used. Specifically, for smaller-size Markovian systems, closed-form conditions are derived. These conditions are extended to larger-size Markovian systems with appropriate modifications. For non-Markovian systems, conjectures will be made based on the Markovian cases and justified by simulations in future work.

3. It should be noted that the partial derivatives involved in bottleneck definition(s) cannot be measured during actual production. Indeed, though many performance

metrics of a production system can be measured from real time data, their sensitivities to a specific parameter cannot be directly measured. As a result, an indirect method is necessary to carry out bottleneck identification based on real time factory floor data, and will be studied in future work.

2.2.3 Production control

In our research,

1. We investigate control policies designed to achieve different objectives for production systems and formulate mathematical models to represent these control policies in the modeling framework above. Specifically, switch-on/off operation control of the machines is modeled to include information such as buffer occupancy, machine status, etc.

2. We derive the mathematical models and analytical methods for evaluating transient performances of the systems with feedback production control. Both Markovian analysis-based approach and simulation are used.

3. Based on the mathematical model derived, we will investigate optimal control policies for different production objectives and compare the efficacy of each policy under different scenarios in future work. Finally, we will formulate practical production control rules that can be implemented by practitioners.

Chapter 3

FINITE PRODUCTION RUN-BASED TRANSIENT PERFORMANCE EVALUATION AND BOTTLENECK ANALYSIS OF SERIAL LINES WITH BERNOULLI MACHINES

3.1 Introduction

To meet the fast-changing market demands, modern manufacturing systems are often designed to be capable of producing a wide range of products. Typically, a certain number of identical or similar goods are grouped together based on customer orders or demand forecast to form a *production run*. Then, production runs are entered into the manufacturing systems to be completed. In many cases, process changeover and transitions (e.g., equipment setup/calibration/cleaning, fixture loading/unloading) are necessary between two consecutive production runs. As a result, while the work-in-process starts to accumulate at the beginning of a production run, the manufacturing system may operate in a transient regime that is qualitatively different from its steady state behavior. Similar transients may be observed towards the end of a production run, when the work-in-process is purged. It should be noted, though, for some manufacturing industries, the volume of a single produc-

tion run can be very large. In this case, the transient stage of such manufacturing processes is negligible compared to the entire production run, and the main portion of the production process can be viewed as being in the steady state. Clearly, this allows one to use the results on the steady state behavior of manufacturing systems, which are vastly available in the literature. On the other hand, there also exist numerous manufacturing processes, where each production run is relatively short. These production systems operate partially (or entirely) in the transient regime and the traditional steady state analysis may become inapplicable.

The research on production/service systems with finite population of jobs or customers can be roughly divided into three directions. The first and most notable one is the extensively studied area of *production scheduling*. The focus of this area is to develop effective and efficient algorithms to identify the sequence to process a group of different jobs at the machines so as to optimize certain performance measures (e.g., makespan, tardiness) of the overall system. Important results in this field are summarized in monographs [70, 71]. It should be pointed out, however, that the nature of the system operation considered in the production scheduling literature is different than the one addressed in this dissertation (e.g., unreliable machines, finite in-process buffers). Moreover, the scheduling research does not, in general, address transient behavior and performance evaluation of the production systems.

The second related research direction is concerned with *finite-source queueing systems*, which are also referred to as *finite-population queueing systems* in some literature. The studies in this area of research often involve a finite number of jobs (sources) circulating within a queueing system. Specifically, after a job exits from the (last) server, it returns to the population and spends a randomly distributed amount of time before entering the queueing system again. A great number of studies in this area employ such queueing systems to model the *machine interference problem* (or other similar resource allocation problems), where the jobs represent machine failures and the servers represent repair operators (see [72–74]). The focus of these studies is typically steady state performance evaluation and optimization under cyclic circulation of the jobs and does not address transient performance within a single cycle.

The third and also the closest body of research to this work assumes a group of finite number of jobs to be served/processed and discusses the system's transient and terminal performance of completing them. This is studied analytically in the framework of single-stage-multi-parallel-server Markovian queueing systems by [75–78]. Computer simulation is used by [79] to study $G/G/1$ queues. The results for multi-stage production systems are rather limited. Among them, a method that automatically generates the state space models of pull-controlled production systems is presented by [80]. The exact distribution of the number of parts produced in a given period of time, the distribution of the time to produce a given number of parts, and the distribution of the cycle time are also derived. It should be noted that, similar to all state-space-based methods, the computational efforts required by this method increases exponentially as a function of the number of stations and of buffer capacities. To combat the curse of dimensionality in such analyses, [81] derive computationally efficient analytical algorithms to approximate the transient performance and the moments of the production run completion time in serial production lines. An industrial case study is also described in the paper to illustrate the applicability of the methods developed. The results, however, are only applicable to systems with machines having the Bernoulli reliability model. For more complex machine models, [82] develop an analytical algorithm based on Markov reward model to evaluate the moments of accumulated production during transients and the completion time for Markovian production systems. On the other hand, the algorithm is only tractable for small-size systems due to its high computational requirement, which is even heavier than the one of the underlying Markov chain. This makes the algorithm impractical even for moderate-size systems. Therefore, contributing to this end is the goal of this Chapter.

3.2 Model and Performance Measures

3.2.1 Model

Consider a serial production line in Figure 3.1 defined by the following assumptions:

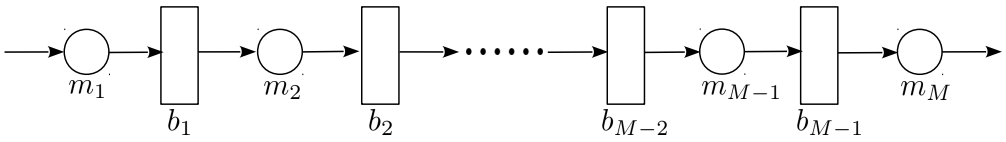


Figure 3.1: Serial production line

(i) The system consists of M machines (represented by circles) and $M-1$ buffers (represented by rectangles). The arrows indicate the direction of parts flow.

(ii) All machines have constant and identical cycle time τ. The time axis is slotted with duration τ.

(iii) Each in-process buffer, b_i, $i = 1, \ldots, M-1$, is characterized by its capacity, $0 < N_i < \infty$.

(iv) The machines obey the Bernoulli reliability model, i.e., machine m_i, $i = 1, \ldots, M$, when it is neither blocked nor starved, produces one part during a time slot with probability p_i and fails to do so with probability $1 - p_i$. Parameter p_i is referred to as the *efficiency* of m_i.

(v) Machine m_i, $i = 2, \ldots, M$, is starved during a time slot if it is up and buffer b_{i-1} is empty at the beginning of the time slot.

(vi) Machine m_i, $i = 1, \ldots, M-1$, is blocked during a time slot if it is up, buffer b_i has N_i parts at the beginning of the time slot and machine m_{i+1} fails to take a part during that time slot (either due to blockage or breakdown). It is assumed that m_M is never blocked.

(vii) The system operates on a finite production run-basis with run size equal to B: All buffers are initially empty and each machine stops operating as soon as it has finished processing B parts.

Remark 2.1: Note that in many production systems, machine cycle time is practically constant or close to being constant. This is the case in most production systems in automotive, electronics, appliance, and other industries. Note also, that the Bernoulli reliability model is applicable to operations where the downtime is, on the average, close to the machine cycle time (see [9, 83, 84] for practical examples using the Bernoulli model). Systems with machines having geometric reliability model are studied in next Chapter and machines having other reliability models (exponential, Weibull, gamma, log-normal, and general, etc.) will be studied in future work.

Remark 2.2: The above assumptions imply that the failures are time-dependent (i.e., a machine may break down during starvation or blockage). Another failure model alternative, operation-dependent failure (i.e., a machine cannot break down during starvation or blockage), is also used in the literature. The behavior and performance of systems defined by both conventions are very similar (see [85]). In this dissertation, we consider time-dependent failures, since this assumption was used in our previous work, which will serve as foundational material to the current research. Extension of the results to systems under operation-dependent failures will be carried out elsewhere.

Remark 2.3: Assumption (vi) implies the blocked-before-service (BBS) convention, under which, a machine may be starved and blocked during the same time slot. Its counterpart, the blocked-after-service (BAS) convention, is also widely used in production systems research (see [10, 86, 87]). While the BBS convention usually leads to a simpler description, the approaches to analyzing systems defined by both conventions are very similar. In this dissertation, we use the BBS convention, since it was used in the our previous studies, which will serve as foundational results to the current work. Extension to systems defined by BBS convention will be pursued elsewhere.

3.2.2 Performance measures

In the framework of the model defined, the performance measures of interest are:

- Production rate, $PR(n)$: the expected number of parts produced by m_M during time slot $n+1$;
- Consumption rate, $CR(n)$: the expected number of parts consumed by m_1 during time slot $n+1$;
- Work-in-process, $WIP_i(n)$: the expected number of parts in buffer b_i, $i = 1, \ldots, M-1$, at the beginning time slot $n+1$;
- Machine starvation $ST_i(n)$: the probability that machine m_i, $i = 2, \ldots, M$, is starved during time slot $n+1$;
- Machine blockage $BL_i(n)$: the probability that machine m_i, $i = 1, \ldots, M-1$, is blocked during time slot $n+1$.

In addition, let CT_i denote the time instant for machine m_i to complete producing all B parts. Clearly, CT_i is a discrete random variable. We denote its probability mass function as:

$$P_{ct_i}(n) = P[CT_i = n], \tag{3.1}$$

and its average and standard deviation as:

$$E(CT_i) = \sum_{n=1}^{\infty} nP_{ct_i}(n), \tag{3.2}$$

$$\sigma(CT_i) = \sqrt{\sum_{n=1}^{\infty} [n - E(CT_i)]^2 P_{ct_i}(n)}. \tag{3.3}$$

It should be noted that, if the production run is sufficiently long, then the system operates mostly in the steady state. As a result, the production completion time may be estimated as B/PR_{ss}, where PR_{ss} denotes the steady state production rate. This method underestimates the average completion time, since it assumes that the production immediately reaches the steady state level as the production run starts. In fact, numerical experiments show that the underestimation is, on average, about 15% when B is about five times M; the error decreases to less than 10% when $B > 10M$, and to less than 5% when $B > 20M$. Clearly, there is a necessity to develop more accurate performance evaluation methods for relatively short production runs, and, thus, is carried out next.

3.3 Exact Performance Analysis: One- and Two-Machine Lines

3.3.1 One-machine lines

Machine with constant efficiency

When the system consists of only one machine (see Figure 3.2(a)), it is characterized by a homogeneous Markov chain with the state being the number of parts that has been completed. Let $f(n)$ denote the number of parts completed by the machine at the end of time slot n. Then, the transition probabilities among the states are given by:

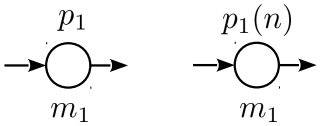


Figure 3.2: One-machine line

$$
\begin{aligned}
&P[f(n+1) = i+1 | f(n) = i] = p_1, \quad i = 0, 1, ..., B-1, \\
&P[f(n+1) = i | f(n) = i] = 1 - p_1, \quad i = 0, 1, ..., B-1, \\
&P[f(n+1) = B | f(n) = B] = 1.
\end{aligned} \tag{3.4}
$$

Next, let $\mathbf{x}_f(n) = [x_{f,0}(n)\ x_{f,1}(n)\ ...\ x_{f,B}(n)]^T$ with $x_{f,i}(n) = P[f(n) = i]$. Using the transition probabilities described in (3.4), the evolution of $\mathbf{x}_f(n)$ is characterized by

$$
\mathbf{x}_f(n+1) = \mathbf{A}_f \mathbf{x}_f(n), \quad \sum_{i=0}^{B} x_{f,i}(n) = 1, \tag{3.5}
$$

where

$$
\mathbf{A}_f = \begin{bmatrix}
1-p_1 & & & & \\
p_1 & 1-p_1 & & & \\
& p_1 & \ddots & & \\
& & \ddots & 1-p_1 & \\
& & & p_1 & 1
\end{bmatrix}. \tag{3.6}
$$

According to assumption (vii), the initial condition of the Markov chain is

$$
x_{f,i}(0) = \begin{cases}
1, & \text{if } i = 0, \\
0, & \text{otherwise.}
\end{cases} \tag{3.7}
$$

Since the system has only one machine, $WIP(n)$ does not exist, $ST(n) = BL(n) = 0$ for all n, and

$$
\begin{aligned}
&PR(n) = CR(n) = [p_1 \mathbf{J}_B \quad 0]\mathbf{x}_f(n), \\
&P_{ct_1}(n) = p_1 x_{f,B-1}(n-1) = [0\ \cdots\ 0\ p_1\ 0]\mathbf{x}_f(n-1),
\end{aligned} \tag{3.8}
$$

where $\mathbf{J}_k$ represents the 1-by-k matrix of ones.

In addition, it can be shown that the production completion time CT_1 follows a negative

binomial distribution defined by B and p_1:

$$P_{ct_1}(n) = P[CT_1 = n] = \binom{n-1}{B-1} p_1^B (1-p_1)^{n-B}, \tag{3.9}$$

and

$$E(CT_1) = \frac{B}{p_1}, \quad \sigma(CT_1) = \sqrt{B\frac{1-p_1}{p_1^2}}. \tag{3.10}$$

Machine with time-varying efficiency

Now, assume that the efficiency of the machine is time-varying: its efficiency during the n-th time slot is $p_1(n)$ (see Figure 3.2(b)). Then, the Markov chain becomes time-inhomogeneous and the evolution of system state $\mathbf{x}_f(n)$ is now given by:

$$\begin{aligned} &\mathbf{x}_f(n+1) = \mathbf{A}_f(n)\mathbf{x}_f(n), \quad \sum_{i=0}^{B} x_{f,i}(n) = 1, \\ &x_{f,i}(0) = \begin{cases} 1, & \text{if } i = 0, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{3.11}$$

where $\mathbf{A}_f(n)$ is obtained by replacing p_1 in $\mathbf{A}_f$ with $p_1(n)$. In addition,

$$PR(n) = CR(n) = [p_1(n)\mathbf{J}_B \quad 0]\mathbf{x}_f(n), \tag{3.12}$$

$$P_{ct_1}(n) = [0 \ \cdots \ 0 \ p_1(n) \ 0]\mathbf{x}_f(n-1). \tag{3.13}$$

Therefore, all performance measures, including the probability distribution of the completion time, can be calculated by iteratively evaluating the equations provided above.

Note that the results of one-machine line with time-varying machine efficiency will be used later in Section 3.4 as a building block to study multi-machine lines.

3.3.2 Two-machine lines

Consider a two-machine line defined by assumptions (i)-(vii) (see Figure 3.3). Let $f_i(n)$

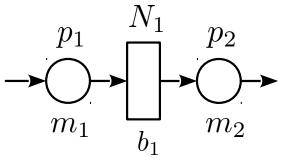


Figure 3.3: Two-machine line

denote the total number of parts that m_i has produced at the end of time slot n and $h(n)$ denote the number of parts in the buffer at the end of time slot n. It follows immediately that

$$f_1(n) - f_2(n) = h(n).$$

Clearly, the system can be characterized by a Markov chain with state defined by either $(h(n), f_1(n))$ or $(h(n), f_2(n))$. Without loss of generality, in this Chapter, we use $(h(n), f_2(n))$ as the system state. The total number of system states is

$$S = \begin{cases} \frac{(2B-N_1+2)(N_1+1)}{2}, & B > N_1, \\ \\ \frac{(B+2)(B+1)}{2}, & B \leq N_1. \end{cases} \tag{3.14}$$

Based on the system description given in Subsection 3.2.1, the transition probabilities of the system can be obtained in (3.15).

$$\begin{aligned} &P[h(n+1) = 0, f_2(n+1) = i+1 | h(n) = 0, f_2(n) = i] = 1 - p_1, \quad i = 0, \ldots, B, \\ &P[h(n+1) = 0, f_2(n+1) = i | h(n) = 1, f_2(n) = i] = p_1, \quad i = 0, \ldots, B, \\ &P[h(n+1) = i, f_2(n+1) = j | h(n) = i, f_2(n) = j] = (1-p_1)(1-p_2), \\ &\qquad\qquad i = 0, \ldots, B, \quad j = 0, \ldots, B-i-1, \end{aligned}$$

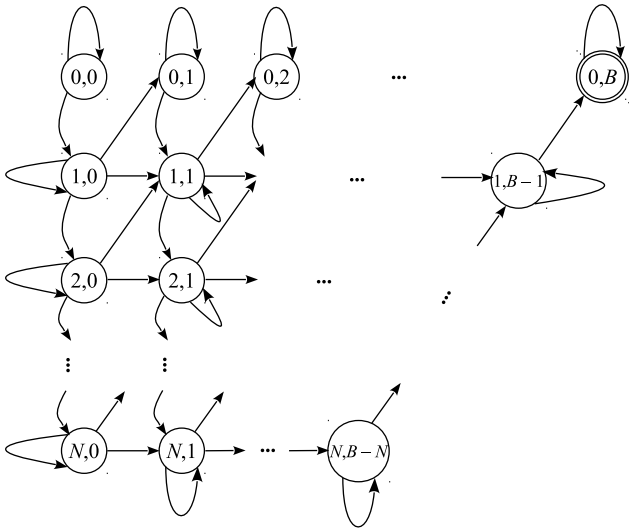
$$
\begin{aligned}
&P[h(n+1)=i, f_2(n+1)=j+1 | h(n)=i, f_2(n)=j] = p_1 p_2, \\
&\qquad\qquad i=0,\ldots,B, \quad j=0,\ldots,B-i-1, \\
&P[h(n+1)=i+1, f_2(n+1)=j | h(n)=i, f_2(n)=j] = p_1(1-p_2), \\
&\qquad\qquad i=0,\ldots,B, \quad j=0,\ldots,B-i-1, \\
&P[h(n+1)=i-1, f_2(n+1)=j+1 | h(n)=i, f_2(n)=j] = (1-p_1)p_2, \\
&\qquad\qquad i=0,\ldots,B, \quad j=0,\ldots,B-i-1, \\
&P[h(n+1)=i, f_2(n+1)=B-i | h(n)=i, f_2(n)=B-i] = 1-p_2, \\
&\qquad\qquad i=0,\ldots,B, \\
&P[h(n+1)=i-1, f_2(n+1)=B-i+1 | h(n)=i, f_2(n)=B-i] = p_2, \quad i=0,\ldots,B, \\
&P[h(n+1)=0, f_2(n+1)=B | h(n)=0, f_2(n)=B] = 1.
\end{aligned}
\tag{3.15}
$$

In addition, if $B > N_1$, additional transition probabilities given in (3.16) are needed.

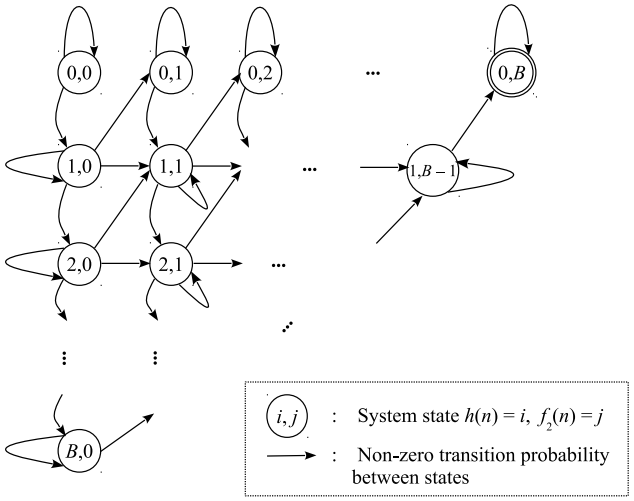
$$
\begin{aligned}
&P[h(n+1)=N_1, f_2(n+1)=j | h(n)=N_1, f_2(n)=j] = 1-p_2, \quad j=0,\ldots,B-N_1-1, \\
&P[h(n+1)=N_1, f_2(n+1)=j+1 | h(n)=N_1, f_2(n)=j] = p_1 p_2, \quad j=0,\ldots,B-N_1-1, \\
&P[h(n+1)=N_1-1, f_2(n+1)=j+1 | h(n)=N_1, f_2(n)=j] = (1-p_1)p_2, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad j=0,\ldots,B-N_1-1.
\end{aligned}
\tag{3.16}
$$

The transition probabilities other than the ones given in (3.15) and (3.16) are zeros. Clearly, the system is defined by an absorbing Markov chain with one absorbing state: $(0, B)$. The state transition diagrams for the Markov chains are given in Figure 3.4.

To study this system, we *linearize* the state space of the system as follows: for system



(a) $B > N_1$



(b) $B \leq N_1$

Figure 3.4: State transition diagrams for two-machine lines

state $(h = i, f_2 = j)$, define

$$k = \begin{cases} \sum_{l=0}^{j-1} (B + 1 - j) + (i + 1), & B \leq N_1, \\ \\ \sum_{l=0}^{j-1} [\min (N_1 + 1, B - l + 2)] + (i + 1), & B > N_1. \end{cases} \tag{3.17}$$

Thus, system state (h, f_2) becomes state k of the Markov chain. Clearly, state 1 refers to system state $(0, 0)$, which is the initial state of the system, while the last state refers to system state $(0, B)$, which is also the absorbing state of the system. Under this arrangement, given $k \in \{1, \ldots, S\}$, the corresponding system state can also be found. For convenience, we denote it as $(h[k], f_2[k])$.

Let $\mathbf{A}_{h,f}$ denote the transition probability matrix of the Markov chain, and let column vector $\mathbf{x}(n) = [x_1(n)\ x_2(n)\ \cdots x_S(n)]^T$ denote the probability distribution of the Markov chain at time n. Then, the system evolution is given by

$$\begin{aligned} &\mathbf{x}(n+1) = \mathbf{A}_{h,f}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \\ &x_i(0) = \begin{cases} 1, & \text{if } i = 1, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{3.18}$$

Then, the transient performance measures of the system can be calculated as

$$\begin{aligned} PR(n) &= P[m_2 \text{ up, } b_1 \text{ not empty, and production not completed on } m_2 \text{ in time slot } n] \\ &= p_2 P[h(n) > 0,\ f_2(n) < B] \\ &= \mathbf{C}_1\mathbf{x}(n), \end{aligned} \tag{3.19}$$

$$\begin{aligned} CR(n) &= P[m_1 \text{ up, not blocked and production not completed on } m_1 \text{ in time slot } n] \\ &= p_1 P[h(n) < N_1, f_1(n) < B] + p_1 p_2 P[h(n) = N_1, f_1(n) < B] \\ &= \mathbf{C}_2\mathbf{x}(n), \end{aligned} \tag{3.20}$$

$$
\begin{aligned}
WIP(n) &= \sum_{i=1}^{N_1} iP[h(n)=i] = \mathbf{C}_3\mathbf{x}(n), && (3.21)\\
BL_1(n) &= P[m_1 \text{ up, } b_1 \text{ full, } m_2 \text{ down in time slot } n] \\
&= p_1(1-p_2)P[h(n)=N_1] \\
&= \mathbf{C}_4\mathbf{x}(n), && (3.22)\\
ST_2(n) &= P[m_2 \text{ up and } b_1 \text{ empty during time slot } n] \\
&= p_2P[h(n)=0] \\
&= \mathbf{C}_5\mathbf{x}(n), && (3.23)\\
P_{ct_1}(n) &= P[m_1 \text{ up and not blocked during time slot } n \text{ and } f_1(n)=B-1] \\
&= p_1P[h(n)<N_1, f_1(n)=B-1] + p_1p_2P[h(n)=N_1, f_1(n)=B-1] \\
&= \mathbf{C}_6\mathbf{x}(n), && (3.24)\\
P_{ct_2}(n) &= P[m_2 \text{ up and not starved during time slot } n \text{ and } f_2(n)=B-1] \\
&= p_2P[h(n)=1, f_2(n)=B-1] \\
&= \mathbf{C}_7\mathbf{x}(n), && (3.25)
\end{aligned}
$$

where

$$
\begin{aligned}
\mathbf{C}_1 &= [c_{1,1}\ c_{1,2}\ \cdots\ c_{1,S}], \quad \mathbf{C}_2 = [c_{2,1}\ c_{2,2}\ \cdots\ c_{2,S}], \\
\mathbf{C}_3 &= [c_{3,1}\ c_{3,2}\ \cdots\ c_{3,S}], \quad \mathbf{C}_4 = [c_{4,1}\ c_{4,2}\ \cdots\ c_{4,S}], \\
\mathbf{C}_5 &= [c_{5,1}\ c_{5,2}\ \cdots\ c_{5,S}], \quad \mathbf{C}_6 = [c_{6,1}\ c_{6,2}\ \cdots\ c_{6,S}], \\
\mathbf{C}_7 &= [c_{7,1}\ c_{7,2}\ \cdots\ c_{7,S}],
\end{aligned} \qquad (3.26)
$$

and

$$
c_{1,k} = \begin{cases} p_2, & \text{if } h[k]>0 \text{ and } k<S, \\ 0, & \text{otherwise,} \end{cases}
$$

$$
\begin{aligned}
c_{2,k} &= \begin{cases} p_1, & \text{if } h[k] < N_1 \text{ and } h[k] + f_2[k] < B, \\ p_1 p_2, & \text{if } h[k] = N_1 \text{ and } h[k] + f_2[k] < B, \\ 0, & \text{otherwise,} \end{cases} \\
c_{3,k} &= h[k], \\
c_{4,k} &= \begin{cases} p_1(1-p_2), & \text{if } h[k] = N_1 \text{ and } h[k] + f_2[k] < B, \\ 0, & \text{otherwise,} \end{cases} \\
c_{5,k} &= \begin{cases} p_2, & \text{if } h[k] = 0 \text{ and } h[k] + f_2[k] < B, \\ 0, & \text{otherwise,} \end{cases} \\
c_{6,k} &= \begin{cases} p_1, & \text{if } h[k] + f_2[k] = B - 1 \text{ and } h[k] \neq N_1, \\ p_1 p_2, & \text{if } h[k] + f_2[k] = B - 1 \text{ and } h[k] = N_1, \\ 0, & \text{otherwise,} \end{cases} \\
c_{7,k} &= \begin{cases} p_2, & \text{if } f_2[k] = B - 1 \text{ and } h[k] = 1, \\ 0, & \text{otherwise.} \end{cases}
\end{aligned}
\tag{3.27}
$$

It should be noted that, while $E(CT_i)$ and $\sigma(CT_i)$ can be calculated based on (3.9), (3.10) and (3.24), (3.25), they can also be calculated using the properties of absorbing Markov chains [88]. For instance, to calculate $E(CT_2)$ and $\sigma(CT_2)$, state $(0, B)$ is viewed as the absorbing state of the Markov chain. Then, $\mathbf{A}_{h,f}$ can be expressed as:

$$
\mathbf{A}_{h,f} = \begin{bmatrix} \mathbf{Q} & \mathbf{0} \\ \mathbf{R} & 1 \end{bmatrix}, \tag{3.28}
$$

where $\mathbf{Q}$ is an $(S-1)$-by-$(S-1)$ matrix, $\mathbf{R}$ is a nonzero 1-by-$(S-1)$ matrix, $\mathbf{0}$ is an $(S-1)$-by-1 zero matrix. Note that the production completion time at machine m_2 is, in fact, the time for the Markov chain to be absorbed. To calculate the mean and variance of

the absorbing time of the Markov chain, we first calculate its fundamental matrix:

$$\mathbf{F} = (\mathbf{I} - \mathbf{Q})^{-1}. \tag{3.29}$$

Then, the expected time before being absorbed when starting in transient state i is the ith entry of the vector

$$\mathbf{t} = \mathbf{1} \cdot \mathbf{F}, \tag{3.30}$$

where $\mathbf{1}$ is a row vector with all entries equal to 1. In addition, the variance on the amount of time before being absorbed when starting in state i is the ith entry of the vector

$$\mathbf{v} = \mathbf{t}(2\mathbf{F} - \mathbf{I}) - \mathbf{t}_{sq}, \tag{3.31}$$

where $\mathbf{t}_{sq}$ represents the Hadamard product of $\mathbf{t}$ and itself.

Since the system starts operating from empty buffer, the average and standard deviation of completion time CT_2 are given by the first entry of $\mathbf{t}$ and the square root of the first entry of $\mathbf{v}$, respectively, i.e.,

$$E(CT_2) = \mathbf{t}_1, \quad \sigma(CT_2) = \sqrt{\mathbf{v}_1}. \tag{3.32}$$

Similarly, $E(CT_1)$ and $\sigma(CT_1)$ can be calculated using the same technique by letting states $(h, B-h)$, $h = 0, \ldots, \min(B, N_1)$, be absorbing.

An illustration of a two-machine Bernoulli line is given in Figure 3.5. Clearly, all system measures are entirely in transient. The average completion time at m_2 is $CT_2 = 29.5$ (indicated by the vertical line in the figure).

Finally, it should be noted that the method developed above can be applied to systems with machines having time-varying machine parameters by replacing p_i with $p_i(n)$ in the formulas.

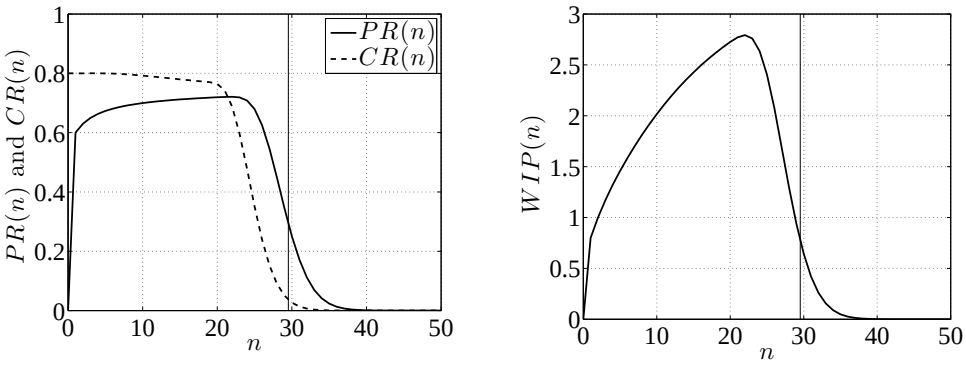


Figure 3.5: Transient performance for $p_1 = 0.8$, $p_2 = 0.75$, $N_1 = 5$ and $B = 20$

3.4 Approximate Performance Analysis: Multi-Machine Lines

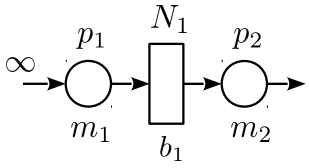
Although the production system defined by assumptions (i)-(vii) is characterized by an ergodic Markov chain regardless of the number of machines in the systems, the number of states of the Markov chain increases rapidly with M, N_i's, and B. For instance, a five-machine line with all buffers having capacity 4 working on a production run of 50 parts has a total of 31875 states. If the number of machines grows to 10 with same buffer capacities and production run size, then the Markov chain will have a total of 99609375 states. Clealy, the direct Markov analysis approach may not be practical for larger systems. In this section, a computationally efficient approach based on aggregation is developed to approximate the system performance measures with high accuracy.

3.4.1 Two-machine lines

Construction of auxiliary lines

As one can see from Subsection 3.3.2, the joint consideration of buffer status and production completion status is one of the main complicating factors of the Markovian analysis for a two-machine line. Therefore, in this subsection, we propose a method to "decompose"

$h(n)$ and $f_i(n)$ by constructing an auxiliary two-machine line and two auxiliary one-machine lines to approximate the production process in the original system. Specifically, the auxiliary two-machine line contains the same machines and buffers as the original system but with production run of infinite number of parts (see Figure 3.6(a)). The two auxiliary one-machine lines both have machines with time-varying efficiency, denoted as $\widehat{p}_1(n)$ and $\widehat{p}_2(n)$ (see Figure 3.6(b) and (c)).



(a) Auxiliary two-machine line

(b) Auxiliary one-machine line 1 (c) Auxiliary one-machine line 2

Figure 3.6: Auxiliary lines for two-machine line aggregation analysis

The method for analyzing the auxiliary two-machine line has been developed in [39]. On the other hand, once $\widehat{p}_1(n)$ and $\widehat{p}_2(n)$ are known, the method described in Subsubsection 3.3.1 can be used to analyze the auxiliary one-machine lines. Next, we will propose a method to calculate the values of $\widehat{p}_1(n)$ and $\widehat{p}_2(n)$ using the auxiliary two-machine line and, based on all three auxiliary lines, approximate the performance measures of the original system.

Let $\widehat{x}_{h,i}(n)$ denote the probability that the buffer of the two-machine auxiliary line has i parts at the end of time slot n and $\widehat{\mathbf{x}}_h(n) = [\widehat{x}_{h,0}(n)\ \widehat{x}_{h,1}(n)\ ...\ \widehat{x}_{h,N_1}(n)]^T$. According to [39],

$$\mathbf{A}_2 = \begin{bmatrix} 1-p_1 & p_2(1-p_1) & 0 & \cdots & 0 \\ p_1 & 1-p_1-p_2+2p_1p_2 & p_2(1-p_1) & \cdots & 0 \\ 0 & p_1(1-p_2) & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1-p_1-p_2+2p_1p_2 & p_2(1-p_1) \\ 0 & 0 & \cdots & p_1(1-p_2) & p_1p_2+1-p_2 \end{bmatrix}. \tag{3.34}$$

the evolution of $\widehat{\mathbf{x}}_h$ is given by:

$$\begin{aligned} &\widehat{\mathbf{x}}_h(n+1) = \mathbf{A}_2(n)\widehat{\mathbf{x}}_h(n), \quad \sum_{i=0}^{N_1} x_{h,i}(n) = 1, \\ &\widehat{x}_{h,i}(0) = \begin{cases} 1, & \text{if } i = 0, \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \tag{3.33}$$

where $\mathbf{A}_2$ is defined in (3.34).

Since m_1 is capable of producing one part if and only if it is up and not blocked, while m_2 is capable of producing if and only if it is up and not starved, we define time-varying efficiencies of the auxiliary one-machine lines as follows:

$$\begin{aligned} \widehat{p}_1(n) &= p_1[1 - \widehat{x}_{h,N_1}(n-1)(1-p_2)], \\ \widehat{p}_2(n) &= p_2[1 - \widehat{x}_{h,0}(n-1)]. \end{aligned} \tag{3.35}$$

In other words, the machine ($\widehat{m}_1$) in auxiliary one-machine line 1 is up if and only if the original machine m_1 is up and the first machine in the auxiliary two-machine line is not blocked during the same time slot. Similarly, the machine ($\widehat{m}_2$) in auxiliary one-machine line 2 is up if and only if the original machine m_2 is up and the second machine in the auxiliary two-machine line is not starved. Now, let $\widehat{x}_{f,j}^{(i)}(n)$ denote the probability that the machine in auxiliary one-machine line i, $i = 1, 2$, completes the j-th part of the production run at the end of time slot n and $\widehat{\mathbf{x}}_f^{(i)}(n) = [\widehat{x}_{f,0}^{(i)}(n)\ \widehat{x}_{f,1}^{(i)}(n)\ ...\ \widehat{x}_{f,B}^{(i)}(n)]^T$. According to

Subsubsection 3.3.1, the evolution of $\widehat{\mathbf{x}}_f^{(i)}(n)$ is given by:

$$\begin{aligned}
&\widehat{\mathbf{x}}_f^{(i)}(n+1) = \widehat{\mathbf{A}}_f^{(i)}(n)\widehat{\mathbf{x}}_f^{(i)}(n), \quad \sum_{j=0}^{B} \widehat{x}_{f,j}^{(i)}(n) = 1, \\
&\widehat{x}_{f,j}^{(i)}(0) = \begin{cases} 1, & \text{if } j = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1, 2,
\end{aligned} \tag{3.36}$$

where transition probability matrix $\widehat{\mathbf{A}}_f^{(i)}(n)$ can be obtained by replacing all p_1's in (3.6) with $\widehat{p}_i(n)$.

In addition, it should be noted that, this approach is implemented based on three smaller Markov chains, one with N_1+1 states and two with $B+1$ states, compared to a bigger Markov chain with S states used in the method described in Subsection III-B (see equation (3.14)). This reduction in state space dimension has the potential to reduce the computational efforts required by the methods.

Formulas for performance measure approximation

Based on the auxiliary lines constructed above, we propose the formulas to approximate the original system performance measures below. First, the production completion times at the machines in the original system are approximated by the production completion times of the two auxiliary one-machine lines. Let $\widehat{P}_{ct_i}(n)$ denote the approximated probability that m_i in the original system completes the production run at the end of time n. Then,

$$\widehat{P}_{ct_i}(n) = \widehat{p}_i(n)\widehat{x}_{f,B-1}^{(i)}(n-1). \tag{3.37}$$

Next, the original system consumption rate is approximated by the production rate of auxiliary one-machine line 1, while the original system's production rate is approximated by

auxiliary one-machine line 2:

$$\widehat{PR}(n) = [\widehat{p}_2(n)\mathbf{J}_B \quad 0]\widehat{\mathbf{x}}_f^{(2)}(n-1), \tag{3.38}$$

$$\widehat{CR}(n) = [\widehat{p}_1(n)\mathbf{J}_B \quad 0]\widehat{\mathbf{x}}_f^{(1)}(n-1). \tag{3.39}$$

To approximate $WIP(n)$, $BL_1(n)$ and $ST_2(n)$, two auxiliary lines need to be combined. Specifically, these performance measures are approximated using their counterparts in the auxiliary two-machine lines and "discounted" by the probability that the production run is completed in the auxiliary one-machine lines:

$$\begin{aligned}
\widehat{WIP}(n) &= [0\ 1\ ...\ N]\widehat{\mathbf{x}}_h(n)(1-\widehat{x}_{f,B}^{(2)}(n-1)),\\
\widehat{ST_2}(n) &= [p_2\ 0\mathbf{J}_N]\widehat{\mathbf{x}}_h(n-1)(1-\widehat{x}_{f,B}^{(1)}(n-1)),\\
\widehat{BL_1}(n) &= [0\mathbf{J}_N\ p_1(1-p_2)]\widehat{\mathbf{x}}_h(n-1)(1-\widehat{x}_{f,B}^{(1)}(n-1)).
\end{aligned} \tag{3.40}$$

Finally, the mean and standard deviation of the completion times are approximated by:

$$\begin{aligned}
E(\widehat{CT}_i) &= \sum_{n=1}^{\infty} n\widehat{P}_{ct_i}(n), \qquad\qquad i = 1, 2,\\
\sigma(\widehat{CT}_i) &= \sqrt{\sum_{n=1}^{\infty}[n - E(\widehat{CT}_i)]^2\widehat{P}_{ct_i}(n)}, \quad i = 1, 2.
\end{aligned} \tag{3.41}$$

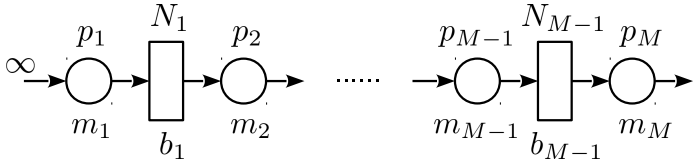
The accuracy of these performance approximations will be discussed in Subsection 3.4.2, after the performance approximation formulas for $M > 2$-machine lines are introduced in the next subsection.

3.4.2 $M > 2$-machine lines

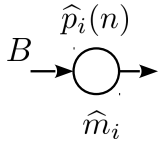
Construction of auxiliary lines

The approach of analyzing $M > 2$-machine lines is similar to the two-machine case. First, an auxiliary M-machine line is introduced with the same machines and buffers in the original

system but with infinite production run size (see Figure 3.7(a)). Then, a total of M auxiliary one-machine lines are introduced, each with a machine having time-varying efficiency, $\widehat{p}_i(n)$, $i = 1, \ldots, M$, and finite production run size equal to B (see Figure 3.7(b)).



(a) Auxiliary M-machine line



(b) Auxiliary one-machine line i

Figure 3.7: Auxiliary lines for performance approximation of M-machine line

Transient behavior of multi-machine Bernoulli serial lines with infinite production run size has been studied in [42], which develops an iterative procedure based on recursive aggregation to approximate the system's transient performance. Based on this work, an algorithm is proposed in [89] to further improve the computational efficiency of the original method in [42]. The idea is to represent the parts flow of the serial line by a set of two-machine lines with machines having time-varying efficiencies (see Figure 3.8).

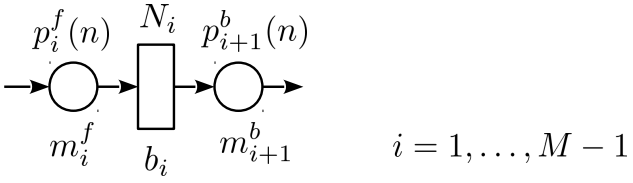


Figure 3.8: Two-machine line representation for transient analysis of M-machine Bernoulli line

Here, $p_i^f(n)$ is used to approximate the aggregated parts producing capability into buffer b_i from all its upstream machines and buffers. Similarly, $p_{i+1}^b(n)$ is used to approximate the aggregated parts drawing capability out of buffer b_i from all its downstream machines and buffers. Using the auxiliary M-machine line and its two-machine line representations, we propose an algorithm below to calculate the parameters of the auxiliary one-machine

systems. This procedure along with the calculation of $p_i^b(n)$ and $p_i^f(n)$ is summarized as follows:

Calculation Procedure 1:

Formulas for performance approximation

Using similar idea as the two-machine case, the following performance approximation formulas are defined − each approximates the corresponding performance measure without the " $\widehat{\ }$ ":

$$
\begin{aligned}
&\widehat{PR}(n) = [\widehat{p}_M(n)\mathbf{J}_B \quad 0]\widehat{\mathbf{x}}_f^{(M)}(n-1),\\
&\widehat{CR}(n) = [\widehat{p}_1(n)\mathbf{J}_B \quad 0]\widehat{\mathbf{x}}_f^{(1)}(n-1),\\
&\widehat{WIP_i}(n) = [0\ 1\ ...\ N_i]\widehat{\mathbf{x}}_h^{(i)}(n)(1-\widehat{x}_{f,B}^{(i+1)}(n-1)), \qquad i=1,\ldots,M-1,\\
&\widehat{ST_i}(n) = [p_i \quad 0\mathbf{J}_{N_{i-1}}]\widehat{\mathbf{x}}_h^{(i-1)}(n-1)(1-\widehat{x}_{f,B}^{(i-1)}(n-1)), \qquad i=2,\ldots,M,\\
&\widehat{BL_i}(n) = [0\mathbf{J}_{N_i} \quad p_i(1-p_{i+1}^b(n))]\widehat{\mathbf{x}}_h^{(i)}(n-1)\cdot(1-\widehat{x}_{f,B}^{(i)}(n-1)), \qquad i=1,\ldots,M-1,\\
&\widehat{P}_{ct_i}(n) = \widehat{p}_i(n)\widehat{x}_{f,B-1}^{(i)}(n-1), \qquad i=1,\ldots,M.
\end{aligned}
\tag{3.42}
$$

In addition,

$$
\begin{aligned}
&E(\widehat{CT}_i) = \sum_{n=1}^{\infty} n\widehat{P}_{ct_i}(n), \quad i=1,\ldots,M,\\
&\sigma(\widehat{CT}_i) = \sqrt{\sum_{n=1}^{\infty}[n-E(\widehat{CT}_i)]^2\widehat{P}_{ct_i}(n)},\ i=1,\ldots,M.
\end{aligned}
\tag{3.43}
$$

3.4.3 Accuracy of the approximation methods

To investigate the accuracy of the performance approximation methods proposed above, numerical experiments were carried out. Specifically, we studied Bernoulli lines with the

number of machines belonging to the following set:

$$M \in \{2, 3, 5, 10, 15, 20\}.$$

Next, for each M, we generated 100,000 lines with system parameters randomly and equiprobably selected from

$$p_i \in (0.7, 1), \quad N_i \in \{1, 2, 3, 4, 5, 6\}, \quad B \in [5, 105].$$

Therefore, a total of 600,000 lines were studied. For each line, thus constructed, we calculated its performance measure approximations using (3.37)-(3.41) or (3.42)-(3.43). For comparison, a simulation program has been created using C++ to estimate the true values of the performance measures and we ran 10,000 replications of the simulation for each line generated above. This results 95% confidence intervals of less than 0.005 for $PR(n)$ and $CR(n)$; 0.05 for $WIP_i(n)$; 0.01 for $ST_i(n)$ and $BL_i(n)$; and 0.01 for CT_i. To quantitatively evaluate the accuracy of the transient performance measure approximations, we calculate the average approximation errors for each line based on:

$$\begin{aligned}
\delta_{PR} &= \frac{1}{T}\sum_{n=1}^{T}\frac{|\widehat{PR}(n) - PR_{sim}(n)|}{PR_{ss}} \cdot 100\%, \\
\delta_{CR} &= \frac{1}{T}\sum_{n=1}^{T}\frac{|\widehat{CR}(n) - CR_{sim}(n)|}{PR_{ss}} \cdot 100\%, \\
\delta_{WIP} &= \frac{\sum_{i=1}^{M-1}\sum_{n=1}^{T}\frac{|\widehat{WIP}_i(n) - WIP_i^{sim}(n)|}{N_i} \cdot 100\%}{(M-1)T}, \\
\delta_{ST} &= \frac{1}{(M-1)T}\sum_{i=2}^{M}\sum_{n=1}^{T}|\widehat{ST}_i(n) - ST_i^{sim}(n)|, \\
\delta_{BL} &= \frac{1}{(M-1)T}\sum_{i=1}^{M-1}\sum_{n=1}^{T}|\widehat{BL}_i(n) - BL_i^{sim}(n)|,
\end{aligned}$$

where PR_{ss} is obtained using the simulation code with 95% confidence interval of less than 0.001, and T is the smallest time instant such that inequality

$$\min\left\{\sum_{n=1}^{T}\widehat{P}_{ct,M}(n),\sum_{n=1}^{T}P_{ct,M}^{sim}(n)\right\}\geq 0.999$$

is observed for this line. The results are summarized in the box plots of Figure 3.9 for different values of machine number M. Note that in a box plot, the bottom and top of the box indicate the first and third quartiles, while the band inside the box indicates the median of the quantity studied. As we can see, the approximation errors of $PR(n)$, $CR(n)$, and $WIP(n)$ are typically below 3% with few outliers reaching up to 10% for small values of M. For $ST(n)$ and $BL(n)$, the average approximation errors are generally below 0.01. It should be noted that the outliers observed during the investigation are mostly from cases with very small production runs. The resulting short production completion times for these cases may "amplify" the larger errors in only a few time instants.

The approximation accuracy of average completion time is evaluated based on:

$$\delta_{E(CT_i)}=\frac{1}{M}\sum_{i=1}^{M}\frac{|E(\widehat{CT}_i)-E(CT_i^{sim})|}{E(CT_i^{sim})}\cdot 100\%.$$

The results are summarized in Figure 3.10(a). It can be observed from the box plots that the median approximation error for completion time is below 0.7%, while the error is below 2% in almost all cases considered. Moreover, it appears that the approximation error increases as the number of machines increases. Also, as illustrated in Figure 3.11, the approximation error tends to be larger when the production run size is either very small or very large. Finally, Figure 3.12 presents the average approximation error of the completion times at each machine for all the 20-machine lines studied above. The 20 boxes in the figure, from left to right, represent the 20 machines in the line from m_1 to m_{20}. As the figure suggests, the approximation of production completion time tends to be less accurate around the middle of the line and with marginally smaller errors towards both ends of the line. Considering

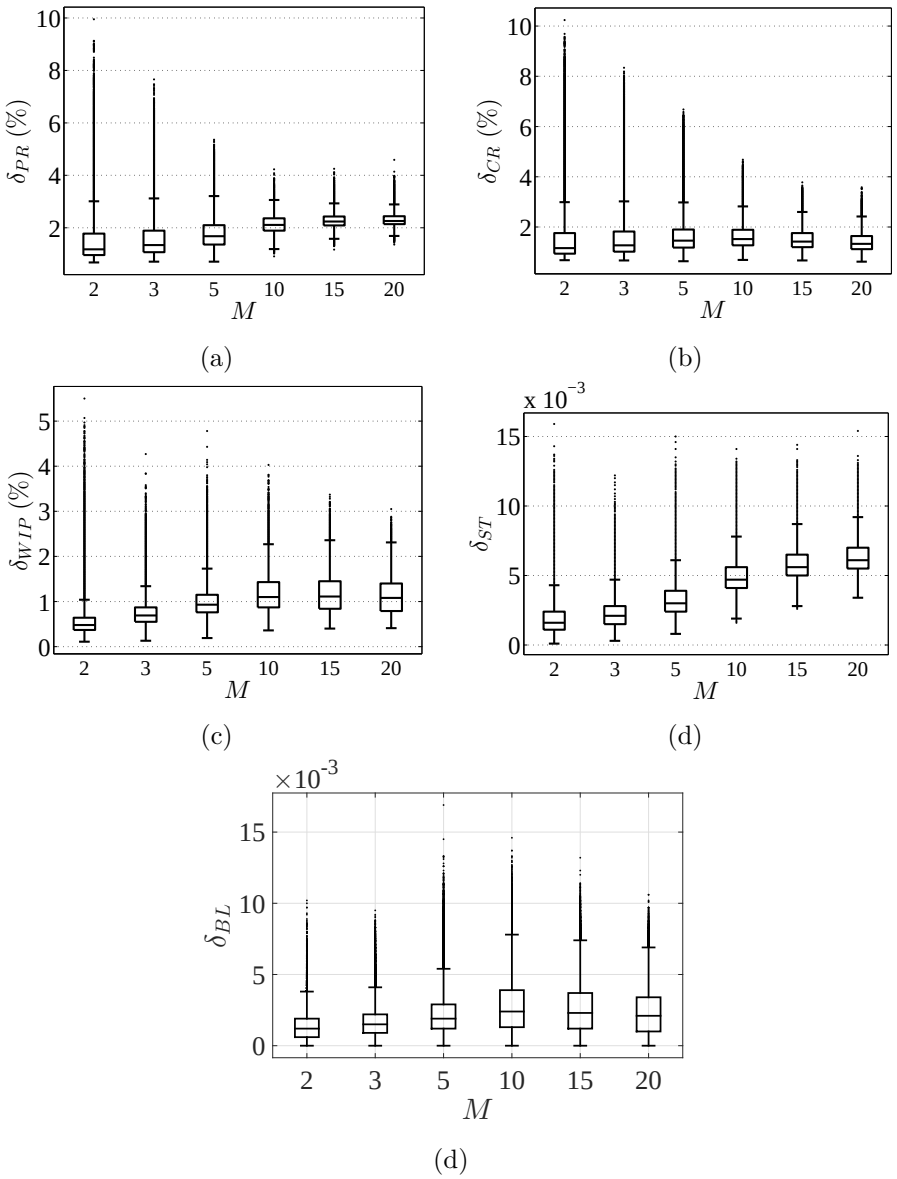


Figure 3.9: Accuracy of transient performance measure approximations

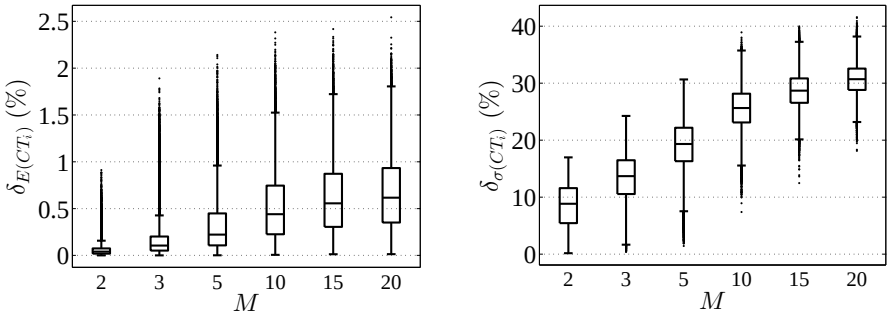


Figure 3.10: Accuracy of completion time approximation

the fact that the machine and buffer parameters of a production line are rarely known in practice with accuracy better than 5%-10%, we conclude that the aggregation-based method proposed in Subsections 3.4.1 and 3.4.2 can be used as effective tools to estimate the transient performance and average production completion time of Bernoulli serial lines with good accuracy. Compared with simulation, the calculation procedure is typically 7-10 times faster. For instance, for the 20-machine lines studied, the average computation time of Calculation Procedure 1 is about 0.07 seconds on a Macbook Pro with 2.3 GHz Intel Core i7 and 16 GB memory, while the average simulation time is about 0.6 seconds. One can easily imagine the saving of computational efforts when it comes to tasks such as searching optimal system parameters.

On the other hand, it should be pointed out that the approximation of the standard deviation of the completion time is not as accurate − the relative error, evaluated based on

$$\delta_{\sigma(CT_i)} = \frac{1}{M}\sum_{i=1}^{M}\frac{|\sigma(\widehat{CT}_i) - \sigma(CT_i^{sim})|}{\sigma(CT_i^{sim})} \cdot 100\%,$$

is typically 10-30% (see Figure 3.10(b)). However, the experiment data shows that the approximation overestimates the actual standard deviation in majority of cases (see Table 3.1). Therefore, $\sigma(\widehat{CT}_i)$ can be viewed as a conservative estimation of the variation in the system's production completion time.

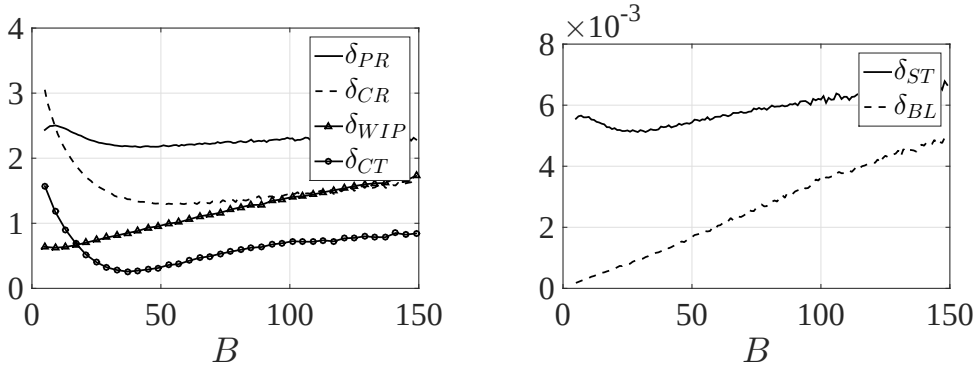


Figure 3.11: Accuracy of the approximation as a function of production run size for $M = 15$

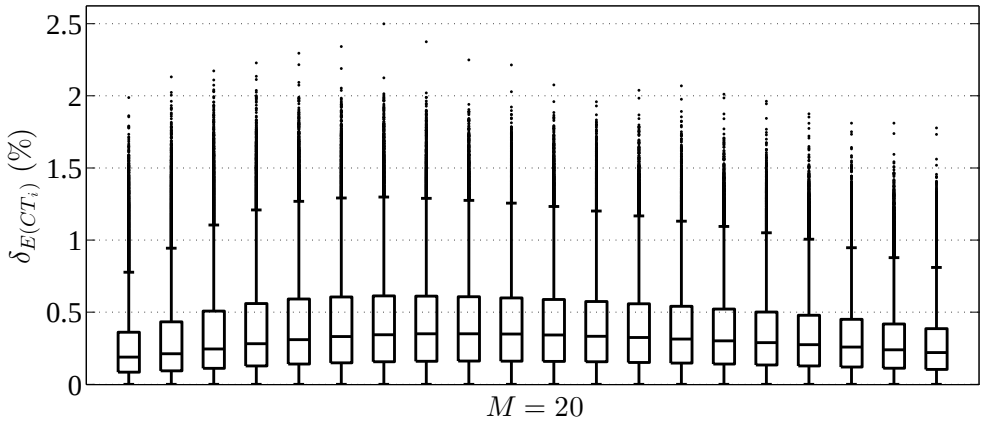


Figure 3.12: Accuracy of production completion time approximation as a function of machine position in 20-machine lines

Table 3.1: Proportion of cases where $\sigma(CT_i)$ is overestimated by $\sigma(\widehat{CT}_i)$

$M = 2$	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
96.38%	98.83%	99.61%	99.83%	99.89%	99.92%

As an illustration, consider the five-machine line with machine efficiency and buffer capacity randomly generated and given in Table 3.2. The production run size, also randomly generated, is $B = 55$. The transient performance measures of this system, obtained using simulation and Calculation Procedure 1 are given in Figure 3.13. From the figure, we can see that the approximation provided by the calculation procedure can track the transient performance measures of the system with high accuracy during almost the entire span of the production period. The largest approximation error typically occurs around the production completion times of the machines. Similar phenomenon has been widely observed in the lines studied. The production completion times for each machine, obtained by both simulation and calculation, are listed in Table 3.3. For this line, the approximation errors of the average completion times are all less than 0.3%.

Finally, it should be noted that the calculation procedure developed above can be applied to systems with machines having time-varying machine parameters by replacing p_i with $p_i(n)$ in the formulas. The analyses carried out in the subsequent sections, however, still assume

that the machine efficiencies are time-invariant.

Table 3.2: Machine and buffer parameters

p_1	p_2	p_3	p_4	p_5
0.9167	0.8089	0.7717	0.8284	0.7774
N_1	N_2	N_3	N_4	–
4	4	5	2	–

Table 3.3: Comparison of average production completion times obtained by calculation and simulation

	m_1	m_2	m_3	m_4	m_5
$E(\widehat{CT}_i)$	69.82	75.00	79.11	83.31	85.24
$E(CT_i^{sim})$	69.67	74.84	79.03	83.33	85.30

3.5 System-theoretic Properties and Continuous Improvement

3.5.1 Monotonicity and reversibility

Monotonicity

Consider a Bernoulli line defined by assumptions (i)-(vii). The average production completion time at each machine $E(\widehat{CT}_i)$ is *practically* monotonically decreasing in p_j, $j = 1, \ldots, M$, and N_j, $j = 1, \ldots, M-1$.

Justification: To verify the validity of this property, we use the 600,000 lines generated during the accuracy investigation carried out in Subsection 3.4.3. For each line, we increase the efficiency of one machine by 0.01 or to 0.9999, whichever is smaller, while keeping other machines efficiency and all buffers fixed, and evaluate the production completion times of the resulting system. Then, reset the efficiency of this machine to its original value and repeat this procedure for all other machines in the line. A similar procedure is carried out for the buffers by increasing each one's capacity by 1 unit. Among all cases studied,

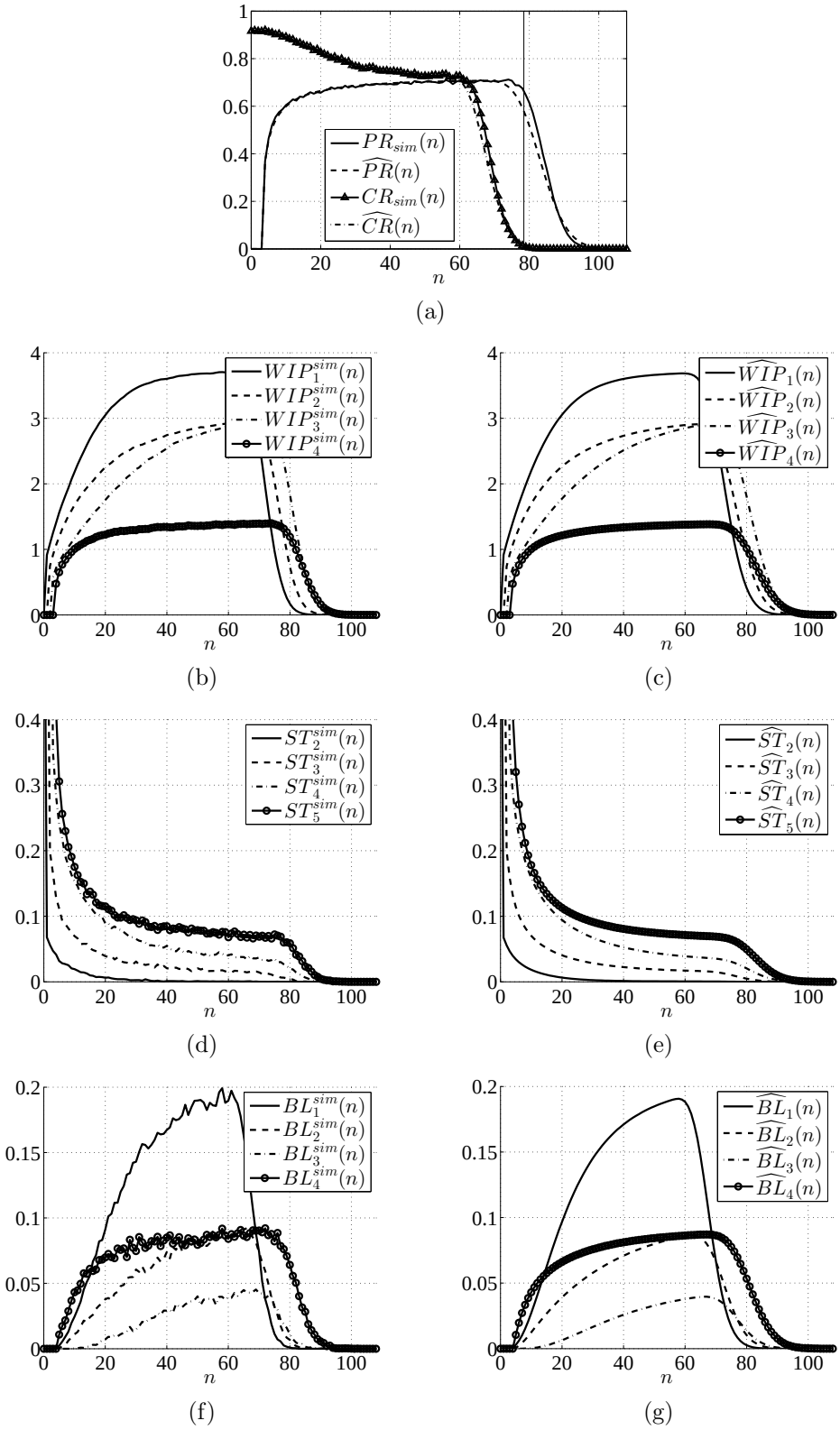


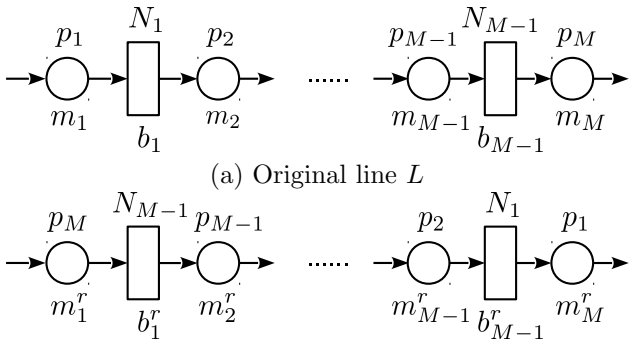
Figure 3.13: Illustration of performance measures approximation using simulation and Calculation Procedure 1

increasing p_j leads to reduction of $E(\widehat{CT}_i)$ in over 99.77% of cases. For the counter cases observed, the maximum increase in $E(\widehat{CT}_i)$ is no greater than 10^{-13}, which is attributed to the rounding errors during the calculation procedures. On the other hand, the proportion of counter cases when increasing N_j is about 19.27%. Among these cases, the increase in $E(\widehat{CT}_i)$ never exceeds 0.6%, while the average is 0.31%. To further investigate the error, we run simulations on these counter cases and no increase in $E(CT_i^{sim})$ is observed in these cases. Therefore, these counter cases are attributed to the approximation error of $E(\widehat{CT}_i)$. Based on the results from the numerical experiments above, we claim that the production completion time possesses the property of monotonicity with respect to machine efficiency and buffer capacity.

Practical implications: The implication of the monotonicity property is obvious: Improving machine efficiency and/or enlarging buffer capacity always lead to shortened average completion time for each machine in the production line.

Reversibility

Consider a Bernoulli line defined by assumptions (i)-(vii) and its reverse (see Figure 3.14). Then, the performance measures of the original line (with superscript L) and the corresponding ones of the reversed line (with superscript L_r) satisfy the following relationship:



(a) Original line L

(b) Reversed line L_r

Figure 3.14: Bernoulli serial line and its reverse

$$E(CT_M^L) \approx E(CT_M^{L_r}),$$
$$\sigma(CT_M^L) \approx \sigma(CT_M^{L_r}),$$
$$PR^L(n) \approx PR^{L_r}(n).$$

Justification: To verify this property, we, again, use the 600,000 lines generated in Subsection 3.4.3. Among all cases studied, the average errors of $\widehat{PR}(n)$, $E(\widehat{CT}_i)$, and $\sigma(\widehat{CT}_i)$ between the original and reversed lines are 0.12%, 0.07%, and 0.07%, respectively. Therefore, we claim that the mean and standard deviation of the production completion time at the last machine as well as the production rate possess the property of reversibility.

Practical implications: It follows immediately from the reversibility property that the original and reversed production lines have the same performance from the perspective of the last machine's output. This is reflected in both the production rate and the production completion time. On the other hand, the behavior of raw material consumption, work-in-process, and production completion times at other machines are different when the line is reversed. This phenomenon is illustrated in Figure 3.15, where the parameters of the

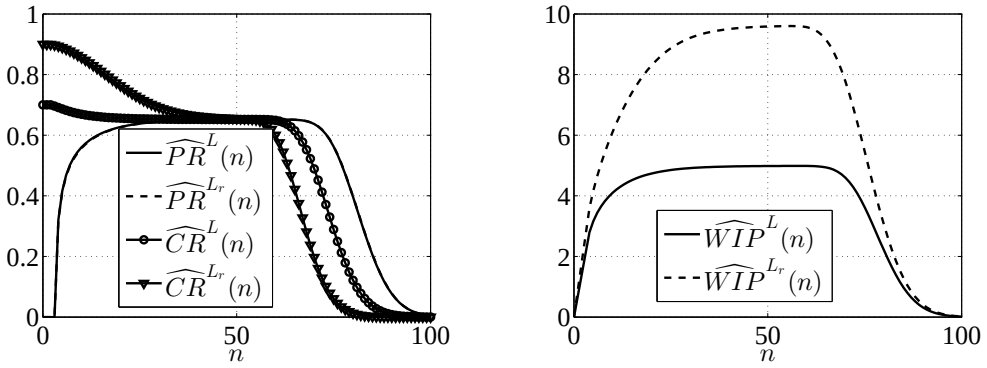


Figure 3.15: Comparison of transient performance for a Bernoulli line and its reverse

original line L are: $\mathbf{p}$ = [0.75 0.8 0.85 0.9 0.95], $\mathbf{N}$ = [3 3 3 3]. As one can see from the figure, the production rates of both lines completely overlap with each other, while obvious differences in consumption rates (see Figure 3.15(a)) and in total work-in-processes (see Figure 3.15(b)) can be directly observed. In addition, when the machines with higher

efficiency are placed upstream (as in the reversed line L_r), the total work-in-process increases. As for the production completion time, the average completion time at the last machine remains approximately the same in both L and L_r (see Table 3.4). For other machines,

Table 3.4: Mean and standard deviation of production completion time in the original and reversed lines

	$E(\widehat{CT}_i)$		$\sigma(\widehat{CT}_i)$	
	L	L_r	L	L_r
m_1	75.9438	68.9015	6.3381	5.5688
m_2	78.5488	73.1267	6.4600	5.9718
m_3	80.5731	77.0778	6.5214	6.2635
m_4	82.2228	80.6551	6.5549	6.4619
m_5	83.5975	83.6522	6.5724	6.5754

however, the production completion times in L_r are all shorter than the corresponding ones in L. Moreover, the standard deviations of the production completion times are also reduced in L_r. These advantages can be attributed to the fact that machines with higher efficiency are capable of pushing parts to the downstream in shorter time and with lower uncertainty. This may allow longer preparation time for the next production run. The drawback, obviously, is the higher level of work-in-process during the production process.

3.5.2 Bottleneck

To improve an existing system, one should focus on improving the bottleneck of the system first. In steady state analysis of Bernoulli lines, the bottleneck is usually defined based on its effect on the throughput of the overall system, i.e.,

Definition [9]: Consider a Bernoulli serial line defined by assumptions (i)-(vi). Machine m_i is the steady state production rate bottleneck (ssPRBN) machine of the system if

$$\left|\frac{\partial PR_{ss}}{\partial p_i}\right| > \left|\frac{\partial PR_{ss}}{\partial p_j}\right|, \quad \forall j \neq i. \tag{3.44}$$

For the systems considered in this Chapter, since the completion of the production run is typically the objective of the production activity, we introduce a new definition for the

bottleneck machine of the system as follows:

Definition: Consider a Bernoulli line defined by assumptions (i)-(vii). Machine m_i is the completion time bottleneck (CTBN) machine of the production line if

$$\left|\frac{\partial E(CT_M)}{\partial p_i}\right| > \left|\frac{\partial E(CT_M)}{\partial p_j}\right|, \quad \forall j \neq i. \tag{3.45}$$

When $B = 1$, it can be immediately obtained that

$$E(CT_M) = \sum_{i=1}^{M} p_i^{-1}, \tag{3.46}$$

and, therefore,

$$\left|\frac{\partial E(CT_M)}{\partial p_i}\right| = p_i^{-2}. \tag{3.47}$$

In other words, when $B = 1$, the CTBN is the machine with lowest efficiency. On the other hand, when $B \to \infty$, $E(CT_M)$ tends to B/PR_{ss} and it is easy to show that the CTBN coincides with the ssPRBN of the system.

To investigate the property of CTBN, for general values of B, numerical experiments are carried out. Specifically, we first select the number of machines M from the following set:

$$M \in \{2, 3, 5, 10, 15, 20\}.$$

Then, for each M, 10,000 lines were generated with machine efficiency and buffer capacity randomly and equiprobably selected from the following sets:

$$p_i \in (0.7, 1), \quad N_i \in \{1, 2, 3, 4, 5\}.$$

Therefore, a total of 60,000 Bernoulli lines were obtained. For each line thus generated, we

study the system bottlenecks for production run size from the following set:

$$B \in \{5, 10, 15, 20, 30, 50, 75, 100, 150, 200\}.$$

The CTBN and ssPRBN of each line are identified by increasing the efficiency of each machine one by one by 0.02 and evaluating the resulting $E(\widehat{CT}_M)$ and PR_{ss} using Calculation Procedure 1 and the method developed in [9], respectively. The machine, which leads to the largest reduction in $E(\widehat{CT}_M)$, is considered as the CTBN of the line under the current production run size, while the one leading to the largest PR_{ss} is identified as the ssPRBN. The results are summarized in Tables 3.5, 3.6, and 3.7.

Table 3.5: Percentage of cases where CTBN is the worst machine

B	$M = 2$	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
5	100%	94.96%	85.42%	68.06%	55.96%	48.22%
10	100%	93.32%	82.24%	62.75%	49.74%	42.24%
15	100%	92.48%	80.36%	59.62%	46.55%	38.54%
20	100%	91.82%	79.26%	57.86%	44.61%	37.03%
30	100%	91.04%	77.73%	55.48%	42.01%	34.32%
50	100%	90.29%	76.09%	52.85%	39.39%	32.22%
75	100%	89.95%	75.15%	51.61%	38.20%	30.66%
100	100%	89.62%	74.78%	50.97%	37.41%	30.03%
150	100%	89.34%	74.35%	50.18%	36.74%	29.29%
200	100%	89.24%	74.00%	49.95%	36.40%	28.97%

Table 3.6: Percentage of cases where CTBN is the ssPRBN

B	$M = 2$	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
5	100%	92.96%	83.67%	67.46%	59.39%	51.12%
10	100%	94.72%	88.04%	75.70%	67.54%	60.84%
15	100%	95.76%	90.49%	80.52%	73.34%	67.66%
20	100%	96.51%	91.91%	83.75%	77.97%	72.22%
30	100%	97.36%	93.77%	87.92%	83.37%	79.14%
50	100%	98.27%	95.86%	92.11%	89.22%	85.88%
75	100%	98.73%	97.19%	94.18%	92.43%	90.14%
100	100%	99.11%	97.82%	95.47%	94.34%	92.65%
150	100%	99.48%	98.43%	96.88%	96.06%	94.93%
200	100%	99.61%	98.90%	97.61%	96.87%	95.90%

It follows immediately from Table 3.5 that in two-machine Bernoulli lines, the less efficient

Table 3.7: Percentage of cases where CTBN is neither the worst machine nor the ssPRBN

B	$M = 2$	$M = 3$	$M = 5$	$M = 10$	$M = 15$	$M = 20$
5	0%	0.71%	2.53%	9.60%	16.26%	23.56%
10	0%	0.65%	2.08%	7.88%	14.69%	20.86%
15	0%	0.53%	1.77%	6.72%	12.77%	18.36%
20	0%	0.49%	1.58%	5.66%	10.74%	16.01%
30	0%	0.42%	1.36%	4.42%	8.45%	12.52%
50	0%	0.32%	1.09%	3.27%	5.85%	8.72%
75	0%	0.24%	0.79%	2.64%	4.13%	6.41%
100	0%	0.22%	0.57%	2.12%	3.25%	4.79%
150	0%	0.14%	0.42%	1.65%	2.41%	3.44%
200	0%	0.12%	0.33%	1.28%	2.01%	2.87%

machine is always the bottleneck of the system, regardless of the production run size or the buffer capacity. It should be noted that, when $M = 2$, it has been shown in [9] that the machine with lower efficiency is, in fact, the steady state production rate bottleneck. In other words, the steady state bottleneck and the transient bottleneck (defined in the manner of (3.45)) are the same. On the other hand, as it follows from Table 3.5, although the worst machine is very likely to be the CTBN for small systems (e.g., $M = 3$ or 5), it is clearly not appropriate to assume this for longer systems (e.g., $M \geqslant 10$). This observation is similar to the steady state case.

For $M > 2$, as one can see from Table 3.6, when the production run size is relatively small, the CTBN and ssPRBN could be different in a fairly noticeable amount of cases. This phenomenon is more evident when M is larger. In addition, as the production run size increases, the system operates longer in a regime close to the steady state. As the result, the CTBN of a Bernoulli line is more likely to be the ssPRBN. Specifically, for production run size over 75, CTBN is the ssPRBN in over 90% of cases studied.

Finally, Table 3.7 implies that the CTBN is practically either the worst machine or the ssPRBN if the line is short ($M < 10$) or the production run is large ($B > 75$). However, when the production run size is small and the line is relatively long, other machines may also become the CTBN. An example is given in Figure 3.16, where the numbers in the rectangles are the capacity of the buffers, and the numbers above the circles are the efficiencies of the

machines. For this system, the worst machine is m_5, while it can be found that machine m_3 is the ssPRBN. In addition, for production run size B from 1 to 120, the CTBN is machine m_2. When production run size is greater than 120, the CTBN shifts to machine m_3, the ssPRBN.

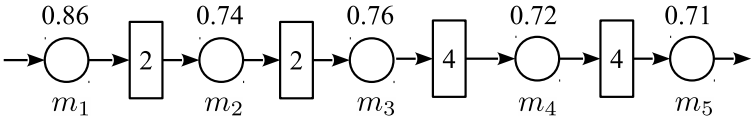


Figure 3.16: Bottlenecks in finite production run-based Bernoulli serial line

3.6 Case Study

To test the applicability of the theoretical study carried out above, part of the results developed is applied in a local lighting equipment assembly line.

3.6.1 System layout and modeling

The layout of the assembly line is shown in Figure 3.17. The system consists of 7 op-

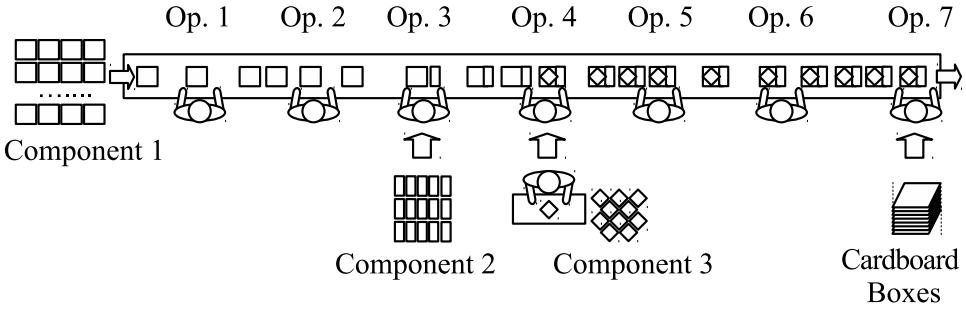


Figure 3.17: Layout of the lighting equipment assembly line

erations, all performed by human operators on a long working shelf. During the process, several components (circuit board, wires, screws, plastic covers, etc.) are assembled into the product through Op.1 to Op. 5 before it is tested at Op. 6 and packed at Op. 7. Based on the system layout, we model the production system as a serial line shown in Figure 3.18.

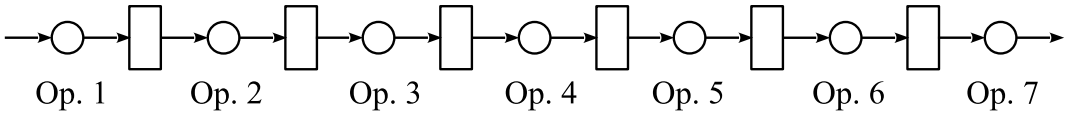


Figure 3.18: Serial line model of the lighting equipment assembly line

To identify the parameters of the operations, a one-week time study was carried out to collect the data during active production hours. The average uptime ($T_{up,i}$), average downtime ($T_{down,i}$), and cycle time (τ_i) for each operation are summarized in Table 3.8. Then, the Bernoulli parameter of each operation is calculated as follows:

$$p_i = \frac{T_{up,i}}{T_{up,i} + T_{down,i}} \cdot \frac{c_i}{\max c_i}, \quad i = 1, \ldots, 7, \tag{3.48}$$

where $c_i = 1/\tau_i$ is the processing speed of Op. i. The results are given in Table 3.9.

Table 3.8: Average uptime, downtime, and cycle time of the operations (all in minutes)

	Op. 1	Op. 2	Op. 3	Op. 4	Op. 5	Op. 6	Op. 7
$T_{up,i}$	4.13	4.01	1.90	2.82	1.65	3.88	3.37
$T_{down,i}$	1.50	1.35	0.82	0.75	0.95	1.85	1.53
τ_i	1.10	1.05	0.98	0.87	0.92	0.95	1.05

Table 3.9: Bernoulli parameters of the operations

p_1	p_2	p_3	p_4	p_5	p_6	p_7
0.58	0.62	0.62	0.79	0.60	0.62	0.57

The capacities of the buffers are obtained by measuring the maximum number of product units that can fit in the shelf space between consecutive operations. The results are summarized in Table 3.10.

Table 3.10: Buffer capacities

N_1	N_2	N_3	N_4	N_5	N_6
4	4	4	3	3	4

3.6.2 Steady state and production run performance analysis

Depending on external orders from the customers, the assembly line may work with a large production run with a single product type or small production runs containing a variety of different types of products. Typically, a large production run usually takes several shifts to complete, while a smaller production run only takes about half or a third of a shift or shorter. Therefore, the former can be viewed as steady state production, while the latter falls into the finite production run-based production addressed in this Chapter.

To validate the model constructed above, we compare the system performance predicted by the model and the one measured on the factory floor. Specifically, we calculate the steady state production rate of the 7-machine Bernoulli serial line constructed above using the PSE Toolbox software [9] and obtain the system's steady state throughput $\widehat{TP}_{ss} = 0.5811$ parts/min. Thus, the system's shift throughput is calculated as 0.5811 (parts/min) $\times$ 435 (min/shift) = 252 (parts/shift). The production personnel in the plant confirmed that, when working with a single large production run throughout a shift, the assembly line typically produced 240–260 parts per shift on average. The analysis also shows that the steady state PRBN is Op. 6, the testing station. Next, we apply the performance analysis method developed in Subsection 3.4.2 to study the finite production run performance of the assembly line. Specifically, we calculate the average production completion time at Op. 7 for different production run sizes using Calculation Procedure 1. In addition, for each B considered, we also estimate the average completion time by $B/\widehat{TP}_{ss}$, i.e., assuming the system operates as

Table 3.11: Production run performance analysis of the lighting equipment assembly line

B	$E(\widehat{CT}_M)$ (min)	$B/\widehat{TP}_{ss}$ (min)	Difference (min)
10	30.92	17.21	13.71
20	49.65	34.42	15.23
30	67.50	51.63	15.88
40	85.04	68.83	16.21
50	102.43	86.04	16.39
100	188.74	172.09	16.65
250	446.92	430.22	16.70

if it is in steady state. This method has been used by the production supervisors in their practice before the case study was carried out. The results are summarized in Table 3.11. As one can see, since the steady state method ignores the transient period of the production due to zero initial WIP, it underestimates the production completion time by about 15 minutes (see the rightmost column of Table 3.11). This underestimation caused a number of issues in production scheduling and customer demand satisfaction as the production management struggled to make up for the underestimated completion time. Based on the case study, it is suggested to the production management that, as a rule-of-thumb, an extra 15 minutes should be added to their old estimation to obtain a more accurate prediction of the production completion time. It should be noted that the 15-min rule-of-thumb is valid for this line given that no significant changes in system parameters are observed. In addition, it is found that the CTBN of the system is Op. 7 (the packaging station) for small production run ($B < 100$), while for larger production run ($B \geq 100$), the CTBN shifts to Op. 6, which is also the ssPRBN.

3.7 Summary

Finite production run-based production systems are widely seen in practice. In this Chapter, we discuss the problems of performance evaluation, system-theoretic properties, bottlenecks in the framework of serial production lines with Bernoulli machines and finite buffers. Using Markovian analysis, closed-form expressions are provided to calculate the performance measures for one- and two-machine lines. For longer lines, a computationally efficient algorithm is developed to approximate the system performance measures with high accuracy. It is shown that the average production completion times at all machines are monotonically decreasing functions of machine efficiency and buffer capacity. In addition, the system production rate and the production completion time at the last machine possess the property of reversibility. Properties of system completion time bottleneck are investigated and a case study is carried out to illustrate the applicability of the methods developed.

Chapter 4

PERFORMANCE ANALYSIS AND SYSTEM THEORETIC PROPERTIES OF SERIAL PRODUCTION LINES WITH GEOMETRIC MACHINES AND FINITE PRODUCTION RUNS

4.1 Introduction

In this Chapter, we extend the idea of the algorithm developed for Bernoulli systems to serial production lines with machines having the geometric reliability model. This is accomplished by applying the transient performance evaluation techniques developed by [43] for geometric serial lines.

Specifically, The Chapter is organized as follows: Section 4.2 introduces the model and defines the performance measures for the systems under consideration. In Section 4.3, we derive formulas to evaluate the performance measures of one- and two-machine systems using exact Markovian analysis. Then, we propose computationally efficient calculation procedures to approximate the performance measures of two- and multi-machine lines with high accuracy in Section 4.4. The system-theoretic properties of production run completion time are discussed in Section 4.5.

4.2 Model and Performance Measures

4.2.1 Model

Consider a serial production line in Figure 4.1 defined by the following assumptions:

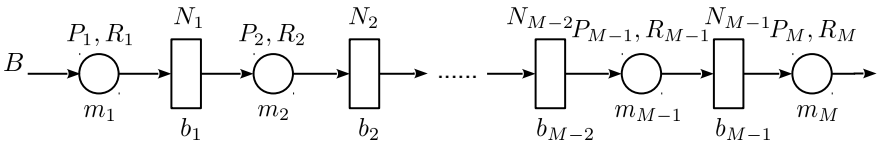


Figure 4.1: Serial production line with geometric machines

(i) The system consists of M machines (represented by circles) and $M-1$ buffers (represented by rectangles). The arrows indicate the direction of parts flow.

(ii) The machines have identical and constant cycle time τ. The time axis is slotted with slot duration τ. Machines begin operating at the beginning of each time slot.

(iii) The machines obey the geometric reliability model: Let $s_i(n) \in \{0 = \text{down}, 1 = \text{up}\}$ denote the state of machine m_i during time slot n, $i = 1, \ldots, M$. Then, the transition probabilities are given by

$$\begin{aligned} &\text{Prob}[s_i(n+1) = 0 | s_i(n) = 1] = P_i, \qquad \text{Prob}[s_i(n+1) = 1 | s_i(n) = 1] = 1 - P_i, \\ &\text{Prob}[s_i(n+1) = 1 | s_i(n) = 0] = R_i, \qquad \text{Prob}[s_i(n+1) = 0 | s_i(n) = 0] = 1 - R_i, \end{aligned} \tag{4.1}$$

where P_i and R_i are referred to as the *breakdown and repair probabilities*, respectively. All machines operate independently from one another.

(iv) Each buffer is characterized by its capacity (i.e., the maximum number of parts the buffer can hold), $1 \leq N_i < \infty$, $i = 1, \ldots, M-1$.

(v) Machine m_i, $i = 2, \ldots, M$, is starved during a time slot if it is up and buffer b_{i-1} is empty at the beginning of the time slot. Machine m_1 is never starved for raw material.

(vi) Machine m_i, $i = 1, \ldots, M-1$, is blocked during a time slot if it is up, buffer b_i has N_i parts at the beginning of the time slot, and machine m_{i+1} fails to take a part during the time slot (due to breakdown or blockage). Machine m_M is never blocked.

(vii) If a machine is up and neither starved nor blocked, it processes one part in one time slot (i.e., takes one part from its upstream buffer at the beginning of the time slot, processes it during the time slot, and sends it to its downstream buffer at the end of the time slot); otherwise, no processing takes place for the machine in this time slot.

(viii) The system operates on a finite production run-basis with the volume of the production run equal to B parts: All buffers are initially empty and each machine stops operating as soon as it has finished processing B parts.

Remark 4.1: Under assumption (iii), the up- and downtime of machine m_i are geometric random variables. Apparently, the mean of its up- and downtime are given by $T_{up,i} = 1/P_i$ and $T_{down,i} = 1/R_i$, respectively. These two quantities are also called mean time between failure (MTBF) and mean time to repair (MTTR), respectively, in practice. In addition, the machine *efficiency*, i.e., the probability (fraction of time) that m_i is up in steady state, is given by $e_i = T_{up,i}/(T_{up,i} + T_{down,i}) = R_i/(R_i + P_i)$.

Remark 4.2: The geometric reliability model is usually applicable, when the machine's average downtime is much longer than its cycle time (e.g., in machining, heat treatment, washing operations). Steady state behavior of the geometric serial lines has been studied in a number of publications in production systems research (see [2, 3, 5, 7]). The geometric model has also been successfully applied in industrial case studies (see, for instance, [90, 91]). It should be also noted that the Bernoulli reliability model can be viewed as a special case of the geometric model when $P_i + R_i = 1$. In other words, paper [81] treats a special case of the system considered here.

4.2.2 Performance measures

In the framework of the model defined above, the performance measures of interest include:

- Production rate, $PR(n)$: the expected number of parts produced by m_M during time slot $n+1$;
- Consumption rate, $CR(n)$: the expected number of parts consumed by m_1 during time slot $n+1$;
- Work-in-process, $WIP_i(n)$: the expected number of parts in buffer b_i, $i = 1, \ldots, M-1$, at the beginning time slot $n+1$.

In addition, let CT_i denote the time instant when machine m_i completes all B parts. Clearly, CT_i is a discrete random variable. We denote its probability mass function as:

$$P_{ct_i}(n) = \text{Prob}[CT_i = n]. \tag{4.2}$$

Clearly, $P_{ct_i}(n) = 0$ for all $0 \leq n < B$. Thus, the mean and standard deviation of CT_i are given by:

$$\mu_{CT_i} = \sum_{n=B}^{\infty} nP_{ct_i}(n), \qquad \sigma_{CT_i} = \sqrt{\sum_{n=B}^{\infty} [n - \mu_{CT_i}]^2 P_{ct_i}(n)}. \tag{4.3}$$

In this Chapter, we will derive analytical methods to calculate these performance measures.

4.3 Exact Performance Evaluation: One- and Two-Machine Lines

4.3.1 One-machine lines

To begin with, we first study systems with only one machine (see Figure 4.2). Due

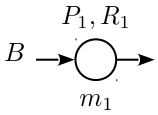


Figure 4.2: One-machine line

to the memoryless property of the geometric random variable, a production system with geometric machines is characterized by a Markov chain. Since a one-machine system does not have a buffer, the state of the system at time n can be defined by the combination of the machine's state during this time slot (denoted as $s(n) \in \{0 = \text{down}, 1 = \text{up}\}$) and the number of products that have been completed at the beginning of this time slot (denoted as $f(n) \in \{0, 1, \ldots, B\}$). As a result, the overall system state can be represented by a pair $(s(n), f(n))$. Clearly, the total number of states is $2 \times (B + 1)$.

On the other hand, since the system stops operating as soon as B parts are completed according to assumption (viii), it must be in state $(1, B-1)$ at the beginning of this time slot. This implies that both states $(0, B)$ and $(1, B)$ indicate that the production run is complete. In other words, we can simply combine the two states into one and view this combined state as an absorbing state. Without loss of generality, let this absorbing state be $(1, B)$. Based on the system description given in Subsection 4.2.1, the transition probabilities of this Markov chain are given by:

$$\text{Prob}[s(n+1) = 0, f(n+1) = a | s(n) = 0, f(n) = a] = 1 - R_1, \quad a = 0, 1, \ldots, B-1,$$

$$\text{Prob}[s(n+1) = 1, f(n+1) = a | s(n) = 0, f(n) = a] = R_1, \quad a = 0, 1, \ldots, B-1,$$

$$\text{Prob}[s(n+1)=0, f(n+1)=a+1 | s(n)=1, f(n)=a] = P_1, \quad a=0,1,\ldots,B-2,$$

$$\text{Prob}[s(n+1)=1, f(n+1)=a+1 | s(n)=1, f(n)=a] = 1-P_1, \quad a=0,1,\ldots,B-2,$$

$$\text{Prob}[s(n+1)=1, f(n+1)=B | s(n)=1, f(n)=B-1] = 1,$$

$$\text{Prob}[s(n+1)=1, f(n+1)=B | s(n)=1, f(n)=B] = 1.$$

The transition probabilities between other state pairs all equal zero. Clearly, the distribution of the production run completion time, CT_1, is related to the state of the system as:

$$P_{ct_1}(n) = P[s(n)=1, f(n)=B-1].$$

In other words, the completion time of the production run is also the time-to-absorption of the Markov chain. Therefore, it is possible to calculate the mean and standard deviation of the production completion time using the properties of absorbing Markov chains [88]. Specifically, arrange the states of the Markov chain from number 1 to number $2B+1$ according to the following arrangement:

$$\text{state number of } (s,f) = s \times B + f + 1.$$

Thus, the initial state of the system is either state 1 (if the machine is initially down) or state $B+1$ (if the machine is initially up), while the absorbing state is state $2B+1$. Let $\mathbf{A}$ denote the transition probability matrix of the Markov chain. Then, $\mathbf{A}$ can be expressed as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q} & \mathbf{0}_{2B,1} \\ \mathbf{V} & 1 \end{bmatrix}, \tag{4.4}$$

where $\mathbf{I}_k$ represents a k-by-k identity matrix and $\mathbf{0}_{k,l}$ represents a k-by-l zero matrix.

Next, we can calculate its fundamental matrix: $\mathbf{F} = (\mathbf{I}-\mathbf{Q})^{-1}$. For this absorbing Markov chain, according to [88], the expected time before being absorbed when starting in state i is

the ith entry of the vector

$$\mathbf{t} = \mathbf{1} \cdot \mathbf{F}, \tag{4.5}$$

where $\mathbf{1}$ is a row vector with all entries equal to 1. In addition, the variance of the amount of time before being absorbed when starting in state i is the ith entry of the vector

$$\mathbf{v} = \mathbf{t}(2\mathbf{F} - \mathbf{I}) - \mathbf{t} \circ \mathbf{t}, \tag{4.6}$$

where "$\circ$" represents the Hadamard product (i.e., the element-wise product of the two vectors involved). If the machine is initially down, i.e., the system starts from $(s = 0, f = 0)$, the mean and standard deviation of completion time CT_1 are given by the first entry of $\mathbf{t}$ and the square root of the first entry of $\mathbf{v}$, respectively. If the machine is initially up, i.e., the system starts from $(s = 1, f = 0)$, μ_{CT_1} and σ_{CT_1} are given by the $(B+1)$-th entry of $\mathbf{t}$ and the square root of the $(B+1)$-th entry of $\mathbf{v}$, respectively.

To calculate the transient performance of this system during the production run, the following procedure is taken: Let $\mathbf{w}(n) = [w_1(n) \ \cdots \ w_S(n)]^T$, where $w_i(n) = \text{Prob}[\text{system in state } i \text{ in time slot } n]$ and $S = 2B + 1$. Then, the evolution of $\mathbf{w}(n)$ is given by

$$\mathbf{w}(n+1) = \mathbf{A}\mathbf{w}(n),$$

with initial condition

$$w_1(0) = \begin{cases} 1, & \text{if } s(0) = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad w_{B+1}(0) = \begin{cases} 1, & \text{if } s(0) = 1, \\ 0, & \text{otherwise,} \end{cases} \qquad w_i(0) = 0, \quad \forall i \neq 1, B+1.$$

Then, according to the state number assignment, the transient performance measures of the one-machine system can be calculated as:

$$\begin{aligned} PR(n) = CR(n) &= \text{Prob}[\text{machine is up and production run is not completed}] \\ &= [\underbrace{0 \ \cdots \ 0}_{B \text{ zeros}} \quad \underbrace{1 \ \cdots \ 1}_{B \text{ ones}} \quad 0]\mathbf{w}(n). \end{aligned} \tag{4.7}$$

Finally, if the machine's breakdown and repair probabilities are time-varying (denoted as $P_1(n)$ and $R_1(n)$ in time slot n), the system is still characterized by a Markov chain, but a time-varying one. In this case, the status of the machine and the number of finished products remain as the state of the Markov chain, while the transition probability matrix $\mathbf{A}$ becomes time-varying (denoted as $\mathbf{A}(n)$) with P_1 and R_1 replaced by $P_1(n)$ and $R_1(n)$, respectively. Thus, the evolution of $\mathbf{w}(n)$ is now given by

$$\mathbf{w}(n+1) = \mathbf{A}(n)\mathbf{w}(n).$$

For the time-varying system, while the performance measures can still be calculated using (4.7), the derivation of (4.5) and (4.6) does not apply any more. As an alternative, we can calculate the mean of the completion time based on its probability distribution as follows:

$$\mu_{CT_1} = \sum_{n=B}^{\infty} n \cdot w_{2B}(n).$$

Numerically, this can be approximated by

$$\mu_{CT_1} \approx \sum_{n=B}^{D_0} n \cdot w_{2B}(n),$$

where

$$1 - \sum_{n=1}^{D_0} w_{2B}(n) < \epsilon \ll 1.$$

4.3.2 Two-machine lines

In the case of two-machine lines (see Figure 4.3), the exact Markovian analysis approach is still applicable. For these systems, the state of the Markov chain is characterized by a quadruple (h, f, s_1, s_2), where h represents the number of parts in the buffer, f is the number

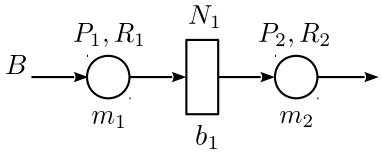


Figure 4.3: Two-machine line

of products that have been completed and s_i, $i = 1, 2$, denotes the state of machine m_i.

$$\alpha(h, f, s_1, s_2) = \begin{cases} \sum_{i=0}^{f-1} 4 \times (N_1 + 1) + (N_1 + 1) \times (s_1 \times 2^{s_1} + s_2) + h + 1, \\ \qquad\qquad \text{if } f \leqslant B - N_1 \text{ and } B > N_1, \\ \sum_{i=0}^{B-N_1} 4 \times (N_1 + 1) + \sum_{i=B-N_1+1}^{f-1} 4 \times (B - i + 1) \\ \qquad +(B - f + 1) \times (s_1 \times 2^{s_1} + s_2) + h + 1, \\ \qquad\qquad \text{if } B - N_1 < f < B \text{ and } B > N_1, \\ \sum_{i=0}^{B-N_1} 4 \times (N_1 + 1) + \sum_{i=B-N_1+1}^{B-1} 4 \times (B - i + 1) + 1, \\ \qquad\qquad \text{if } f = B \text{ and } B > N_1, \\ \sum_{i=0}^{f-1} 4 \times (B - i + 1) + (B - f + 1) \times (s_1 \times 2^{s_1} + s_2) + h + 1, \\ \qquad\qquad \text{if } f < B \text{ and } B \leqslant N_1, \\ \sum_{i=0}^{B-1} 4 \times (B - i + 1) + 1, \\ \qquad\qquad \text{if } f = B \text{ and } B \leqslant N_1. \end{cases} \tag{4.8}$$

Clearly, the final states after completing the production run are $(0, B, 0, 0)$, $(0, B, 0, 1)$, $(0, B, 1, 0)$, $(0, B, 1, 1)$. They imply that the system has produced B parts, the buffer occupancy is 0, and the machine states can be either 0 (*down*) or 1 (*up*) after completion. To simplify the analysis, these four states are combined into one absorbing state $(0, B, -, -)$. In addition, since the machines stop operating when it has finished B parts, h and f must satisfy $h + f \leq B$. Thus, the total number of system states is given by

$$S = \begin{cases} \sum_{i=0}^{B-N_1} 4(N_1 + 1) + \sum_{i=B-N_1+1}^{B-1} 4(B - i + 1) + 1, & \text{if } B > N_1, \\ \sum_{i=0}^{B-1} 4(B - i + 1) + 1, & \text{if } B \leqslant N_1. \end{cases} \tag{4.9}$$

To analyze this Markov chain, we define the mapping (4.8) to assign a unique state number α from 1 to S to each of the system state.

To calculate the transient performances of such system during the production run, the following procedure is taken: Let $\mathbf{w_2}(n) = [w_{2,1}(n) \ \cdots \ w_{2,S}(n)]^T$, where $w_{2,i}(n) = \text{Prob[system in state } i \text{ in time slot } n]$. Then, the evolution of $\mathbf{w_2}(n)$ is given by

$$\mathbf{w_2}(n+1) = \mathbf{A_2}\mathbf{w_2}(n),$$

where $\mathbf{A_2}$ is the transition matrix with the transition probabilities among system states defined by:

$$\left.\begin{aligned}
&P[w_{2,\alpha(a,b,0,0)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = (1-R_1)(1-R_2),\\
&P[w_{2,\alpha(a,b,0,1)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = (1-R_1)R_2,\\
&P[w_{2,\alpha(a,b,1,0)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = R_1(1-R_2),\\
&P[w_{2,\alpha(a,b,1,1)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = R_1R_2,
\end{aligned}\right\} \quad \begin{aligned} &a = 0, ..., N_1;\\ &b = 0, ..., B-1;\\ &a+b \leqslant B; \end{aligned} \tag{4.10}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(0,b,0,0)}(n+1)|w_{2,\alpha(0,b,0,1)}(n)] = (1-R_1)P_2,\\
&P[w_{2,\alpha(0,b,0,1)}(n+1)|w_{2,\alpha(0,b,0,1)}(n)] = (1-R_1)(1-P_2),\\
&P[w_{2,\alpha(0,b,1,0)}(n+1)|w_{2,\alpha(0,b,0,1)}(n)] = R_1P_2,\\
&P[w_{2,\alpha(0,b,1,1)}(n+1)|w_{2,\alpha(0,b,0,1)}(n)] = R_1(1-P_2),
\end{aligned}\right\} \quad b = 0, 1, ..., B-1; \tag{4.11}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(a+1,b,0,0)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = P_1(1-R_2),\\
&P[w_{2,\alpha(a+1,b,0,1)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = P_1R_2,\\
&P[w_{2,\alpha(a+1,b,1,0)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = (1-P_1)(1-R_2),\\
&P[w_{2,\alpha(a+1,b,1,1)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = (1-P_1)R_2,
\end{aligned}\right\} \quad \begin{aligned}
&a = 0,1,...,N_1-1;\\
&b = 0,1,...,B-1;\\
&a+b < B;
\end{aligned} \tag{4.12}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(a,b,0,0)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = P_1(1-R_2),\\
&P[w_{2,\alpha(a,b,0,1)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = P_1R_2,\\
&P[w_{2,\alpha(a,b,1,0)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = (1-P_1)(1-R_2),\\
&P[w_{2,\alpha(a,b,1,1)}(n+1)|w_{2,\alpha(a,b,1,0)}(n)] = (1-P_1)R_2,
\end{aligned}\right\} \quad \begin{aligned}
&a = 0,1,...,N_1-1;\\
&b = 0,1,...,B-1;\\
&a+b = B;
\end{aligned} \tag{4.13}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(1,b,0,0)}(n+1)|w_{2,\alpha(0,b,1,1)}(n)] = P_1P_2,\\
&P[w_{2,\alpha(1,b,0,0)}(n+1)|w_{2,\alpha(0,b,1,1)}(n)] = P_1(1-P_2),\\
&P[w_{2,\alpha(1,b,1,0)}(n+1)|w_{2,\alpha(0,b,1,1)}(n)] = (1-P_1)P_2,\\
&P[w_{2,\alpha(1,b,1,1)}(n+1)|w_{2,\alpha(0,b,1,1)}(n)] = (1-P_1)(1-P_2),
\end{aligned}\right\} \quad b = 0,1,...,B-1; \tag{4.14}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(a-1,b+1,0,0)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = (1-R_1)P_2,\\
&P[w_{2,\alpha(a-1,b+1,0,1)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = (1-R_1)(1-P_2),\\
&P[w_{2,\alpha(a-1,b+1,1,0)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = R_1P_2,\\
&P[w_{2,\alpha(a-1,b+1,1,1)}(n+1)|w_{2,\alpha(a,b,0,0)}(n)] = R_1(1-P_2),
\end{aligned}\right\} \quad \begin{aligned}
&a = 0,1,...,N_1;\\
&b = 0,1,...,B-2;\\
&a+b \leqslant B;
\end{aligned} \tag{4.15}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(N,b,0,0)}(n+1)|w_{2,\alpha(N,b,1,0)}(n)] = P_1(1-R_2),\\
&P[w_{2,\alpha(N,b,0,1)}(n+1)|w_{2,\alpha(N,b,1,0)}(n)] = P_1R_2,\\
&P[w_{2,\alpha(N,b,1,0)}(n+1)|w_{2,\alpha(N,b,1,0)}(n)] = (1-P_1)(1-R_2),\\
&P[w_{2,\alpha(N,b,1,1)}(n+1)|w_{2,\alpha(N,b,1,0)}(n)] = (1-P_1)R_2,
\end{aligned}\right\} \quad \begin{aligned} &b = 0,1,...,B-1;\\ &b \leqslant B-N_1; \end{aligned} \tag{4.16}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(a,b+1,0,0)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = P_1P_2,\\
&P[w_{2,\alpha(a,b+1,0,1)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = P_1(1-P_2),\\
&P[w_{2,\alpha(a,b+1,1,0)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = (1-P_1)P_2,\\
&P[w_{2,\alpha(a,b+1,1,1)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = (1-P_1)(1-P_2),
\end{aligned}\right\} \quad \begin{aligned} &a = 1,...,N_1;\\ &b = 0,1,...,B-2;\\ &a+b < B; \end{aligned} \tag{4.17}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(a-1,b+1,0,0)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = P_1P_2,\\
&P[w_{2,\alpha(a-1,b+1,0,1)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = P_1(1-P_2),\\
&P[w_{2,\alpha(a-1,b+1,1,0)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = (1-P_1)P_2,\\
&P[w_{2,\alpha(a-1,b+1,1,1)}(n+1)|w_{2,\alpha(a,b,1,1)}(n)] = (1-P_1)(1-P_2),
\end{aligned}\right\} \quad \begin{aligned} &a = 1,...,N_1;\\ &b = 0,1,...,B-2;\\ &a+b = B; \end{aligned} \tag{4.18}$$

$$\left.\begin{aligned}
&P[w_{2,\alpha(0,B,-,-)}(n+1)|w_{2,\alpha(1,B-1,0,1)}(n)] = 1,\\
&P[w_{2,\alpha(0,B,-,-)}(n+1)|w_{2,\alpha(1,B-1,1,1)}(n)] = 1,\\
&P[w_{2,\alpha(0,B,-,-)}(n+1)|w_{2,\alpha(0,B,-,-)}(n)] = 1,
\end{aligned}\right\} \quad \begin{aligned} &a = 1,...,N_1;\\ &b = 0,1,...,B-2;\\ &a+b < B. \end{aligned} \tag{4.19}$$

Note that state S is the absorbing state $(0, B, -, -)$.

The exact transient performance measures of the two-machine systems are summarized

as follows:

$$
\begin{aligned}
&PR(n+1) = \text{Prob}[\cup\{w_{2,\alpha(a,b,s_1,1)}(n)\}], \quad a > 0, \quad b < B, \quad s_1 \in \{0,1\},\\
&CR(n+1) = \text{Prob}[\cup\{w_{2,\alpha(a,b,1,0)}(n)\}] + \text{Prob}[\cup\{w_{2,\alpha(c,d,1,1)}(n)\}],\\
&\qquad\qquad a < N_1, \quad a+b < B, \quad c+d < B,\\
&WIP(n+1) = \sum_{i=0}^{N_1} i\text{Prob}[\cup\{w_{2,\alpha(a,b,s_1,s_2)}(n)\}], \quad a = i, \quad s_1, s_2 \in \{0,1\},\\
&BL_1(n+1) = \text{Prob}[\cup\{w_{2,\alpha(N_1,b,1,0)}(n)\}], \quad N_1 + b < B,\\
&ST_2(n+1) = \text{Prob}[\cup\{w_{2,\alpha(0,b,s_1,1)}(n)\}], \quad b < B, \quad s_1 \in \{0,1\},\\
&P_{ct_1}(n) = \text{Prob}[\cup\{w_{2,\alpha(a,b,1,1)}(n)\}] + \text{Prob}[\cup\{w_{2,\alpha(c,d,1,0)}(n)\}],\\
&\qquad\qquad a+b = B-1, \quad a \le N_1, \quad c+d = B-1, \quad c < N_1,\\
&P_{ct_2}(n) = \text{Prob}[\cup\{w_{2,\alpha(1,B-1,s_1,1)}(n)\}], \quad s_1 \in \{0,1\}.
\end{aligned}
\tag{4.20}
$$

The completion time at each machine can be calculated as:

$$
\mu_{CT_1} = \sum_{n=B}^{\infty} n \cdot P_{ct_1}(n), \qquad \mu_{CT_2} = \sum_{n=B}^{\infty} n \cdot P_{ct_2}(n).
$$

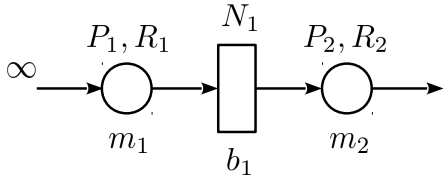
Similar to the one-machine case, the mean and standard deviation of the completion time of the production run at the last machine, i.e., μ_{CT_2}, can also be calculated based on the property of the absorbing state of the Markov chain.

4.4 Aggregation-based Approximate Performance Evaluation for Multi-Machine Lines

4.4.1 Aggregation procedure for two-machine lines

While Subsection 4.3.2 derives the exact formulas for performance evaluation in two-machine lines, the complexity of the approach becomes much greater than the one-machine

case. For example, consider a two-machine geometric line with a buffer of capacity $N_1 = 15$ and production run size $B = 80$. According to (4.9), the underlying Markov chain has a total of 4701 states. On the other hand, a one-machine line with $B = 80$ only has 161 states. In this subsection, an approximate method with significantly less computational requirement is pursued. The core of this method is called *equivalent aggregation* developed in [43], which aims to represent the aggregated behavior of a two-machine geometric line using an *equivalent* single machine. Specifically, consider a two-machine geometric line with infinite production run (i.e., $B = \infty$) shown in Figure 4.4(a). For this line, two types of aggregations are defined: *backward aggregation* and *forward aggregation*. In the former, the in-process buffer b_1 and the downstream machine m_2 are aggregated in the backward direction with m_1 to form virtual geometric machine $\widehat{m}_1$ with time-varying breakdown and repair probabilities $\widehat{P}_1(n)$ and $\widehat{R}_1(n)$, respectively (see Figure 4.4(b)). In the latter, the buffer and the upstream machine are aggregated in the forward direction with m_2 to form virtual geometric machine $\widehat{m}_2$ with time-varying breakdown and repair probabilities $\widehat{P}_2(n)$ and $\widehat{R}_2(n)$, respectively (see Figure 4.4(c)). Under this representation, the consumption of raw parts at the input of the line is characterized by virtual machine $\widehat{m}_1$, while the production of finished products is characterized by virtual machine $\widehat{m}_2$. The formulas for calculating virtual machine parameters $\widehat{P}_1(n)$, $\widehat{R}_1(n)$, $\widehat{P}_2(n)$ and $\widehat{R}_2(n)$ are derived by [43].



(a) Two-machine geometric serial line

(b) Backward aggregation (c) Forward aggregation

Figure 4.4: Two-machine geometric line and its equivalent aggregation

Now consider the resulting two virtual one-machine systems, $\widehat{m}_1$ and $\widehat{m}_2$. Assume that each operates independently on a production run of B parts. Apparently, the results from Subsection 4.3.1 become applicable. Let $\widehat{\mathbf{w}}^{(i)} = [w_1^{(i)} \ \cdots \ w_S^{(i)}]^T$, $i = 1, 2$, where $S = 2B + 1$ and $w_j^{(i)}$ denotes the probability that the production run on $\widehat{m}_1$ (if $i = 1$) or $\widehat{m}_2$ (if $i = 2$) is in state j of its corresponding Markov chain. Note that since the breakdown and repair probabilities of $\widehat{m}_1$ and $\widehat{m}_2$ are all time-varying, both transition probability matrices are also time-varying. Let $\widehat{\mathbf{A}}^{(1)}(n)$ and $\widehat{\mathbf{A}}^{(2)}(n)$ denote the transition probability matrices of the Markov chains for $\widehat{m}_1$ and $\widehat{m}_2$ to process B parts, respectively. Then, following the discussion of Subsection 4.3.1, we have

$$\widehat{\mathbf{w}}^{(1)}(n+1) = \widehat{\mathbf{A}}^{(1)}(n)\widehat{\mathbf{w}}^{(1)}(n), \quad \widehat{\mathbf{w}}^{(2)}(n+1) = \widehat{\mathbf{A}}^{(2)}(n)\widehat{\mathbf{w}}^{(2)}(n).$$

To evaluate (approximate) the performance of the original system in Figure 4.3, the following formulas are proposed:

$$\widehat{PR}(n) = [\underbrace{0 \ \cdots \ 0}_{B \text{ zeros}} \quad \underbrace{1 \ \cdots \ 1}_{B \text{ ones}} \quad 0]\widehat{\mathbf{w}}^{(2)}(n), \tag{4.21}$$

$$\widehat{CR}(n) = [\underbrace{0 \ \cdots \ 0}_{B \text{ zeros}} \quad \underbrace{1 \ \cdots \ 1}_{B \text{ ones}} \quad 0]\widehat{\mathbf{w}}^{(1)}(n), \tag{4.22}$$

$$\widehat{P}_{ct_i}(n) = \widehat{w}_{2B}^{(i)}(n), \tag{4.23}$$

$$\widehat{\mu}_{CT_i} = \sum_{n=B}^{\infty} n \cdot \widehat{w}_{2B}^{(i)}(n). \tag{4.24}$$

In addition, let $WIP_1^{(\infty)}(n)$ denote the expected work-in-process of the two-machine line of Figure 4.4(a) (i.e., with infinite production run $B = \infty$) at the end of time slot n, which can be calculated using the formulas derived by [43]. We propose to approximate the expected number of products in the buffer at the end of time slot n of the original system (i.e., with

finite production run of B parts) as follows:

$$\widehat{WIP}_1(n) = WIP_1^{(\infty)}(n) \cdot \left(1 - \sum_{j=1}^{n} \widehat{P}_{ct_2}(j)\right). \tag{4.25}$$

As one can see, $\widehat{WIP}_1(n)$ is calculated by reducing $WIP_1^{(\infty)}(n)$ with the weight equal to the probability that the production run is still unfinished at m_2.

Compared to the exact analysis approach presented in Subsection 4.3.2, the aggregation-based method requires computations on three smaller Markov chains instead of a large one. Consider again a two-machine geometric line with buffer capacity 15 and production run size 80. The aggregation-based approach only needs to deal with one Markov chain with 388 states (for the two-machine line with infinite production run) and two other Markov chains (for the one-machine lines), each with 161 states. The exact analysis approach, on the other hand, has to deal with a 4701-state Markov chain.

To assess the accuracy of the approximation method, numerical experiments were carried out. A C++ program was developed to implement the equivalent aggregation method and the approximation formulas above. Next, we generated 100,000 two-machine geometric lines with system parameters randomly and uniformly selected from:

$$R_i \in (0.05, 0.5), \quad e_i \in (0.6, 0.99). \tag{4.26}$$

As a result, the average downtime of a machine was randomly selected from 2 to 20 cycle times, with efficiency from 60% to 99%. These parameter ranges were used so that they can represent typical production situations on the factory floor. Then, the breakdown probability can be calculated based on the relationship that $P_i = R_i(1/e_i - 1)$. The capacity of the buffer was randomly selected from

$$N_1 \in \{\lceil T_{down,i} \rceil, \lceil T_{down,i} \rceil + 1, \ldots, 5\lceil T_{down,i} \rceil\}. \tag{4.27}$$

The initial state of each machine was selected up or down with probability 0.5. The size of the production run was randomly selected from

$$B \in \{20, \ldots, 120\}. \tag{4.28}$$

To evaluate the accuracy of approximations (4.21)-(4.25), we calculate the approximation errors for each line based on:

$$\Delta_{CT_i} = \frac{|\widehat{\mu}_{CT_i} - \mu_{CT_i}|}{\mu_{CT_i}} \cdot 100\%, \quad i = 1, 2, \tag{4.29}$$

$$\Delta_{PR} = \frac{1}{T} \sum_{n=1}^{T} \frac{\left|\widehat{PR}(n) - PR(n)\right|}{PR_{ss}} \cdot 100\%, \tag{4.30}$$

$$\Delta_{CR} = \frac{1}{T} \sum_{n=1}^{T} \frac{\left|\widehat{CR}(n) - CR(n)\right|}{PR_{ss}} \cdot 100\%, \tag{4.31}$$

$$\Delta_{WIP_i} = \frac{1}{T} \sum_{n=1}^{T} \frac{\left|\widehat{WIP_i}(n) - WIP_i(n)\right|}{N_i} \cdot 100\%, \quad i = 1, \tag{4.32}$$

where μ_{CT_i}, $PR(n)$, $CR(n)$, and $WIP_i(n)$ are the performance measures obtained by exact analysis, PR_{ss} is the steady state production rate, and T is the smallest time instant such that inequality

$$\max\left\{\sum_{n=1}^{T} \widehat{P}_{ct,2}(n), \sum_{n=1}^{T} P_{ct,2}(n)\right\} \geq 0.999$$

is observed for this line. Note that, for Δ_{PR}, Δ_{CR}, and Δ_{WIP_i}, the errors are normalized to PR_{ss} and N_i to maintain consistent accuracy assessment throughout the entire production run period. The statistics of these metrics are summarized in Table 4.1. As one can see, the average approximation error of μ_{CT_i} is well below 1% and the error is below 2.31% for 99% of the cases studied. The approximation errors for $PR(n)$ and $CR(n)$ are similar: within 1% on average and below 2.9% in 99% of the cases. The approximation error for $WIP_1(n)$ is the highest among all but still within 1% on average. It should be noted that the outliers

observed during the investigation are mostly from cases with very small production runs. The resulting short production completion times for these cases "amplify" the larger errors, which may be just 2 or 3 time slots in absolute difference. Considering the fact that the machine and buffer parameters of a production line are rarely known in practice with accuracy better than 5%-10%, we conclude that the aggregation-based method proposed above can be used as effective tools to estimate the transient performance and average production run completion time of two-machine geometric serial lines with good accuracy.

Table 4.1: Approximation error of equivalent aggregation in two-machine lines

	Δ_{CT_1}	Δ_{CT_2}	Δ_{PR}	Δ_{CR}	Δ_{WIP_1}
Median	0.11%	0.10%	0.39%	0.20%	0.61%
Mean	0.32%	0.31%	0.56%	0.38%	1.16%
Standard deviation	0.46%	0.49%	0.58%	0.51%	1.16%
90th percentile	0.69%	0.67%	1.25%	0.93%	2.26%
99th percentile	2.31%	2.46%	2.89%	2.74%	5.44%

4.4.2 Aggregation procedure for $M > 2$-machine lines

For $M > 2$-machine lines, while the exact Markovian analysis-based approach is still theoretically applicable, the number of states in the Markov chain grows exponentially and the computational resources required is far from being practical. Thus, a computationally efficient method must be pursued. To accomplish this, note that an aggregation-based calculation procedure was derived by [43] to approximate the transient performance of a serial production line with geometric machines and *infinite* production run (i.e., $B = \infty$). In this procedure, the M-machine serial production line is represented by $M-1$ virtual two-machine lines constructed around each buffer in the M-machine line (see Figure 4.5). Specifically, at buffer b_i, virtual machines m_i^f and m_{i+1}^b are introduced to behave as an "aggregation" of all machines and buffers upstream and downstream of b_i, respectively. Their breakdown and repair probabilities $P_i^f(n)$, $R_i^f(n)$, $P_{i+1}^b(n)$, $R_{i+1}^b(n)$ are time-varying so as to capture the transient behavior of the system. The calculation of the values of these parameters is

designed such that the in- and outbound part streams at buffer b_i in the M-machine line are approximately equal to those in the virtual two-machine line. The formulas for calculating $P_i^f(n)$, $R_i^f(n)$, $P_{i+1}^b(n)$, and $R_{i+1}^b(n)$ can be found in [43] and omitted in this Chapter.

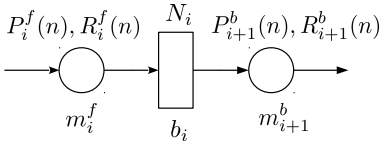


Figure 4.5: Two-machine line representation at buffer b_i for an M-machine geometric serial line

At this point, we can apply the equivalent aggregation technique described in Subsection 4.4.1 to each of these virtual two-machine lines. This results in two virtual one-machine systems for each of the $M-1$ virtual two-machine lines above (see Figure 4.6). These

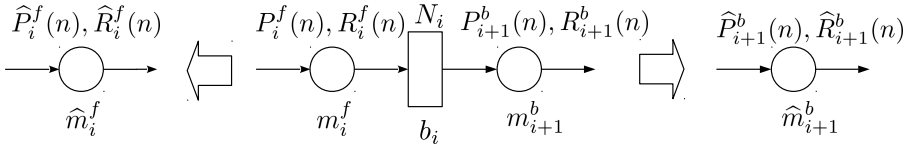


Figure 4.6: Equivalent aggregation of the virtual two-machine lines

machines ($\widehat{m}_i^f$ and $\widehat{m}_{i+1}^b$) incorporate the effects of starvation and blockage from upstream and downstream, and represent the parts flow at each single machine in the serial line. Let $\widehat{P}_i^f(n)$, $\widehat{R}_i^f(n)$, $\widehat{P}_{i+1}^b(n)$, $\widehat{R}_{i+1}^b(n)$ denote the breakdown and repair probabilities of the one-machine systems. Then, it can be shown based on the aggregation procedure developed by [43] that

$$\widehat{P}_i^f(n) = \widehat{P}_i^b(n), \quad \widehat{R}_i^f(n) = \widehat{R}_i^b(n), \quad i = 2, \ldots, M-1. \tag{4.33}$$

In other words, virtual machines $\widehat{m}_i^f$ and $\widehat{m}_i^b$ are equivalent for $i = 2, \ldots, M-1$. To avoid confusion, we use $\widehat{m}_i$ represent either case and denote its parameters as $\widehat{P}_i(n)$, $\widehat{R}_i(n)$, $i = 2, \ldots, M-1$. For m_1 and m_M, only $\widehat{m}_1^b$ and $\widehat{m}_M^f$ exist from equivalent aggregation. To maintain a consistent notation system, these two virtual machines are denoted as $\widehat{m}_1$ and $\widehat{m}_M$, respectively, with parameters $\widehat{P}_1(n)$, $\widehat{R}_1(n)$, $\widehat{P}_M(n)$, and $\widehat{R}_M(n)$.

Now consider the resulting M virtual one-machine systems, $\widehat{m}_i$'s. Assume that each operates independently on a production run of B parts. Again, the results from Subsection 4.3.1 become applicable. Let $\widehat{\mathbf{w}}^{(i)} = [w_1^{(i)} \ \cdots \ w_S^{(i)}]^T$, $i = 1, \ldots, M$, where $S = 2B + 1$ and $w_j^{(i)}$ denotes the probability that the production run on $\widehat{m}_i$ is in state j of its corresponding Markov chain. In addition, let $\widehat{\mathbf{A}}^{(i)}(n)$ denote the transition probability matrix of the Markov chains at time n for $\widehat{m}_i$ to process B parts. Then,

$$\widehat{\mathbf{w}}^{(i)}(n+1) = \widehat{\mathbf{A}}^{(i)}(n)\widehat{\mathbf{w}}^{(i)}(n), \quad i = 1, \ldots, M.$$

To evaluate the performance of the original system (Figure 4.1), the following expressions are proposed:

$$\widehat{PR}(n) = [\underbrace{0 \ \cdots \ 0}_{B \text{ zeros}} \quad \underbrace{1 \ \cdots \ 1}_{B \text{ ones}} \quad 0]\widehat{\mathbf{w}}^{(M)}(n), \tag{4.34}$$

$$\widehat{CR}(n) = [\underbrace{0 \ \cdots \ 0}_{B \text{ zeros}} \quad \underbrace{1 \ \cdots \ 1}_{B \text{ ones}} \quad 0]\widehat{\mathbf{w}}^{(1)}(n), \tag{4.35}$$

$$\widehat{P}_{ct_i}(n) = \widehat{w}_{2B}^{(i)}(n), \quad i = 1, \ldots, M, \tag{4.36}$$

$$\widehat{\mu}_{CT_i} = \sum_{n=1}^{\infty} n \cdot \widehat{w}_{2B}^{(i)}(n), \quad i = 1, \ldots, M. \tag{4.37}$$

In addition, let $WIP_i^{(\infty)}(n)$ denote the expected work-in-process of the virtual two-machine line of Figure 4.5 with infinite production run $B = \infty$, which can be calculated using the formulas derived in [43]. Similar to the two-machine case, we propose to approximate the work-in-process for the original system (i.e., with finite production run size B) using:

$$\widehat{WIP}_i(n) = WIP_i^{(\infty)}(n) \cdot \left(1 - \sum_{j=1}^{n} \widehat{P}_{ct_{i+1}}(j)\right), \quad i = 1, \ldots, M-1. \tag{4.38}$$

The accuracy of the performance approximation method was justified using a set of simulation experiments. Specifically, a simulation program has been created using C++ to

estimate the true values of the performance measures. We randomly generated 100,000 lines for each $M \in \{3, 5, 10, 15, 20\}$ for a total of 500,000 lines. The parameters of the lines were selected from the same ranges as in the two-machine case. For each line, thus constructed, we ran 10,000 replications of the simulation. This results in 95% confidence intervals of less than 0.005 for $PR(n)$ and $CR(n)$; 0.05 for $WIP_i(n)$; and 0.01 for μ_{CT_i}. In addition, we evaluated the performance measures of each line using the aggregation-based approximation method (4.34)-(4.38), and compared the results with those obtained by simulations. The relative errors between the two methods were calculated and summarized in Tables 4.2-4.5. As one can see, the approximation errors of CT and $PR(n)$ grow as a function of M, while those of $CR(n)$ and $WIP(n)$ appear to be stationary with respect to M. Although the approximation errors are greater than the two-machine case and also greater than the Bernoulli line case (see [43]), the values are still viewed as accurate enough comparing to the 5%-10% precision error contained in the machine and buffer parameters of a production line in practice.

Table 4.2: Approximation error of μ_{CT_i} in M-machine lines

M	Mean	Median	Std	90th pctl	99th pctl
3	0.91%	0.44%	1.33%	2.27%	6.66%
5	1.88%	1.18%	2.06%	4.52%	9.72%
10	3.11%	2.45%	2.63%	6.75%	11.55%
15	3.86%	3.32%	2.80%	7.72%	12.23%
20	4.45%	4.01%	2.85%	8.33%	12.64%

Table 4.3: Approximation error of $PR(n)$ in M-machine lines

M	Mean	Median	Std	90th pctl	99th pctl
3	1.44%	1.07%	1.15%	2.80%	6.00%
5	2.32%	1.77%	1.73%	4.55%	8.77%
10	3.87%	3.18%	2.10%	6.96%	9.79%
15	4.30%	3.93%	2.16%	7.28%	10.63%
20	5.14%	4.82%	2.30%	8.28%	11.65%

As an illustration of the accuracy of the approximation method, consider a ten-machine line with machine and buffer parameters randomly generated as shown in Table 4.6. The

Table 4.4: Approximation error of $CR(n)$ in M-machine lines

M	Mean	Median	Std	90th pctl	99th pctl
3	1.16%	0.85%	1.04%	2.27%	5.32%
5	1.64%	1.23%	1.43%	3.28%	7.35%
10	1.76%	1.29%	1.58%	3.66%	7.88%
15	1.40%	1.04%	1.24%	2.90%	6.17%
20	1.30%	0.96%	1.16%	2.70%	5.75%

Table 4.5: Approximation error of $WIP(n)$ in M-machine lines

M	Mean	Median	Std	90th pctl	99th pctl
3	1.44%	1.11%	1.08%	2.90%	5.16%
5	1.71%	1.50%	0.97%	2.98%	4.97%
10	1.87%	1.68%	0.92%	3.08%	4.84%
15	1.60%	1.49%	0.74%	2.61%	3.80%
20	1.57%	1.48%	0.71%	2.52%	3.53%

production run volume is 80 parts and the completion times at each machine obtained by simulation and the approximation method are given in Table 4.7, along with the relative errors between them. The transients of $PR(n)$, $CR(n)$ and $WIP_i(n)$, obtained by simulation and the approximation method, are plotted in Figure 4.7 as functions of time n. As one can see, the proposed method can approximate the transients of the consumption rate with higher accuracy than those of the production rate. This phenomenon is commonly observed among numerous cases and can be seen by comparing the corresponding entries in Table 4.3 and Table 4.4 as well. One possible reason can be attributed to the empty buffers when the production run begins, which leads to less variability for parts entering the system. In addition, the approximation accuracy for $PR(n)$ is lower near the end of the production run, which is also observed in many cases studied. More detailed investigations of the approximation accuracy at different phases of a production run and effective methods to reduce such errors will be carried out in future work.

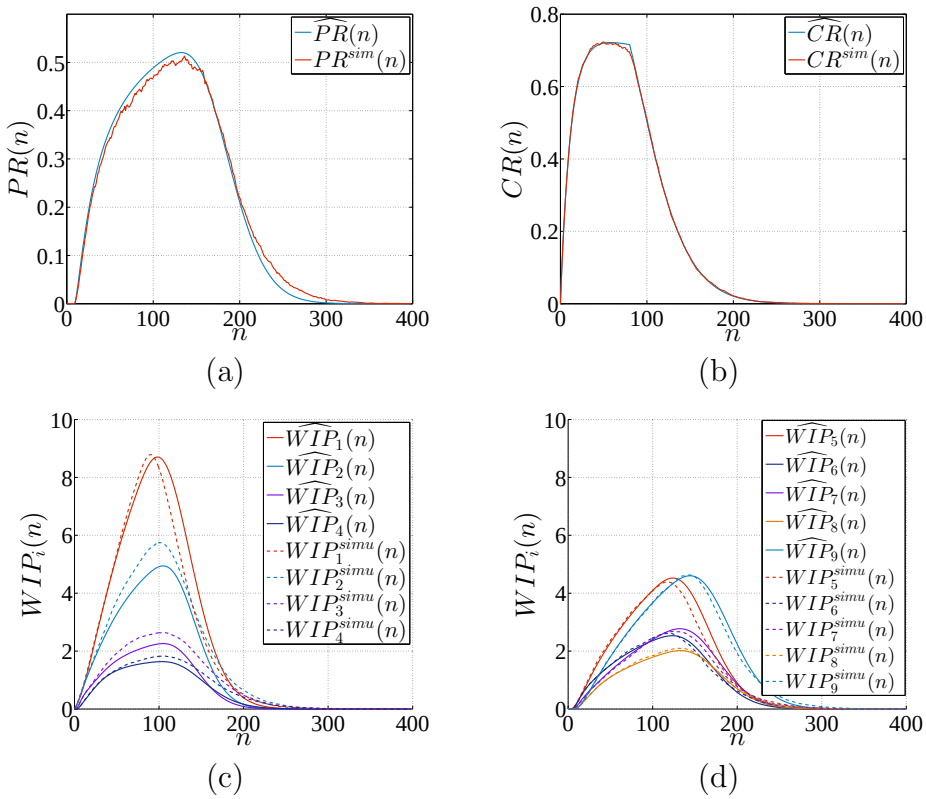


Figure 4.7: Transient performance of a ten-machine geometric line with a production run of 80 parts

Table 4.6: System parameters

i	1	2	3	4	5	6	7	8	9	10
P_i	0.022	0.041	0.047	0.016	0.011	0.036	0.020	0.019	0.019	0.044
R_i	0.061	0.162	0.218	0.204	0.054	0.150	0.129	0.145	0.132	0.134
N_i	25	15	8	5	42	16	22	10	22	–
$s_i(0)$	0	1	0	0	1	1	0	0	1	1

Table 4.7: Completion time approximation

i	1	2	3	4	5	6	7	8	9	10
$\mu_{CT_i}^{sim}$	127.51	145.62	156.85	161.81	165.02	174.97	180.36	186.48	190.97	201.53
$\widehat{\mu}_{CT_i}$	127.07	141.36	148.26	151.69	161.12	170.34	175.47	181.35	185.60	195.42
Δ_{CT_i}	0.35%	2.93%	5.48%	6.26%	2.36%	2.64%	2.71%	2.75%	2.81%	3.03%

4.5 System-theoretic Properties

While the properties of geometric serial lines during *steady state* have been investigated, the *transient behavior* has not been systematically discussed. In this section, we study some of the most important system properties in the framework of geometric serial lines completing finite production runs.

4.5.1 Stationary completion time

As noted by [43, 92], the transients of a serial line with geometric machines can be attributed to three sources: the transients of individual machines from their initial states, the transients of the buffers from initial occupancy, and their coupled interactions. When working with a production run, the initial occupancy of all buffers is fixed (empty) but the initial states of the machines are still variables. To "filter out" the effects of machine initial states on the production run completion and to focus only on the effects caused by the machines' breakdown and repair probabilities and buffer capacity, we introduce below the definition of *stationary completion time*. Specifically, let $\mu_{CT_i}(\mathbf{s}(0))$ denote a production run's completion time at machine m_i under initial machine state $\mathbf{s}(0) = [s_1(0)\ s_2(0) \dots s_M(0)]$. For geometric machine m_i, if it is in steady state, then we have

$$P[s_i(0) = \xi_i] = e_i^{\xi_i}(1 - e_i)^{1-\xi_i}, \quad \xi_i \in \{0, 1\}, \tag{4.39}$$

and, thus, for $\boldsymbol{\xi} = [\xi_1\ \xi_2 \dots \xi_M]$ and $\xi_i \in \{0, 1\}$,

$$P[\mathbf{s}(0) = \boldsymbol{\xi}] = \prod_{i=1}^{M} e_i^{\xi_i}(1 - e_i)^{1-\xi_i}, \tag{4.40}$$

Next, define *stationary completion time* as

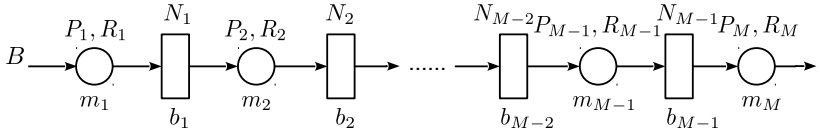
$$\bar{\mu}_{CT_i} = \sum_{\boldsymbol{\xi} \in \{0,1\}^M} \mu_{CT_i}(\mathbf{s}(0) = \boldsymbol{\xi}) P[\mathbf{s}(0) = \boldsymbol{\xi}]. \tag{4.41}$$

Clearly, $\bar{\mu}_{CT_i}$ can be viewed as the *average* of the production run's completion time across all possible machine initial states.

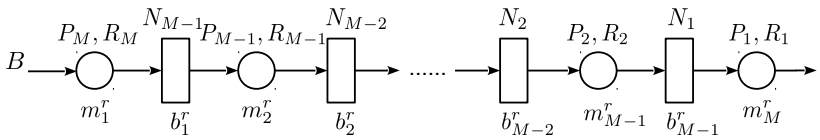
In the subsequent part of this section, we will discuss the properties of $\bar{\mu}_{CT_i}$ as a function of machine and buffer parameters P_i, R_i, and N_i.

4.5.2 Reversibility and monotonicity

Reversibility: Consider a geometric line defined by assumptions (i)-(viii) and its reverse (see Figure 4.8). Then, the stationary completion times of the original line (with superscript



(a) Original line L



(b) Reversed line L_r

Figure 4.8: Geometric serial line and its reverse

L) and the reversed line (with superscript L_r) are *practically always* equivalent, i.e.,

$$\bar{\mu}^{L}_{CT_M} \approx \bar{\mu}^{L_r}_{CT_M}. \tag{4.42}$$

Justification: The justification of the reversibility property was carried out using the same 600,000 lines generated in the previous section. Stationary completion times for each line and its reverse were calculated. The average relative error between $\bar{\mu}^{L}_{CT_M}$ and $\bar{\mu}^{L_r}_{CT_M}$ is less than 0.15% and the 90th percentile is less than 0.5%. Therefore, we claim that

the reversibility property holds. Note that, since the property is justified using numerical experiments instead of analyitcal proof, we use the term "practically always" in the statement of this and all subsequent properties studied.

Practical implications: The reversibility property implies that it takes, on average, the same amount of time for the same production run to exit the *last* machine in both the original and the reversed production lines. Meanwhile, the behavior of other performance measures, (e.g., average work-in-process, and production completion times at other machines) may be different when the line is reversed. In general, when better (e.g., more reliable) machines are placed towards to front of the line, the completion times at machines other than the last one can be reduced due to less variability, and, thus, allow longer preparation time for the next production run in practice. This placement, however, may lead to larger work-in-process during the production process. Similar observations have been made for serial lines with Bernoulli machine as well (see [81]).

Monotonicity: Consider a geometric line defined by assumptions (i)-(viii). Then, its stationary completion time $\bar{\mu}_{CT_i}$, $i = 1, \ldots, M$, is *practically always*

- monotonically decreasing in N_j and R_j, $j = 1, \ldots, M$, and
- monotonically increasing in P_j, $j = 1, \ldots, M$.

Justification: The justification of the monotonicity property was carried out using the same 600,000 lines generated in the previous section. Stationary completion time of each line was evaluated with the machine and buffer parameters changed, one at a time. As a result, no counterexample were found. Therefore, we claim that the monotonicity property holds.

Practical implications: For serial lines with Bernoulli machines, [81] show that the completion time is monotonically increasing in machine efficiency and buffer capacity as well. In the case of geometric lines, the above property indicates that increasing the machines' uptime, enlarging buffer capacity, and/or decreasing the machines' downtime always lead to

shortened average completion time for every machine in the production line. On the other hand, it should be noted that the above result does not address the monotonic property of $\bar{\mu}_{CT_i}$ with respect to the efficiency of the machines, e_i. Indeed, since e_i depends on both P_i and R_i in that $e_i = T_{up,i}/(T_{up,i} + T_{down,i}) = R_i/(R_i + P_i)$, changing e_i cannot uniquely determine the corresponding changes in P_i and R_i. As a matter of fact, even for fixed e_i, different combinations of P_i and R_i may lead to different system performance. This is discussed next.

4.5.3 Effects of up- and downtime

Consider an M-machine serial line defined by assumptions (i)-(viii). Assume the parameters of one of the machines, m_{i_0}, are modified such that the machine's efficiency remains the same, while its breakdown and repair probabilities are modified simultaneously by the same factor $k > 0$ (see Table 4.8). Clearly, the efficiency of the machine remains the same, while the machine's average up- and downtime will either increase (if $k < 1$) or decrease (if $k > 1$).

Table 4.8: Parameters of machine m_{i_0}

	Breakdown prob.	Repair prob.	Avg. uptime	Avg. downtime	Efficiency
Before	P_{i_0}	R_{i_0}	$1/P_{i_0}$	$1/R_{i_0}$	$\frac{R_{i_0}}{P_{i_0}+R_{i_0}}$
After	kP_{i_0}	kR_{i_0}	$1/(kP_{i_0})$	$1/(kR_{i_0})$	$\frac{kR_{i_0}}{kP_{i_0}+kR_{i_0}}$

Property 1 *Let $\bar{\mu}_{CT_M}^{before}$ and $\bar{\mu}_{CT_M}^{after}$ denote the stationary completion time of the serial line before and after the parameter modification shown in Table 4.8. Then,* practically always*,*

- $\bar{\mu}_{CT_M}^{before} > \bar{\mu}_{CT_M}^{after}$, *if* $k > 1$*;*
- $\bar{\mu}_{CT_M}^{before} < \bar{\mu}_{CT_M}^{after}$, *if* $k < 1$.

Justification: The justification of this property was carried out using the same 600,000 lines generated in the previous section. Without loss of generality, for each line, we examined the property for each machine by randomly selecting k for $(0, 1)$. The resulting stationary

completion time was evaluated and compared with the one of the original line. As a result, no counterexample to this property were found. Therefore, we claim that the Property 1 holds.

Practical implication: This property implies that when the efficiency of a machine is fixed, the ones with *shorter up- and downtimes* are preferred to reduce the completion time of a completion time. This property is also observed in steady state operation as shorter up- and downtime can lead to larger steady state production rate, PR_{ss}. This is mainly due to the finite "protection" capacity offered by the buffers, as longer downtime requires larger buffers to avoid machine starvations and blockages.

As an illustration, consider a five-machine geometric serial line. Assume $e_i = 0.8$, $R_i = 0.1$, $i \in \{1, 2, 4, 5\}$, and all buffers have identical capacity, $N_i = 10$. For this system, we increase the average downtime of machine m_3 from 3 to 20, while fixing its efficiency at $e_3 = 0.8$. The resulting stationary completion time of the system for a production run of $B = 50$ parts as a function of $T_{down,3}$ is given in Figure 4.9. As one can see, the stationary completion time increases with $T_{down,3}$ (almost linearly) as the above property suggests.

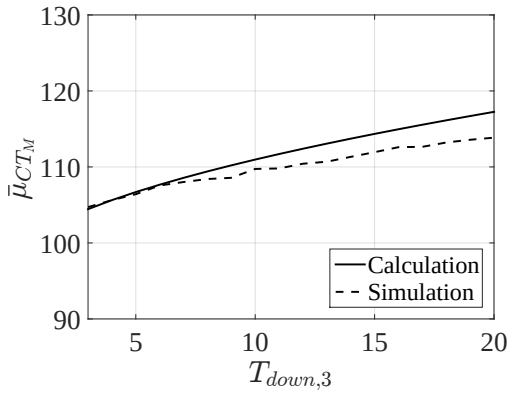


Figure 4.9: Stationary completion time vs. T_{down} while fixing machine efficiency

On the other hand, however, one may argue that the disadvantage of machine with longer up- and downtimes may be leveraged by using larger buffer capacities. This aspect of the system property is addressed next.

Definition 1 *Consider two M-machine serial lines defined by assumptions (i)-(viii):* L_1

and L_2. Let P_{i,L_j}, R_{i,L_j} denote the breakdown and repair probabilities of the i-th machine in line L_j, respectively, and let N_{i,L_j} denote the capacity of the i-th buffer in line L_j. Then, serial lines L_1 and L_2 are called steady state similar (SSS) *if*

$$\frac{P_{i,L_1}}{P_{i,L_2}} = \frac{R_{j,L_1}}{R_{j,L_2}} = \frac{N_{l,L_2}}{N_{l,L_1}} = k, \quad \forall i,j \in \{1,\ldots,M\},\ l \in \{1,\ldots,M-1\}. \tag{4.43}$$

In other words, the up- and downtimes of the machines in a line are proportional to the corresponding ones in the other line. Moreover, it can be shown that two SSS lines L_1 and L_2 have identical *steady state* production rates, i.e., $PR_{ss}^{L_1} = PR_{ss}^{L_2}$. Let $\bar{\mu}_{CT_M}^{L_j}$ denote the stationary completion time of the same production run at the M-th machine in line L_j, $j = 1, 2$.

Property 2 *Consider two SSS geometric lines defined above. Then,* practically always*,*

- $\bar{\mu}_{CT_M}^{L_1} < \bar{\mu}_{CT_M}^{L_2}$, *if* $k > 1$*;*
- $\bar{\mu}_{CT_M}^{L_1} > \bar{\mu}_{CT_M}^{L_2}$, *if* $k < 1$.

Justification: The justification of this property was carried out using the same 600,000 lines generated in the previous section as L_1's. Without loss of generality, for each line, we examined the property for generating an L_2 line by randomly selecting k from $(0, 1)$. The resulting stationary completion time was evaluated and compared with the one of the original line. As a result, no counterexample to this property were found. Therefore, we claim that the Property 2 holds.

Practical implication: This property implies that even when larger buffers are used to accommodate longer downtimes and to maintain same performance level during steady state, the ones with *shorter up- and downtimes* are still preferred to reduce the completion time of a production run. This is due to the fact that a production run starts with all buffers being empty and that systems with shorter up- and downtime machines and smaller buffers reach

steady state from empty buffers faster than systems with longer up- and downtime machines and larger buffers.

As an illustration, consider a five-machine geometric serial line, in which all machines have identical parameters, i.e., $P_i =: P$, $R_i =: R$, and all buffers have identical capacity, $N_i =: N$. For this system, we increase T_{down} of all machines from 3 to 20, while fixing the efficiency of the machines at $e = R/(P + R) = 0.8$ and the buffer capacity equal to one downtime, i.e., $N = T_{down} = 1/R$. Clearly, while T_{down} changes, the systems remain SSS since the machine efficiency remains fixed and the buffer capacity changes in proportion with T_{down}. The resulting stationary completion time of the system for a production run of $B = 50$ parts as a function of T_{down} is given in Figure 4.10. As one can see, the stationary completion time increases with T_{down} (almost linearly).

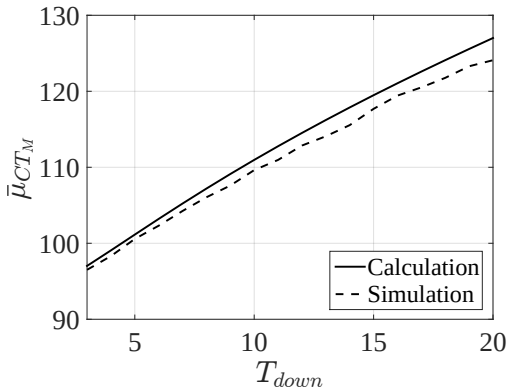


Figure 4.10: Stationary completion time vs. T_{down} for SSS lines

4.6 Summary

In this Chapter, we study the behavior of serial production lines with geometric machines and finite buffers, completing a production run of a certain number of parts. Exact closed-form expressions are derived based on Markovian analysis to calculate the performance measures for one- and two-machine systems. For systems with multiple machines, computationally efficient algorithms are developed based on equivalent aggregation to ap-

proximate the system performance measures with high accuracy. In addition, numerical experiments show that if the machines are all in steady state when the system starts operation, then the average completion times at all machines are monotonic functions of machine and buffer parameters. Moreover, the average completion time at the last machine remains the same when the parts flow is reversed. Finally, it is shown that machines with shorter up- and downtimes tend to result in shorter average completion time due to better protection provided by the buffers and faster transients.

Chapter 5

TRANSIENT PERFORMANCE ANALYSIS OF CLOSED PRODUCTION LINES WITH BERNOULLI MACHINES, FINITE BUFFERS AND CARRIERS

5.1 Introduction

In many production systems, the products usually undergo a number of processing steps, arranged in a serial manner, before entering the next stage (e.g., warehouse, shipping). While the actual physical layout of the system may take different patterns (e.g., L, U or S shapes) for saving space and convenient monitoring, they all can be viewed as serial lines. In addition, to allow easier material handling, many production lines are equipped with dedicated *carriers* that are used to transport the intermediate products in the production process. Specifically, a raw part entering the system is first loaded on a carrier (sometimes referred to as pallet, skid, etc.) at the input of the first machine, and then transported on this carrier to all subsequent machines and buffers. Finally, upon completion of the last operation, the finished part is unloaded, while the associated carrier is released and sent back to the first machine to pick up another incoming raw part. Since, in this situation, the number of parts in the system is bounded by the number of available carriers, such serial lines are called *closed with respect to carriers* (or just *closed*).

Performance evaluation of closed production lines have been discussed in the literature for decades (see a review article [93]). Specifically, asymptotic reliable two-machine, two-buffer closed lines are studied in [32] by converting the closed line into an equivalent open serial line. A case study at an automotive paint shop is described to demonstrate the efficacy of the approach. In addition, paper [33] presents a decomposition approach to approximate the system production rate for homogeneous closed production lines. The decomposition approach is extended in [34] to closed lines with geometric machines by introducing a virtual failure mode to account for the blocking or starving effect due to downstream and upstream machines. However, due to low computational efficiency, such a method is only applicable to small systems. Following this work, algorithms are developed in [94] and [95] for approximate performance calculation of larger closed lines. In addition, throughput approximation methods for finite buffered closed production systems with unreliable machines and exponentially distributed processing times are developed in [96]. An approximation analytical method for evaluating the average values of throughput and buffer levels of closed lines with three machines are proposed in [97]. An application study of closed loop system analysis and improvement at an automotive body shop is introduced in [98].

Moreover, various system-theoretical properties of closed production lines have also been investigated. For example, the effects of the number of carriers on closed line throughput is discussed in [99], which develops an empirical formula to calculate the optimal number of carriers.

It should be noted that, the results reported in the current literature are only applicable to steady state operations of closed lines. On the other hand, the recent advancement in smart manufacturing has generated great demand to study the transient and dynamic behavior of production systems, which is of critical importance in developing real-time production control algorithms. At present, most of the results on production systems transients are for open serial lines and assembly systems (see, for instance, [81,100,101]), while transient behavior of closed lines has not been studied. Thus, this paper is contributed to this end. Specifically, we

look at closed production lines with finite buffers and machines having Bernoulli reliability model, and derive analytical formulas and algorithms to calculate the transient evolution of the performance measures of the production line.

In this Chapter, we study transient performance of closed serial lines with machines having the Bernoulli reliability model. Specifically, exact mathematical model for the system considered is derived based on Markov analysis. Then, formulas for calculating the system's performance measures during transients are obtained based on the model. Finally, an approximation method is proposed to estimate a closed Bernoulli line's performance in completing a finite production run.

The remainder of the Chapter is organized as follows: Section 5.2 introduces the model and defines the performance measures for the systems under consideration. In Section 5.3, we derive exact formulas to calculate the transient performance measures of closed Bernoulli lines under given initial conditions, using Markovian analysis. Then, an approximation method for performance evaluation of closed Bernoulli lines completing a finite production run is proposed in Section 5.4. The accuracy of this method is justified using numerical simulations.

5.2 Model and Performance Measures

5.2.1 Model

Consider a closed-loop production line in Figure 5.1 defined by the following assumptions:

(i) The system consists of M machines (represented by circles), $M-1$ in-process buffers and one carriers buffer (represented by rectangles). The arrows indicate the direction of carriers flow.

(ii) The machines, m_i, $i = 1, \ldots, M$, have constant and identical cycle time τ. The time

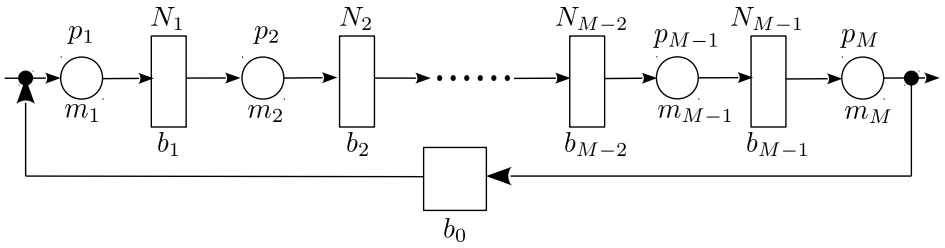


Figure 5.1: Bernoulli closed production line

axis is slotted with duration τ.

(iii) Each in-process buffer, b_i, $i = 1, \ldots, M-1$, is characterized by its capacity, $0 < N_i < \infty$.

(iv) The parts are transported within the system on carriers, i.e., the parts are placed on carriers at the input of machine m_1, and m_1 is starved for carriers when b_0 is empty; the parts are removed from carriers at the output of machine m_M, and m_M is blocked when b_0 is full and m_1 is down or blocked. The total number of carriers is S and the capacity of the empty carrier buffer, b_0, is N_0.

(v) The machines obey the Bernoulli reliability model, i.e., machine m_i, $i = 1, \ldots, M$, has two reliability status: *up* with probability p_i and *down* with $1 - p_i$. Parameter p_i is called the *efficiency* of m_i.

(vi) Machine m_i, $i = 1, \ldots, M$ is starved during a time slot if it is *up* and buffer b_{i-1} is empty at the beginning of the time slot.

(vii) Machine m_i, $i = 1, \ldots, M-1$, is blocked during a time slot if it is *up*, buffer b_i has N_i parts at the beginning of the time slot and machine m_{i+1} fails to take a part during that time slot. Machine m_M is blocked if buffer b_0 is full and m_1 fails to take a part during the time slot.

(viii) The system is assumed as operating under a finite production run-based regime with the run size equal to B [81]. All the machines are started up at the same time at the beginning of the production and each machine stops operating as soon as it has

finished processing all B parts.

(ix) Assume $N_0 \geqslant S$. All the empty carriers are in the carrier buffer N_0 initially. This is practical in factory since carriers buffer is usually large enough to hold all the carriers at the beginning and hence there is no blockage for the last machine but still starvation for the first machine.

5.2.2 Performance measures

In this paper, the performance measures of interest are:

Production Rate, $PR(n)$ = the expected number of finished parts produced by m_M in time slot $n+1$;

Consumption Rate, $CR(n)$ = the expected number of raw parts consumed by m_1 in time slot $n+1$;

Work-in-process of buffer b_i, $WIP_i(n)$ = the expected number of parts in buffer b_i at the beginning of time slot $n+1$;

Machine Starvation, $ST_i(n)$ = the probability that the machine m_i is starved in time slot $n+1$;

Machine Blockage, $BL_i(n)$ = the probability that the machine m_i is blocked in time slot $n+1$.

Note that, WIP_0 denotes the expectation of the free carriers number.

Since the Bernoulli random variable is memoryless, the closed production line modeled by under assumptions (i)-(vii) is characterized by a homogeneous Markov chain with the state being the number of parts/carriers in the in-process buffers and the carrier return buffer. In the subsequent sections, we present exact and approximate methods to calculate these performance measures in transient regime.

5.3 Exact Transient Performance Evaluation of Closed Production Lines

Transient performance evaluation of a production system amounts to calculating the evolution of the system's performance measures from a given initial condition. In this section, we derive exact formulas to accomplish this.

Consider a closed production line defined by assumptions (i)-(vii). Let $h_i(n)$, $i = 0, \cdots, M-1$, denote the number of parts/carriers in buffer b_i at the end of time slot n. The state of the Markov chain that characterizes the system is given by

$$\mathbf{h}(n) = [h_0(n) \quad h_1(n) \quad h_2(n) \quad \cdots \quad h_{M-1}(n)],$$
$$h_i(n) \in \{0,\ 1,\ \cdots,\ N_i\}, \quad i = 0,\ 1,\ \cdots,\ M-1.$$

Then, according to the operation of the system defined by assumptions (i)-(vii), the dynamics of the production system are given by:

$$\begin{aligned} h_i(n+1) &= h'_i(n+1) + \beta_i(n+1) \cdot \gamma_i(n+1) \cdot \min\{h_{i-1}(n), N_i - h'_i(n+1), 1\}, \\ &\qquad\qquad i = 1, 2, \cdots, M-1, \\ h_0(n+1) &= h'_0(n+1) + \beta_M(n+1) \cdot \gamma_M(n+1) \cdot \min\{h_{M-1}(n), 1\}, \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} h'_{M-1}(n+1) &= h_{M-1}(n) - \beta_M(n+1) \cdot \gamma_M(n+1) \cdot \min\{h_{M-1}(n), 1\}, \\ h'_i(n+1) &= h_i(n) - \beta_{i+1}(n+1) \cdot \gamma_i(n+1) \cdot \min\{h_i(n), N_{i+1} - h'_{i+1}(n+1), 1\}, \\ &\qquad\qquad i = M-2, \cdots, 1, \\ h'_0(n+1) &= h_0(n) - \beta_1(n+1) \cdot \gamma_1(n+1) \cdot \min\{h_0(n), N_1 - h'_1(n+1), 1\}, \end{aligned}$$

$$\beta_i(n) \in \begin{cases} 1, \text{ if } m_i \text{ is } up, \\ 0, \text{ if } m_i \text{ is } down, \end{cases} \qquad i \in \{1, 2, \cdots, M\},$$

$$\gamma_i(n) = \begin{cases} 1, \text{ if } N_i - h_i(n-1) > 0, \\ 1, \text{ if } N_i - h_i(n-1) = 0, \qquad \text{and } \beta_{i+1}(n) = 1,\ \gamma_{i+1}(n) = 1, \\ 0, \text{ otherwise}, \end{cases}$$

$$i \in \{1, \cdots, M-1\},$$

$$\gamma_M(n) = \begin{cases} 1, \text{ if } N_0 - h_0(n-1) > 0, \\ 1, \text{ if } N_0 - h_0(n-1) = 0, \qquad \text{and } \beta_1(n) = 1,\ \gamma_1(n) = 1, \\ 0, \text{ otherwise.} \end{cases}$$

Clearly, the maximum number of states of this Markov chain is $S = \prod_{i=0}^{M-1}(N_i + 1)$. To calculate the transition probability matrix of this Markov chain, we first arrange all the system states based on the buffer occupancy, as illustrated in Table 5.1. Then, given

Table 5.1: Arrangement of the system states

State	h_0	h_1	$\cdots$	h_{M-2}	h_{M-1}
1	0	0	$\cdots$	0	0
2	0	0	$\cdots$	0	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$N_{M-1}+1$	0	0	$\cdots$	0	N_{M-1}
$N_{M-1}+2$	0	0	$\cdots$	1	0
$N_{M-1}+3$	0	0	$\cdots$	1	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$S-1$	N_0	N_1-1	$\cdots$	N_{M-2}	$N_{M-1}-1$
S	N_0	N_1	$\cdots$	N_{M-2}	N_{M-1}

any buffer state $\mathbf{h} = [h_0\ h_1\ \cdots\ h_{M-1}]$, its corresponding state number under the above arrangement is given by:

$$\alpha(\mathbf{h}) = \sum_{i=0}^{M-1} h_i \xi_{i+1} + 1, \tag{5.2}$$

where

$$\xi_i = \begin{cases} \prod_{j=i}^{M-1}(N_j+1), & i = 0, ..., M-1, \\ 1, & i = M. \end{cases} \tag{5.3}$$

In addition, let vector $\mathbf{Q}_\alpha$ represent the buffer state $\mathbf{h}$ that corresponds to state number α, and let $Q_{\alpha,i}$ denote the occupancy of buffer b_i under this state. For example, if $M = 3, N_0 = 3, N_1 = 5, N_2 = 4$, then buffer occupancies combination $\mathbf{h} = [1\ 2\ 4]$ corresponds to state number $\alpha(\mathbf{h}) = 53$. On the other hand, $\mathbf{Q}_{53}$ represents system state with buffer occupancy $\mathbf{h} = [1\ 2\ 4]$, i.e., $Q_{53,0} = 1$, $Q_{53,1} = 2$, $Q_{53,2} = 4$.

It should be noted that many of the states in Table 5.1 are unreachable due to fixed carrier number. Indeed, all reachable states in this Markov chain should satisfy $\sum_{k=0}^{M-1} h_k = C$, i.e., the total number of parts/carriers in all buffers is exactly equal to the total carrier number C. Thus, we can construct a state space of this Markov chain with only the reachable states based on the above constraint. The procedure to obtain this reduced state space is given in Table 5.2, where $\widetilde{S}$ is the total number of reachable states and $\mathbf{R}_j$ represents the buffer state that corresponds to each reachable state. For example, consider the same three-machine closed line above with carrier number $C = 7$. For this system, the maximum state number is $S = \prod_{i=0}^{M-1}(N_i + 1) = 120$. After applying the state reduction procedure, only $\widetilde{S} = 17$ reachable states remain. Furthermore, state $\mathbf{Q}_{53} = [1\ 2\ 4]$ in the original state space becomes $\mathbf{R}_4$ in the reduced state space.

To calculate the transition probabilities among the system states, note that for each time slot the sample space is comprised of a total of 2^M combinations of machine status. Let $s_i(n) = 0$ (down), 1 (up), denote the status of machine m_i during time slot n. Then,

$$P[s_1(n) = \zeta_1, ..., s_M(n) = \zeta_M] = \prod_{i=1}^{M} p_i^{\zeta_i}(1-p_i)^{1-\zeta_i}, \quad \zeta_i \in \{0,\ 1\}. \tag{5.4}$$

Thus, for reachable state i, $i = 1, \cdots, \widetilde{S}$, we can enumerate all 2^M combinations of machine

Table 5.2: Procedure for state space reduction

start
  Let j=1
  for i=1:S
    if $\sum_{k=0}^{M-1} Q_{i,k} = C$;
      $\mathbf{R}_j = \mathbf{Q}_i$;
      j++;
    elseif $\sum_{k=0}^{M-1} Q_{i,k} \neq C$
      continue
  end
  Let $\widetilde{S}$= j;
end

status and determine the corresponding outcome states using equation (5.1). Then, the combinations of machine status that lead to the same outcome state j, $j = 1, \cdots, \widetilde{S}$, are identified and the probabilities of these combinations are summed to obtain the transition probability from the original state i to this particular outcome state j. Repeat this procedure for all $\widetilde{S}$ states then all transition probabilities as well as the transition probability matrix of the Markov chain are obtained. Let $\mathbf{x}(n) = [x_1(n) \quad \cdots \quad x_{\widetilde{S}}(n)]^T$, where $x_j(n)$ denote the probability that the system is in state $\mathbf{R}_j$, and let $\mathbf{A}_M^{cl}$ denote the transition probability matrix after the state space reduction. Then, the evolution of system state becomes:

$$\mathbf{x}(n+1) = \mathbf{A}_M^{cl}\mathbf{x}(n), \quad \mathbf{x}(0) = [0\ 0\ ...\ x_{j_0} = 1\ 0\ ...\ 0], \tag{5.5}$$

where $\mathbf{R}_{j_0}$ represents the initial buffer state. Then, under the reduced state space, the system's transient performance measures can be calculated by

$$PR(n) = \mathbf{V}_1\mathbf{x}(n), \quad CR(n) = \mathbf{V}_2\mathbf{x}(n),$$
$$WIP_i(n) = \mathbf{V}_{3,i}\mathbf{x}(n), \quad i = 0, \cdots, M-1,$$

$$
\begin{aligned}
BL_i(n) &= \mathbf{V}_{4,i}\mathbf{x}(n), \quad i = 1, \cdots, M, \\
ST_i(n) &= \mathbf{V}_{5,i}\mathbf{x}(n), \quad i = 1, \cdots, M,
\end{aligned} \tag{5.6}
$$

where

$$
\begin{aligned}
&\mathbf{V}_1 = [v_{1,1} \quad v_{1,2} \quad \cdots \quad v_{1,\widetilde{S}}], \quad \mathbf{V}_2 = [v_{2,1} \quad \cdots \quad v_{2,\widetilde{S}}], \\
&\mathbf{V}_{3,i} = [R_{1,i} \quad R_{2,i} \quad \cdots \quad R_{\widetilde{S},i}], \quad i = 0, 1, ..., M-1, \\
&\mathbf{V}_{4,i} = [v_{4,i,1} \quad v_{4,i,2} \quad \cdots \quad v_{4,i,\widetilde{S}}], \quad i = 1, ..., M-1, \\
&\mathbf{V}_{4,M} = [v_{4,M,1} \quad v_{4,M,2} \quad \cdots \quad v_{4,M,\widetilde{S}}], \\
&\mathbf{V}_{5,i} = [v_{5,i,1} \quad v_{5,i,2} \quad \cdots \quad v_{5,i,\widetilde{S}}], \quad i = 1, ..., M,
\end{aligned}
$$

with $R_{j,i}$ being the jth element of $\mathbf{R}_i$ and

$$
\begin{aligned}
&D_1 = \arg_{\max\ u}\{\sum_{i=0}^{u} R_{j,i} = \sum_{i=0}^{u} N_i\}, \quad D_2 = \arg_{\max\ u}\{\sum_{i=2}^{u} R_{j,i} = \sum_{i=2}^{u} N_i\}, \\
&D_{3,i} = \arg_{\max\ u}\{\sum_{k=i+1}^{u} R_{j,i} = \sum_{k=i+1}^{u} N_k\}, \quad D_{4,i} = \arg_{\max\ u}\{\sum_{k=0}^{u} R_{j,i} = \sum_{k=0}^{u} N_k\}, \\
&D_5 = \arg_{\max\ u}\{\sum_{i=0}^{u} R_{j,i} = \sum_{i=0}^{u} N_i\},
\end{aligned}
$$

$$
v_{1,j} = \begin{cases}
0, & \text{if } R_{j,M-1} = 0 \\
p_M \prod_{i=1}^{D_1+1} p_i, & \text{if } R_{j,M-1} = N_0 \text{ and } D_1 < M-1, \\
\prod_{i=1}^{D_1+1} p_i, & \text{if } R_{j,M-1} = N_0 \text{ and } D_1 = M-1, \\
p_M, & \text{otherwise},
\end{cases}
$$

$$
v_{2,j} = \begin{cases}
0, & \text{if } R_{j,0} = 0, \\
\prod_{i=1}^{D_2+1} p_i, & \text{if } R_{j,0} = N_1, \\
p_1, & \text{otherwise},
\end{cases}
$$

$$v_{4,i,j} = \begin{cases} p_i(1 - \prod_{k=i+1}^{D_{3,i}+1} p_k), & \text{if } R_{j,i} = N_i,\ D_{3,i} < M-1, \\ p_i(\prod_{k=i+1}^{M} p_k)(1 - \prod_{l=1}^{D_{4,i}+1} p_l), & \text{if } R_{j,i} = N_i,\ D_{3,i} = M-1,\ D_{4,i} < i-2, \\ 1 - \prod_{i=1}^{M} p_i, & \text{if } R_{j,i} = N_i,\ D_{3,i} = M-1,\ D_{4,i} = i-2, \\ 0, & \text{otherwise,} \end{cases}$$

$$v_{4,M,j} = \begin{cases} 1 - p_M \prod_{i=1}^{D_5+1} p_i, & \text{if } R_{j,0} = N_0 \text{ and } D_5 < M-1, \\ 1 - p_M \prod_{i=1}^{D_5+1} p_i, & \text{if } R_{j,0} = N_0 \text{ and } D_5 = M-1, \\ 0, & \text{otherwise,} \end{cases}$$

$$v_{5,i,j} = \begin{cases} p_i, & \text{if } R_{j,i-1} = 0, \\ 0, & \text{otherwise.} \end{cases}$$

5.4 Decomposition-based Approximation Analysis for Closed Lines with Finite Production-runs

The previous section discusses the calculation of a closed Bernoulli line's evolution from a given initial condition. Clearly, as time n tends to infinity, the system state $\mathbf{x}(n)$ and the transient performance measures all approach their steady state values. On the other hand, if the system is commanded to produce exactly a given number of products, the steady state may never be reached. In this scenario, it is apparently of interest to understand the system's behavior/performance as a function of time, and to quantify the time needed to complete all products required. This is studied in this section.

Specifically, we assume that the closed production line is operated to complete a total of B products (referred to as a *production run* of B products). In addition, we assume that $C \leq N_0$ and all carriers reside in the return carrier buffer b_0 initially. Finally, a machine stops operating as soon as it has finished processing B products. Production run-based

manufacturing is commonly seen in manufacturing processes with products having high variety of customization. For such systems, in addition to the performance measures defined in Subsection 5.2.2, it is of clear importance to be able to calculate the completion times of a production run at the machines. Let ct_i denote the time when machine m_i completes all B parts. We denote its probability mass function as

$$P_{ct_i}(n) = P[ct_i = n], \tag{5.7}$$

and let CT_i denote its expected value. In this section, we derive an analytical algorithm to approximate CT_i and the system's transient performance measures with high accuracy.

5.4.1 Decomposition-based approximation algorithm

The behavior of a production run-based closed Bernoulli line is still defined by a Markov chain. In addition to buffer occupancies, the state of this Markov chain now contains the number of finished parts at each machine. This enlargement of the state space, however, may lead to intractable computational requirement even under the state space reduction described in Section 5.3. Thus, to overcome this problem, a decomposition-based approximation method is proposed.

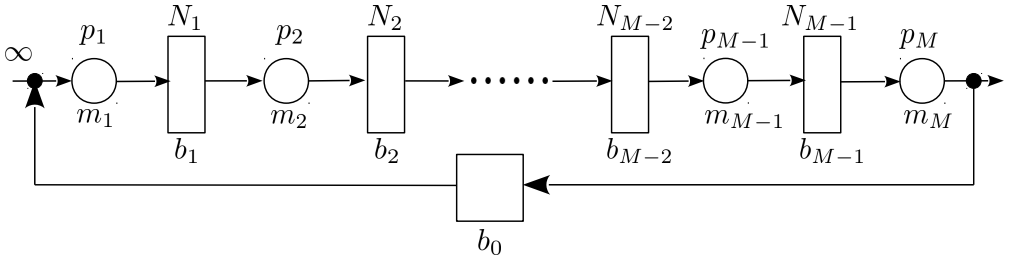
Specifically, two types of auxiliary lines are introduced to approximate the performance of an M-machine closed Bernoulli line (referred as the *original* line). The first is an M-machine closed line with the same machines and buffers as the original line except that this auxiliary line has infinite raw material at the input of m_1 (see Figure 5.2(a)). This line is constructed to capture the system dynamics caused by the circulation of carriers. Clearly, the transient performance of this auxiliary line can be analyzed using (5.5) and (5.6) under initial condition $\mathbf{R}_{\widetilde{S}} = [C\ 0\ 0\ \cdots\ 0]$.

Secondly, a set of auxiliary one-machine lines with production-run size B are constructed (see Figure 5.2(b)), where $\widehat{m}_i$ is a virtual machine in place of the machine m_i in the original

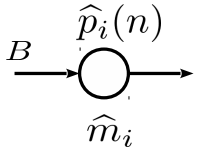
line. The efficiency of $\widehat{m}_i$ is time-varying and denoted as $\widehat{p}_i(n)$ such that

$$\widehat{p}_i(n) = p_i - BL_i^{(\infty)}(n) - ST_i^{(\infty)}(n) + BL_i^{(\infty)}(n)ST_i^{(\infty)}(n), \tag{5.8}$$

where $BL_i^{(\infty)}(n)$ and $ST_i^{(\infty)}(n)$ are the blockage and starvation probabilities of m_i at time slot n in the auxiliary M-machine line.



(a) Auxiliary M-machine line



(b) Auxiliary one-machine line

Figure 5.2: Auxiliary lines for decomposition-based approximation

To analyze the auxiliary one-machine lines, note that each one of them is defined by a Markov chain with the state being the number of parts completed by this machine (refer to [81]). Let $\mathbf{x}_f^{(i)}(n) = [x_{f,0}^{(i)}(n)\ x_{f,1}^{(i)}(n)\ ...\ x_{f,B}^{(i)}(n)]^T$, where $x_{f,j}^{(i)}(n)$ denote the probability that machine $\widehat{m}_i$ has produced j parts at the end of time slot n. The evolution of $\mathbf{x}_f^{(i)}(n)$ is then given by the following linear time-variant equation:

$$\mathbf{x}_f^{(i)}(n+1) = \mathbf{A}_f^{(i)}(n)\mathbf{x}_f^{(i)}(n), \tag{5.9}$$

with initial condition

$$\mathbf{x}_f^{(i)}(0) = [1\ 0\ \cdots\ 0\ 0]^T.$$

The time-varying transition probability matrix $\mathbf{A}_f^{(i)}(n)$ during time slot n can be calculated as follows:

$$\mathbf{A}_f^{(i)}(n) = \begin{bmatrix} 1-\widehat{p}_i(n) & & & & \\ \widehat{p}_i(n) & 1-\widehat{p}_i(n) & & & \\ & \widehat{p}_i(n) & \ddots & & \\ & & \ddots & 1-\widehat{p}_i(n) & \\ & & & \widehat{p}_i(n) & 1 \end{bmatrix}, \tag{5.10}$$

where $\widehat{p}_i(n)$ is calculated based on equation (5.8) from the auxiliary M-machine line transients.

As one can see, these one-machine lines are constructed to incorporate the dynamics of product/carrier flow through $\widehat{p}_i$'s and are operated under the same production run of the original closed line. Thus, we propose to approximate the production run completion time of the closed line based on the completion time of the auxiliary one-machine lines as follows

$$\widehat{P}_{ct_i}(n) = \mathbf{V}_{6,i}(n)\mathbf{x}_f^{(i)}(n), \quad i = 1, \cdots, M, \tag{5.11}$$

where

$$\mathbf{V}_{6,i}(n) = [\mathbf{0}_{1,B-1} \quad \widehat{p}_i(n) \quad 0], \quad i = 1, \cdots, M.$$

Finally, combining both the M-machine auxiliary line and the one-machine auxiliary lines, we propose to approximate the performance measures of the original closed line with finite production-run as follows:

$$\begin{aligned} \widehat{PR}(n) &= \mathbf{V}_1\mathbf{x}(n)(1-x_{f,B}^{(M)}), \widehat{CR}(n) = \mathbf{V}_2\mathbf{x}(n)(1-x_{f,B}^{(1)}), \\ \widehat{WIP_i}(n) &= \mathbf{V}_{3,i}\mathbf{x}(n)(1-x_{f,B}^{(i+1)}), \quad i = 0, \cdots, M-1, \\ \widehat{BL_i}(n) &= \mathbf{V}_{4,i}\mathbf{x}(n)(1-x_{f,B}^{(i)}), \quad i = 1, \cdots, M, \\ \widehat{ST_i}(n) &= \mathbf{V}_{5,i}\mathbf{x}(n)(1-x_{f,B}^{(i)}), \quad i = 1, \cdots, M, \\ \widehat{CT_i}(n) &= \sum_{n=1}^{\infty} n\widehat{P}_{ct_i}(n), \quad i = 1, \cdots, M. \end{aligned} \tag{5.12}$$

The idea of these performance approximations is to *discount* the transient performance of the line with infinite production run (calculated based on the auxiliary M-machine line using (5.5) and (5.6)) by the probability that the production run is not completed yet at the corresponding machine.

5.4.2 Accuracy of the proposed approximation method

To investigate the accuracy of the performance approximation method proposed above, numerical experiments were carried out. Specifically, we studied closed Bernoulli lines with M belonging to {3,4,5,6}. For each M, a total of 10,000 lines were generated with system parameters randomly and uniformly selected from the following sets:

$$\begin{aligned} &B \in \{20, 21, \ldots, 100\},\ p_i \in (0.7, 1),\ i = 1, ..., M, \\ &N_i \in \{3, 4, 5\},\ i = 1, ..., M-1, \quad N_0 \in \{6, 7, 8\},\ C \in \{M, M+1, ..., N_0\}. \end{aligned} \tag{5.13}$$

For each line, thus constructed, we calculated its performance measure approximations using equation (5.12). For comparison, a simulation program has been created according to the system dynamics to estimate the true values of the performance measures. To evaluate the accuracy of the proposed method, we calculate the average approximation errors for each line based on:

$$\begin{aligned} \delta_{PR} &= \frac{1}{T}\sum_{n=1}^{T}\frac{|\widehat{PR}(n) - PR_{sim}(n)|}{PR_{ss}} \cdot 100\%, \\ \delta_{CR} &= \frac{1}{T}\sum_{n=1}^{T}\frac{|\widehat{CR}(n) - CR_{sim}(n)|}{PR_{ss}} \cdot 100\%, \\ \delta_{WIP} &= \frac{\sum_{i=0}^{M-1}\sum_{n=1}^{T}\frac{|\widehat{WIP}_i(n) - WIP_i^{sim}(n)|}{N_i} \cdot 100\%}{MT}, \\ \delta_{CT} &= \frac{1}{M}\sum_{n=1}^{M}\frac{|\widehat{CT}_i - CT_i^{sim}|}{CT_i^{sim}} \cdot 100\%, \end{aligned}$$

$$\delta_{ST} = \frac{1}{MT}\sum_{i=1}^{M}\sum_{n=1}^{T}|\widehat{ST}_i(n) - ST_i^{sim}(n)|,$$
$$\delta_{BL} = \frac{1}{MT}\sum_{i=1}^{M}\sum_{n=1}^{T}|\widehat{BL}_i(n) - BL_i^{sim}(n)|, \tag{5.14}$$

where PR_{ss} is the line's steady state production rate obtained using simulation, and T is the smaller time instant such that

$$\min\left\{\sum_{n=1}^{T}\widehat{P}_{ct_M}(n), \sum_{n=1}^{T}P_{ct_M}^{sim}(n)\right\} \geq 0.999. \tag{5.15}$$

The results of the approximation error of the proposed method are summarized using box plots in Figure 5.3. It can be seen from the figure, the medians of δ_{PR}, δ_{CR} and δ_{WIP} are typically around 1-3%. The medians approximation error of the production run completion time are even smaller: below 0.4% in all cases studied. The results on δ_{BL} and δ_{ST} also show that the approximation method has good accuracy. Considering the machine and buffer parameters of a production line are rarely known in practice with accuracy better than 5%-10%, we conclude that the proposed approximation method can provide accurate performance evaluation for such systems.

As an illustration, consider the five-machine line shown in Figure 5.5, where the numbers above the machines and buffers are their efficiencies and capacities, respectively. In addition, assume that the system has a total of $C = 10$ carriers and a production run of $B = 60$ products. The transient performance of this line, evaluated by both simulation (left column) and the proposed approximation method (right column) are given in Figure 5.4. The production run completion times at each machine, evaluated by both methods, are summarized in Table 5.3. Clearly, the proposed approximation method is capable of evaluating the transient performance measures of the closed line with high accuracy. As far as computation time is concerned, it takes 24.6 seconds to obtain the simulation results using Matlab on a

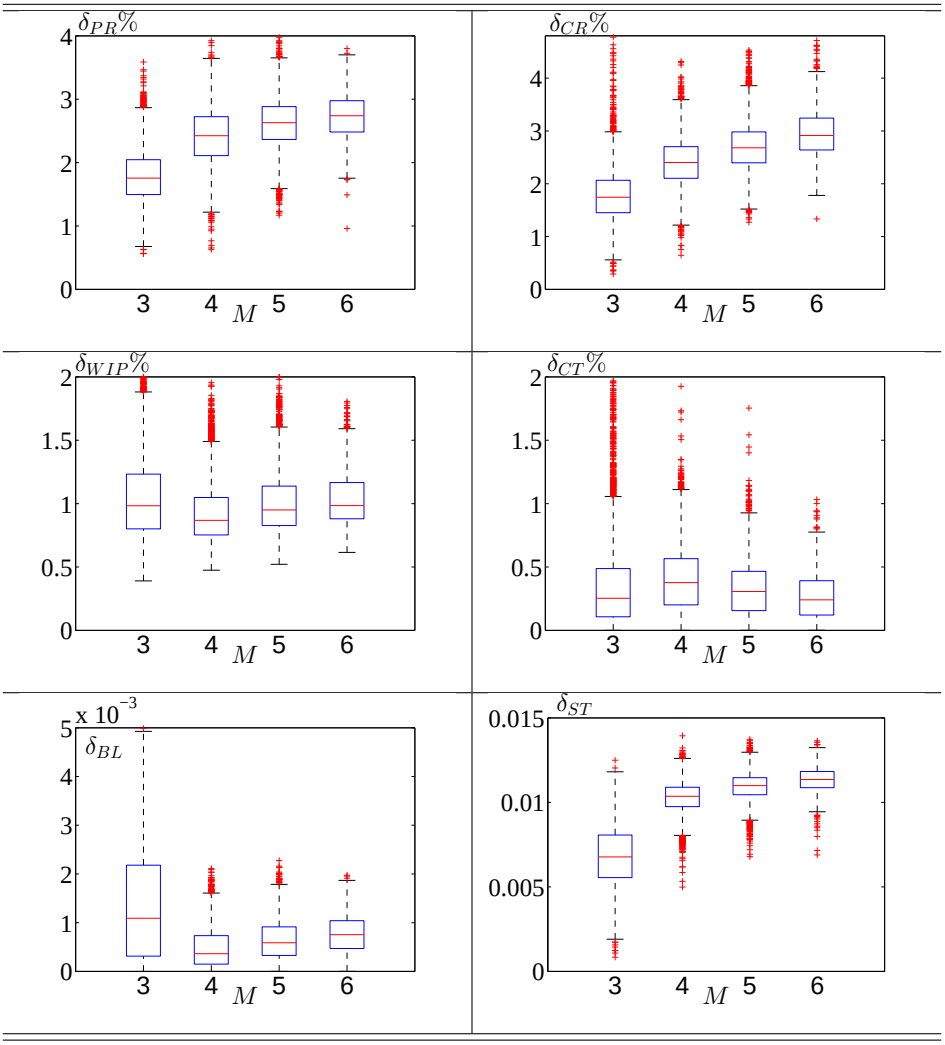


Figure 5.3: Approximation error of the proposed method

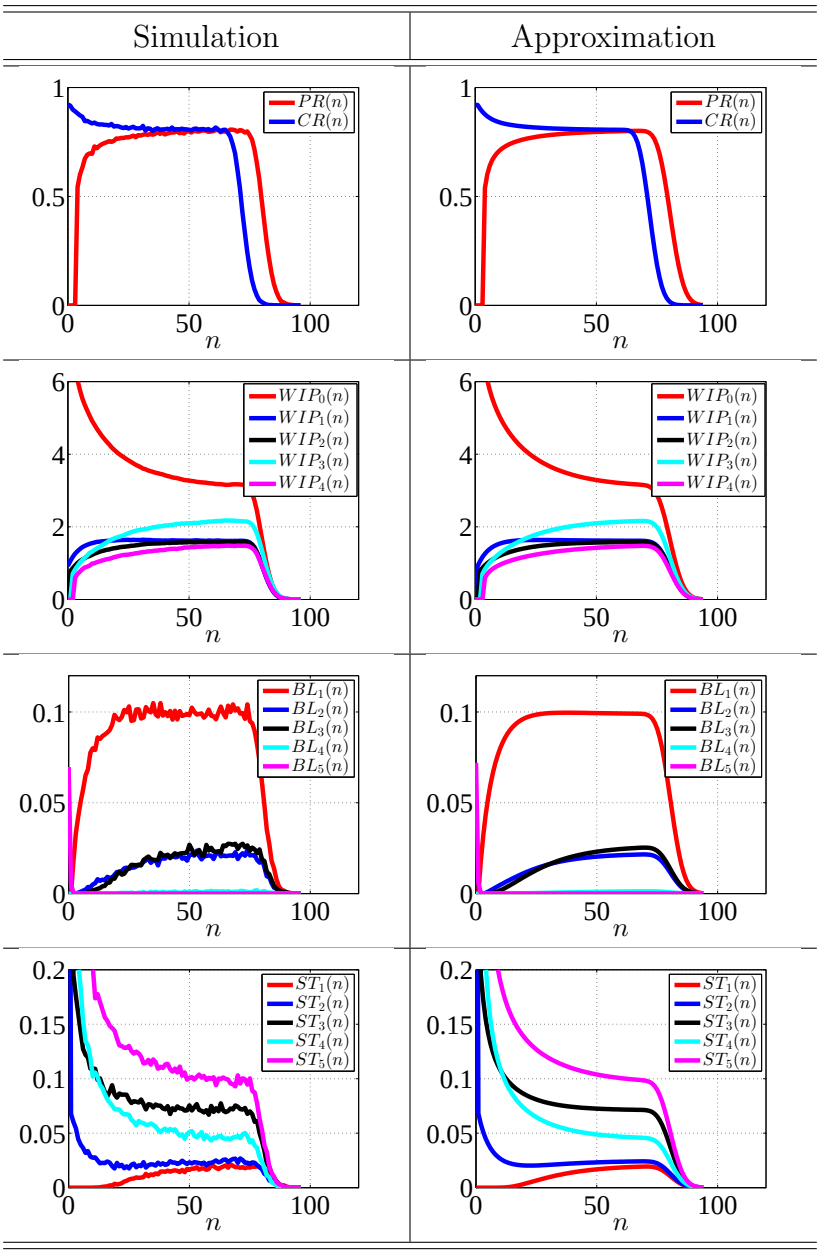


Figure 5.4: Comparison of simulation and approximation methods for transient performance evaluation

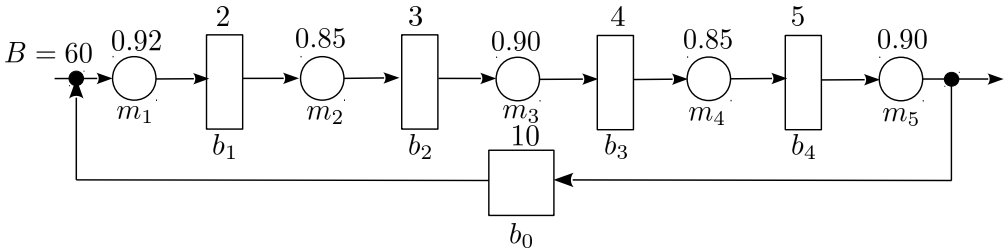


Figure 5.5: 5-machine closed Bernoulli line with finite production run size B=60

computer with Intel Core i7-4770 CPU and 8GB RAM, while the proposed approximation method takes only 9.2 seconds on the same computer.

Table 5.3: Completion time at each machine

	CT_1	CT_2	CT_3	CT_4	CT_5
Approximation	72.8	74.9	76.7	79.5	81.2
Simulation	73.0	75.0	76.9	79.6	81.5

5.5 Summary

In this Chapter, we study transient performance of closed serial lines with machines having the Bernoulli reliability model. Specifically, exact mathematical model for the system considered is derived based on Markov analysis. Then, formulas for calculating the system's performance measures during transients are obtained based on the model. Finally, an approximation method is proposed to estimate a closed Bernoulli line's performance with a finite production run.

To extend the current results in future work, machines with other reliability models (e.g., geometric or exponential) will be studied; system transient properties will be investigated and compared to those studied in steady state; from the aspects of energy-efficient control and best throughput performance, the machine operation and carriers control will also be studied.

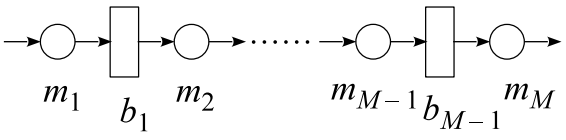
Chapter 6

PERFORMANCE ANALYSIS OF ASSEMBLY SYSTEMS WITH BERNOULLI MACHINES AND FINITE BUFFERS DURING TRANSIENT

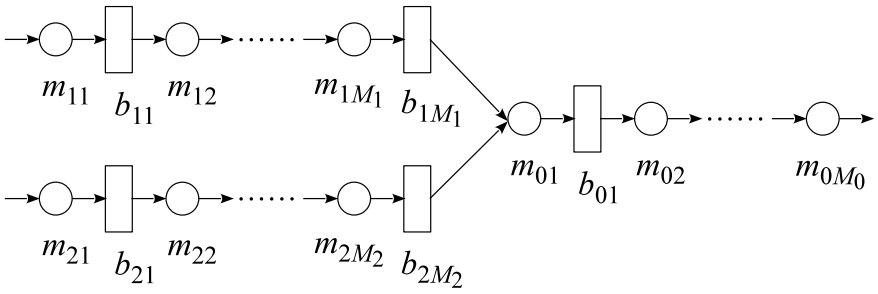
6.1 Introduction

Although practical production systems may take various physical topologies, serial lines (see Figure 6.1(a)) and assembly systems (see Figure 6.1(b)) are two of the most fundamental structures used in various manufacturing environments. In the literature, however, while serial production lines have been studied extensively, assembly systems have received much less investigations.

Early studies on assembly systems consider only multi-queue-one-server cases, where several types of parts arrive at a single server to be assembled together (see [102–104]). Inspired by these work, three-server systems with finite queue capacities have been studied in [105–107]. In these studies, two servers represent component part production, and the other server represents the assembly operation. In addition, queueing model based assembly systems have been further studied in [108–111] and the references therein. The problem of steady state performance evaluation in assembly systems with unreliable machines and finite buffers has been studied in [26–31, 84, 112, 113]. Specifically, paper [26] develops a



(a) Serial line



(b) Assembly system

Figure 6.1: Serial production line and assembly system

decomposition technique to approximate the steady state throughput for assembly systems with machines having geometric models and identical processing times, while paper [27] studies assembly systems with geometric machines and non-identical processing times by transforming the assembly system into a serial line. In addition, paper [28] extends the analysis to assembly systems with geometric processing times. The decomposition technique was later generalized to assembly systems with exponential machines and deterministic processing times in [29]. An improved version of the algorithm is proposed in [30]. Moreover, the methods developed in [29, 30] were further extended in [112] by considering machines having exponential reliability model and exponentially distributed processing times. Steady state performance evaluation, continuous improvement, and bottleneck identification in assembly systems with Bernoulli machines have been investigated in [84, 113] using recursive aggregation. Finally, throughput approximation and bottleneck identification during steady state in assembly systems with non-exponential machines have been studied in [31] based on numerical simulations.

Despite these important results regarding assembly system performance evaluation, it should be noted that most of the studies reported in the current literature focus on steady

state only, while the transients of such systems have not been systematically studied. To the best of our knowledge, analytical study of assembly system transients only appears in paper [114], which studies the transient throughput of a class of one-server Markovian assembly-like queue with infinite queueing capacity. Therefore, the goal of this Chapter is to derive analytical models that describe the transients of assembly systems with finite buffers and machines having the Bernoulli reliability model, and to develop analytical methods for their performance evaluation during transients.

6.2 Model and Performance Measures

6.2.1 Model

Consider an assembly system shown in Figure 6.1(b), where circles represent machines and rectangles represent buffers. The system is defined by the following assumptions:

(i) The end product of the system requires two components. One component (Component 1) is processed by machines m_{1i}'s, $i = 1, \ldots, M_1$, in a serial manner. We refer to this part of the system (from machine m_{11} to buffer b_{1M_1}) as *Component Line 1*. Similarly, the other component (Component 2) is processed by machines m_{2i}'s, $i = 1, \ldots, M_2$, also in a serial manner. This part of the system (from machine m_{21} to buffer b_{2M_2}) is referred to as *Component Line 2*.

(ii) Machine m_{01} assembles one finished unit from Component Line 1 and one finished unit from Component Line 2 into one product unit. The product will be further processed by all machines that follow m_{01}. This part of the system (from machine m_{01} to machine m_{0M_0}) is referred to as the *Main Product Line*.

(iii) The machines have identical and constant cycle time τ. The time axis is slotted with slot duration τ. Machines begin operating at the beginning of each time slot.

(iv) The machines obey the Bernoulli reliability model, i.e., machine m_{ij}, being neither

blocked nor starved, processes a product unit during a time slot with probability p_{ij} and fails to do so with probability $1 - p_{ij}$. Parameter $p_{ij} \in (0, 1)$ is referred to as the *efficiency* of machine m_{ij}. The status of all machines are independent.

(v) Each buffer is characterized by its capacity, $0 < N_{ij} < \infty$, i.e., the maximum number of product units that the buffer can hold.

(vi) Machine m_{ij}, $i \in \{0, 1, 2\}$, $j \in \{2, \ldots, M_i\}$, is starved during a time slot if it is up and buffer $b_{i(j-1)}$ is empty at the beginning of the time slot. Machine m_{01} is starved if it is up and either buffer b_{1M_1} or buffer b_{2M_2} is empty. Machines m_{11} and m_{21} are never starved for raw parts.

(vii) Machine m_{ij}, $i \in \{0, 1, 2\}$, $j \in \{1, \ldots, M_i - 1\}$, is blocked during a time slot if it is up, buffer b_{ij} has N_{ij} product units at the beginning of the time slot, and machine $m_{i(j+1)}$ fails to take a part (due to machine breaking down or blockage) during the time slot. Machine m_{iM_i}, $i \in \{1, 2\}$, is blocked if it is up, buffer b_{iM_i} is full and machine m_{01} fails to take a part (due to machine breaking down, blockage, or starvation for a component). Machine m_{0M_0} is never blocked.

(viii) The system operates for a total of T time slots.

Note that the above model contains only two component lines and one assembly operation. The analysis of assembly systems with multiple component lines and several assembly operations is similar. The approach is to approximate the system performance by decomposing the original assembly system into a number of interacting serial lines. The detailed discussion on complex assembly systems is provided in Section 6.5.

6.2.2 Performance measures

To carry-out rigorous real-time analysis and control of production systems, transient performance measures must be used. In the framework of Bernoulli assembly systems defined by assumptions (i)-(viii), the performance measures of interest are

- *Production rate*, $PR(n)$: the expected number of finished parts produced by m_{0M_0} in time slot $n+1$;
- *Consumption rate*, $CR_i(n)$: the expected number of raw parts consumed by machine m_{i1}, $i = 1, 2$, in time slot $n+1$;
- *Work-in-process*, $WIP_{ij}(n)$: the expected number of parts in buffer b_{ij} at the beginning of time slot $n+1$;
- *Machine starvation*, $ST_{ij}(n)$: the probability that machine m_{ij}, $i \in \{0, 1, 2\}$, $j \in \{2, \ldots, M_i\}$, is starved by buffer $b_{i(j-1)}$ in time slot $n+1$;
- *Machine blockage*, $BL_{ij}(n)$: the probability that machine m_{ij}, $i \in \{0, 1, 2\}$, $j \in \{1, \ldots, M_i\}$, is blocked by buffer b_{ij} in time slot $n+1$.

In addition, since the assembly machine m_{01} can be starved by either component line, we define its starvations as:

$$\begin{aligned} ST_{01,1}(n) &= P[m_{01} \text{ starved by buffer } b_{1M_1} \text{ in time slot } n+1], \\ ST_{01,2}(n) &= P[m_{01} \text{ starved by buffer } b_{2M_2} \text{ in time slot } n+1]. \end{aligned}$$

Moreover, since the system can only produce (consume) zero or one part during one time slot. The production rate and consumption rates can be expressed as follows:

$$\begin{aligned} PR(n) &= P[m_{0M_0} \text{ is up and not starved in time slot } n+1], \\ CR_i(n) &= P[m_{i1} \text{ is up and not blocked in time slot } n+1], \qquad i = 1, 2. \end{aligned}$$

Note that in steady state, i.e., when $n \to \infty$, the conservation law holds:

$$\lim_{n\to\infty} PR(n) = \lim_{n\to\infty} CR_1(n) = \lim_{n\to\infty} CR_2(n).$$

During transients, however, the system production rate and consumption rates may take different values. Similarly, one of the most fundamental and widely used results in queueing theory, Little's Law, also only applies to steady state.

Using a recursive aggregation procedure, a method for estimating the *steady state* values of these performance measures has been developed in [9]. In this Chapter, we develop methods to evaluate these performance measures during transients.

6.3 Mathematical Model and Exact Performance Evaluation

Since the Bernoulli reliability model is "memoryless" and the buffer capacities are finite, the assembly system defined by assumptions (i)-(viii) is characterized by an ergodic Markov chain with the states being the occupancy of the buffers. To study a Markovian system, we first need to obtain its transition probability matrix. This is accomplished as follows: Let $h_{ij}(n)$ denote the number of parts in buffer b_{ij} at the end of time slot n. Then, the state of the Markov chain is given by

$$\mathbf{h}(n) \;=\; [h_{01}(n)\ \ldots\ h_{0(M_0-1)}(n)\quad h_{11}(n)\ \ldots\ h_{1M_1}(n)\quad h_{21}(n)\ \ldots\ h_{2M_2}(n)],$$

where $h_{ij}(n) \in \{0, 1, \ldots, N_{ij}\}$. Clearly, the number of states of this Markov chain is

$$S = \left[\prod_{j=1}^{M_0-1}(N_{0j}+1)\right] \cdot \left[\prod_{j=1}^{M_1}(N_{1j}+1)\right] \cdot \left[\prod_{j=1}^{M_2}(N_{2j}+1)\right].$$

Let $s_{ij}(n) \in \{1{=}\text{up},\ 0{=}\text{down}\}$ denote the status of machine m_{ij} during time slot n. According to model assumptions (i)-(viii),

$$P\left[\bigcap_{all\ ij}\{s_{ij}(n)=\alpha_{ij}\}\right]=\left[\prod_{j=1}^{M_0}p_{0j}^{\alpha_{0j}}(1-p_{0j})^{1-\alpha_{0j}}\right]\cdot\left[\prod_{j=1}^{M_1}p_{1j}^{\alpha_{1j}}(1-p_{1j})^{1-\alpha_{1j}}\right]$$
$$\cdot\left[\prod_{j=1}^{M_2}p_{2j}^{\alpha_{2j}}(1-p_{2j})^{1-\alpha_{2j}}\right],$$
$$\alpha_{ij}\in\{0,1\},\quad i\in\{0,1,2\},\quad j\in\{1,\ldots,M_i\},$$

and there exist a total of $2^{M_0+M_1+M_2}$ different combinations of machine status. For each of these combinations of machine status, the transition of the system states (i.e., buffer occupancies) can be obtained based on assumptions (i)-(viii) as follows:

- For the *Main Product Line*, if $M_0>1$, i.e., there is at least one buffer in this part of the system:

$$h_{0j}(n) = \widetilde{h}_{0j}(n)+s_{0j}(n)\cdot\min\{h_{0(j-1)}(n-1),N_{0j}-\widetilde{h}_{0j}(n),1\}, \tag{6.1}$$
$$2\le j\le M_0-1,$$
$$h_{01}(n) = \widetilde{h}_{01}(n)+s_{01}(n)\cdot\min\{h_{1M_1}(n-1),h_{2M_2}(n-1),N_{01}-\widetilde{h}_{01}(n),1\}, \tag{6.2}$$

  where $\widetilde{h}_{0j}(n)$ represents the occupancy of buffer b_{0j} as soon as the downstream machine removes a part from b_{0j} at the beginning of time slot n and is given by:

$$\widetilde{h}_{0(M_0-1)}(n)=h_{0(M_0-1)}(n-1)-s_{0M_0}(n)\cdot\min\{h_{0(M_0-1)}(n-1),1\}, \tag{6.3}$$
$$\widetilde{h}_{0j}(n)=h_{0j}(n-1)-s_{0(j+1)}(n)\cdot\min\{h_{0j}(n-1),N_{0(j+1)}-\widetilde{h}_{0(j+1)}(n),1\}, \tag{6.4}$$
$$1\le j\le M_0-2.$$

  These equations are derived based on model assumptions (i)-(viii). To help understanding these equations, consider an internal buffer b_{0j}, $1<j<M_0-1$, as an example. As one can see from equation (6.4), a part will be removed from the buffer at the beginning of time slot n if its downstream machine is up ($s_{0(j+1)}(n)=1$), the buffer itself is not empty ($h_{0j}(n-1)\ge 1$) and its downstream buffer is not full ($N_{0(j+1)}-\widetilde{h}_{0(j+1)}(n)\ge 1$).

Then, according to (6.1), a new part will be placed in this buffer at the end of time slot n if its upstream machine is up ($s_{0j}(n) = 1$), the buffer itself is not full ($N_{0j} - \widetilde{h}_{0j}(n) \geq 1$), and its upstream buffer is not empty ($h_{0(j-1)}(n-1) \geq 1$).

If $M_0 = 1$, then the Main Product Line has no buffer, and, thus, no state transition is present in this part of the system.

- For *Component Line* i, $i \in \{1, 2\}$:

$$h_{ij}(n) = \widetilde{h}_{ij}(n) + s_{ij}(n) \cdot \min\{h_{i(j-1)}(n-1), N_{ij} - \widetilde{h}_{ij}(n), 1\}, \quad 2 \leq j \leq M_i, \tag{6.5}$$

$$h_{i1}(n) = \widetilde{h}_{i1}(n) + s_{i1}(n) \min\{N_{i1} - \widetilde{h}_{i1}(n), 1\}, \tag{6.6}$$

where

$$\widetilde{h}_{iM_i}(n) = h_{iM_i}(n) - s_{01}(n) \cdot \min\{h_{1M_1}(n-1), h_{2M_2}(n), N_{01} - \widetilde{h}_{01}(n), 1\}, \tag{6.7}$$

$$\widetilde{h}_{ij}(n) = h_{ij}(n) - s_{i(j+1)}(n) \cdot \min\{h_{ij}(n-1), N_{i(j+1)} - \widetilde{h}_{i(j+1)}(n), 1\}, \quad 1 \leq j \leq M_i - 1. \tag{6.8}$$

In the above expressions, $\widetilde{h}_{ij}(n)$ represents the occupancy of buffer b_{ij} as soon as the downstream machine removes a part from b_{ij} at the beginning of time slot n.

Therefore, for each state, we can enumerate all $2^{M_0+M_1+M_2}$ combinations of machine status and determine the corresponding outcome states using the equations above. Then, the combinations of machine status that lead to the same outcome state are grouped together and the probabilities of these combinations are added to obtain the transition probability from the original state to this particular outcome state. Repeat this procedure for all S states and all the transition probabilities of this Markov chain can be obtained. To facilitate the subsequent derivations, we linearize the multi-dimension state representation $\mathbf{h}(n)$ by arranging them from state 1 to state S based on a mixed radix numeral system with each

digit representing the occupancy of a buffer. Specifically, in this numeral system, each number has $(M_0 + M_1 + M_2 - 1)$ positions, $f_1 f_2 \ldots f_{M_0+M_1+M_2-1}$ and

$$f_i = \begin{cases} h_{0i}, & i = 1, \ldots, M_0 - 1, \\ h_{1(i-M_0+1)}, & i = M_0, \ldots, M_0 + M_1 - 1, \\ h_{2(i-M_0-M_1+1)}, & i = M_0 + M_1, \ldots, M_0 + M_1 + M_2 - 1. \end{cases} \tag{6.9}$$

In addition, the base of the i-th position, g_i, is defined as

$$g_i = \begin{cases} N_{0i}, & i = 1, \ldots, M_0 - 1, \\ N_{1(i-M_0+1)}, & i = M_0, \ldots, M_0 + M_1 - 1, \\ N_{2(i-M_0-M_1+1)}, & i = M_0 + M_1, \ldots, M_0 + M_1 + M_2 - 1. \end{cases} \tag{6.10}$$

As a result, the state number of a given $\mathbf{h}(n)$ can be calculated as follows:

$$\begin{aligned} \text{State number} &= 1 + \sum_{i=1}^{M_0+M_1+M_2-1} \left(f_i \prod_{j=i+1}^{M_0+M_1+M_2-1} g_j \right) \\ &= \left(\sum_{j=1}^{M_0-1} h_{0j}\beta_{0j} \right) + \left(\sum_{j=1}^{M_1} h_{1j}\beta_{1j} \right) + \left(\sum_{j=1}^{M_2} h_{2j}\beta_{2j} \right) + 1, \end{aligned} \tag{6.11}$$

where

$$\begin{aligned} \beta_{0j} &= \begin{cases} \left[\prod_{k=j+1}^{M_0}(N_{0k}+1)\right]\left[\prod_{i=1}^{2}\prod_{k=1}^{M_i}(N_{ik}+1)\right], & \text{for } 1 \le j \le M_0 - 2, \\ \left[\prod_{k=1}^{M_1}(N_{1k}+1)\right]\left[\prod_{k=1}^{M_2}(N_{2k}+1)\right], & \text{for } j = M_0 - 1, \end{cases} \\ \beta_{1j} &= \begin{cases} \left[\prod_{k=j+1}^{M_1}(N_{1k}+1)\right]\left[\prod_{k=1}^{M_2}(N_{2k}+1)\right], & \text{for } 1 \le j \le M_1 - 1, \\ \prod_{k=1}^{M_2}(N_{2k}+1), & \text{for } j = M_1, \end{cases} \\ \beta_{2j} &= \begin{cases} \prod_{k=j+1}^{M_2}(N_{2k}+1), & \text{for } 1 \le j \le M_2 - 1, \\ 1, & \text{for } j = M_2. \end{cases} \end{aligned} \tag{6.12}$$

$$
\begin{aligned}
h_{0j}[r] &= \left\lfloor \frac{r - \sum_{k=1}^{j-1} h_{0k}[r]\beta_{0k}}{\beta_{0j}} \right\rfloor, \\
h_{1j}[r] &= \left\lfloor \frac{r - \sum_{k=1}^{M_0-1} h_{0k}[r]\beta_{0k} - \sum_{k=1}^{j-1} h_{1k}[r]\beta_{1k}}{\beta_{1j}} \right\rfloor, \\
h_{2j}[r] &= \left\lfloor \frac{r - \sum_{k=1}^{M_0-1} h_{0k}[r]\beta_{0k} - \sum_{k=1}^{M_1} h_{1k}[r]\beta_{1k} - h_{0k}[r]\beta_{0k} - \sum_{k=1}^{j-1} h_{2k}[r]\beta_{2k}}{\beta_{2j}} \right\rfloor,
\end{aligned}
\tag{6.13}
$$

Under this arrangement, state 1 represents the system state where all buffers are empty, while state S represents the state where all buffers are full. For example, assume $M_0 = 2$, $M_1 = M_2 = 1$, and $N_{01} = 2$, $N_{11} = 3$, $N_{21} = 4$. Then, state $h_{01} = 1$, $h_{11} = 2$, $h_{21} = 3$ can be written as "123" in the above numeral system and its corresponding state number is $1 \times 4 \times 5 + 2 \times 5 + 3 + 1 = 34$.

It should be noted that there are other methods to linearize the system states. However, we choose to use the one above since it is straightforward and convenient in numerical calculations. On the other hand, by reversing (6.11) and (6.12), the corresponding buffer occupancy, $h_{ij}[r]$, for a given state number r can be calculated using (6.13), where $\lfloor a \rfloor$ represents the floor operation, i.e., the largest integer smaller than a.

Let $x_i(n)$, $i = 1, \ldots, S$, $n = 0, 1, \ldots$, denote the probability that the Markov chain is in state i at the end of time slot n and let $\mathbf{A}$ denote the transition probability matrix of the Markov chain. As mentioned above, this matrix can be obtained by enumerating all machine status combinations for each state of the Markov chain. Assume that buffer b_{ij} has $h_{ij}(0)$ parts at time 0. Then, the evolution of $\mathbf{x}(n) = [x_1(n) \ \ldots \ x_S(n)]^T$, $n = 1, 2, \ldots$, can be described by the following constrained linear time-invariant equation:

$$
\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \quad n = 0, 1, \ldots, T, \tag{6.14}
$$

with initial condition

$$\mathbf{x}(0) = [0 \quad 0 \quad \cdots \quad x_{H_0}(0) = 1 \quad \cdots \quad 0 \quad 0]^T, \tag{6.15}$$

where H_0 is the state number of initial buffer occupancy $\mathbf{h}(0)$, which can be calculated using (6.11). In Control Theory, vector $\mathbf{x}(n)$ is often referred to as the *state* of linear system (6.14). Based on the system representation above and the definitions of the transient performance measures given in Subsection 6.2.2, the closed-form formulas to calculate the performance measures can be expressed in (6.16)-(6.26), where the $\mathbf{D}$'s are row vectors with S entries given by:

$$\begin{aligned}
PR(n) &= P[m_{0M_0} \text{ up and all its upstream buffers are not empty at time slot } n] \\
&= \begin{cases} p_{0M_0} P[h_{0(M_0-1)}(n) > 0], & \text{if } M_0 > 1, \\ p_{0M_0} P[h_{1M_1}(n) > 0 \text{ and } h_{2M_2}(n) > 0], & \\ & \text{if } M_0 = 1, \end{cases} \\
&= \mathbf{D}_1 \mathbf{x}(n), && (6.16) \\
CR_i(n) &= P[m_{i1} \text{ up and not blocked in time slot } n] && (6.17) \\
&= p_{i1} - BL_{i1}(n) = \mathbf{D}_{2,i} \mathbf{x}(n), \quad i = 1, 2, && (6.18) \\
WIP_{ij}(n) &= E[h_{ij}(n)] = \sum_{k=1}^{N_{ij}} k P[h_{ij}(n) = k] = \mathbf{D}_{3,ij} \mathbf{x}(n), && (6.19) \\
ST_{ij}(n) &= P[m_{ij} \text{ up and buffer } b_{i(j-1)} \text{ empty at time slot } n] && (6.20) \\
&= p_{ij} P[h_{i(j-1)}(n) = 0] = \mathbf{D}_{4,ij} \mathbf{x}(n), \qquad i = 0, 1, 2, \quad 2 \le j \le M_i, && (6.21) \\
ST_{01,i}(n) &= P[m_{01} \text{ up and buffer } b_{iM_i} \text{ empty at time slot } n] && (6.22) \\
&= p_{01} P[h_{iM_i}(n) = 0] = \mathbf{D}_{5,i} \mathbf{x}(n), \quad i = 1, 2, && (6.23) \\
& && (6.24)
\end{aligned}$$

$$BL_{ij}(n) = P[m_{ij} \text{ up, buffer } b_{ij} \text{ full, and its downstream machine down,} \quad (6.25)$$
$$\text{blocked or starved at time slot } n]$$

$$= \begin{cases} p_{ij}\left[(1-p_{i(j+1)}) + BL_{i(j+1)}(n)\right] P[h_{ij}(n) = N_{ij}], \\ \qquad\qquad i = 0,1,2,\ 1 \le j \le M_i - 1, \\ p_{ij}\left[(1-p_{01}) + BL_{01}(n)\right] P[h_{ij}(n) = N_{ij},\ h_{i^*M_{i^*}}(n) > 0] \\ \qquad\qquad + p_{ij} P[h_{ij}(n) = N_{ij},\ h_{i^*M_{i^*}}(n) = 0],\ i = 1,2,\ j = M_i, \end{cases}$$

$$= \mathbf{D}_{6,ij}\,\mathbf{x}(n), \quad i = 0,1,2, \quad 2 \le j \le M_i. \quad (6.26)$$

$$\mathbf{D}_1 = [d_{1,1}\ d_{1,2}\ \cdots\ d_{1,S}], \quad (6.27)$$

$$d_{1,i} = \begin{cases} p_{0M_0}, & \text{if } M_0 > 1 \text{ and } h_{0(M_0-1)}[i] > 0, \\ p_{0M_0}, & \text{if } M_0 = 1,\ h_{1M_1}[i] > 0,\ h_{2M_2}[i] > 0, \\ 0, & \text{otherwise,} \end{cases} \quad (6.28)$$

$$\mathbf{D}_{2,i} = p_{i1}[1\ 1\ \cdots\ 1] - \mathbf{D}_{6,i1}, \quad i = 1,2, \quad (6.29)$$

$$\mathbf{D}_{3,ij} = [d_{3,ij,1}\ d_{3,ij,2}\ \cdots\ d_{3,ij,S}], \quad d_{3,ij,k} = h_{ij}[k], \quad (6.30)$$

$$\mathbf{D}_{4,ij} = [d_{4,ij,1}\ d_{4,ij,2}\ \cdots\ d_{4,ij,S}], \quad (6.31)$$

$$d_{4,ij,k} = \begin{cases} p_{ij}, & \text{if } h_{i(j-1)}[k] = 0, \\ 0, & \text{otherwise,} \end{cases} \quad (6.32)$$

$$\mathbf{D}_{5,i} = [d_{5,i,1}\ d_{5,i,2}\ \cdots\ d_{5,i,S}], \quad (6.33)$$

$$d_{5,i,k} = \begin{cases} p_{01}, & \text{if } h_{iM_i}[k] = 0, \\ 0, & \text{otherwise,} \end{cases} \qquad i = 1,2, \quad (6.34)$$

$$\mathbf{D}_{6,ij} = \begin{bmatrix} d_{6,ij,1} & d_{6,ij,2} & \cdots & d_{6,ij,S} \end{bmatrix}, \quad (6.35)$$

$$d_{6,ij,k} = \begin{cases} p_{ij}(1 - p_{i(j+1)} + d_{6,i(j+1),k}), \\ \qquad \text{if } h_{ij}[k] = N_{ij} \text{ and } ij \neq 1M_1, 2M_2, \\ p_{ij}(1 - p_{01} + \widetilde{d}_{i,k}), \\ \qquad \text{if } h_{ij}[k] = N_{ij} \text{ and } ij = 1M_1, 2M_2, \\ 0, \qquad \text{otherwise}, \end{cases} \tag{6.36}$$

with $h_{ij}[\cdot]$ defined in (6.13) and

$$d_{6,0M_0,k} = 0, \quad k = 1, \ldots, S, \tag{6.37}$$

$$\widetilde{d}_{i,k} = \begin{cases} p_{ij}, & \text{if } h_{i^*M_{i^*}}[k] = 0, \\ d_{6,01,k}, & \text{otherwise}, \end{cases} \tag{6.38}$$

$$i^* = \begin{cases} 1, & \text{if } i = 2, \\ 2, & \text{if } i = 1. \end{cases} \tag{6.39}$$

It should be noted that the dimension of the system grows exponentially with respect to the total number of machines in the system and it may take enormous amount of memory and computing power to implement the calculations above. Therefore, from the computational perspective, equations (6.16)-(6.26) are practical only for small size systems. In fact, using a Macbook Pro with 2.3GHz Intel Core i7 CPU and 16 GB memory, it takes over 5 minutes to calculate the transient performance of an assembly system with $M_0 = M_1 = M_2 = 3$, all machines having efficiency 0.9, all buffers having capacity 2, and $T = 100$, while simulation of the same system of 10,000 runs only takes about 0.2 seconds.

Although simulations can be used as an alternative for system performance evaluation, the results usually contain random errors and the process may still be time-consuming. Clearly, a computationally efficient method based on analytical calculation is necessary to this end, and is developed next.

6.4 Aggregation-based Approximate Performance Evaluation

6.4.1 Background material: Recursive aggregation for transient analysis of Bernoulli serial lines

As it was mentioned earlier, a method based on recursive aggregation for transient performance analysis of serial lines with Bernoulli machines has been developed in [25,42]. Since this method is used as a basis for studying assembly system, we briefly describe it below.

Consider a serial production line shown in Figure 6.1(a) with Bernoulli machines and finite buffers. Let p_i denote the efficiency of machine m_i and let N_i denote the capacity of buffer b_i. Then, an iterative procedure with s being the iteration counter is developed (see [25,42]). Specifically, the procedure is carried out by alternating between "backward aggregation" (represented by superscript b) and "forward aggregation" (represented by superscript f) defined as follows:

Recursive Procedure 1:

$$\begin{aligned}
&p_i^b(s+1;n) = \frac{p_i}{p_i^f(s;n)} \cdot CR(n-1; \mathbf{p}_i^f(s;n), \mathbf{p}_{i+1}^b(s+1;n), N_i, h_i(0)),\\
&i = 1,\dots,M-1, \quad s = 0,1,\dots, \quad n = 1,\dots,T,\\
&p_i^f(s+1;n) = \frac{p_i}{p_i^b(s+1;n)} \cdot PR(n-1; \mathbf{p}_{i-1}^f(s+1;n), \mathbf{p}_i^b(s+1;n), N_{i-1}, h_{i-1}(0)),\\
&i = 2,\dots,M, \quad s = 0,1,\dots, \quad n = 1,\dots,T,
\end{aligned} \tag{6.40}$$

with initial condition $p_i^f(0;n) = p_i$, $i = 1,\dots,M$, and boundary condition $p_1^f(s;n) = p_1$, $p_M^b(s;n) = p_M$, $s = 0,1,\dots$, and

$$\mathbf{p}_i^f(s;n) = [p_i^f(s;1)\ p_i^f(s;2)\ \dots\ p_i^f(s;n)],$$

$$\mathbf{p}_i^b(s;n) = [p_i^b(s;1)\ p_i^b(s;2)\ \dots\ p_i^b(s;n)],$$

and $PR(n-1;\mathbf{v}_1,\mathbf{v}_2,v_3,v_4)$ and $CR(n-1;\mathbf{v}_1,\mathbf{v}_2,v_3,v_4)$ denote the production rate and consumption rate, respectively, during time slot n, of a two-machine Bernoulli line with buffer capacity v_3, initial buffer occupancy v_4 and time-dependent efficiencies of the first and the second machine given by vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. Formulas for calculating these two functions are also given in [42].

A total of 1800000 production lines with randomly generated machine and buffer parameters have been studied in [42]. For all cases studied, it is found that, for any $n \in \{1,2,\ldots,T\}$, there exist limits $p_i^b(n)$ and $p_i^f(n)$, such that

$$\lim_{s\to\infty} p_i^b(s;n) = p_i^b(n), \quad \lim_{s\to\infty} p_i^f(s;n) = p_i^f(n).$$

Using these limiting values, the dynamics of the serial line can be represented by a group of

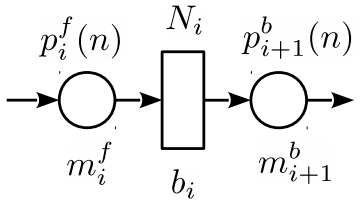


Figure 6.2: Interpretation of aggregated machines in Bernoulli serial lines

two-machine Bernoulli lines with time-varying machine efficiencies (see Figure 6.2): From the perspective of buffer b_i, the overall part-producing effect from all its upstream machines and buffers during time slot n is represented by *forward Bernoulli machine* m_i^f with efficiency $p_i^f(n)$; similarly, the overall part-drawing effect from all its downstream machines and buffers during time slot n is represented by *backward Bernoulli machine* m_{i+1}^b with efficiency $p_{i+1}^b(n)$.

6.4.2 An improved aggregation algorithm for transient analysis of Bernoulli serial lines

It should be noted that, although Recursive Procedure 1 can significantly reduce the computational efforts compared to simulation and exact evaluation, multiple iterations (typically 10-20 iterations) are needed for the procedure to converge. Below, we improve the computational efficiency of the procedure by directly solving for the limiting values instead of using recursive iterations. This improved procedure is described below.

Note that, we can rewrite the aggregation equations in terms of the limiting values as (6.41) and (6.42),

$$\begin{aligned} p_i^b(n) &= \frac{p_i}{p_i^f(n)} CR(n-1; \mathbf{p}_i^f(n), \mathbf{p}_{i+1}^b(n), N_i, h_i(0)) \\ &= p_i \left[1 - (1 - p_{i+1}^b(n)) P_{N_i}(n-1; \mathbf{p}_i^f(n), \mathbf{p}_{i+1}^b(n), N_i, h_i(0))\right], \end{aligned} \tag{6.41}$$

$$\begin{aligned} p_i^f(n) &= \frac{p_i}{p_i^b(n)} PR(n-1; \mathbf{p}_{i-1}^f(n), \mathbf{p}_i^b(n), N_{i-1}, h_{i-1}(0)) \\ &= p_i \left[1 - P_0(n-1; \mathbf{p}_{i-1}^f(n-1), \mathbf{p}_i^b(n-1), N_{i-1}, h_{i-1}(0))\right]. \end{aligned} \tag{6.42}$$

where $P_k(n; \mathbf{v}_1, \mathbf{v}_2, v_3, v_4)$ denote the probability that the buffer has k parts at the end of time slot n in a two-machine Bernoulli line with buffer capacity v_3, initial buffer occupancy v_4 and efficiencies of the first and the second machine given by vectors $\mathbf{v}_1$ and $\mathbf{v}_2$. Since the initial buffer occupancy is given and $p_M^b(n) = p_M$ and $p_1^f(n) = p_1$, it is possible to directly solve $p_i^f(n)$ and $p_{i+1}^b(n)$ by increasing n from 1 to T. Specifically, let $\mathbf{x}^{(i)}(n) = [x_0^{(i)}(n)\ x_1^{(i)}(n)\ \ldots,\ x_{N_i}^{(i)}(n)]^T$, where $x_k^{(i)}(n)$ is the probability that the buffer has k parts at the end of time slot n in the two-machine Bernoulli line shown in Figure 6.2. The calculation procedure is given as follows:

Calculation Procedure 1:

Step 0 : Let $n = 1$. The boundary condition of the procedure is $p_M^b(n) = p_M$ and $p_1^f(n) = p_1$. The initial condition is:

$$x_k^{(i)}(0) = \begin{cases} 1, & \text{if } h_i(0) = k, \\ 0, & \text{otherwise.} \end{cases} \qquad i = 1, \ldots, M-1.$$

Step 1 : Calculate $p_i^f(n)$, for all $i = 2, \ldots, M$, as follows: $p_i^f(n) = p_i \left[1 - x_0^{(i-1)}(n-1)\right]$.

Step 2 : Calculate $p_i^b(n)$ for all $i = 1, \ldots, M-1$, in the descending order of i, i.e., first calculate $p_{M-1}^b(n)$ and last $p_1^b(n)$ as follows: $p_i^b(n) = p_i \left[1 - (1 - p_{i+1}^b(n))x_{N_i}^{(i)}(n-1)\right]$.

Step 3 : Calculate $\mathbf{x}^{(i)}(n)$ for all $i = 1, \ldots, M-1$ based on

$$\mathbf{x}^{(i)}(n) = \mathbf{A}_2(p_i^f(n), p_{i+1}^b(n), N_i)\mathbf{x}^{(i)}(n-1),$$

where $\mathbf{A}_2(p_i^f(n), p_{i+1}^b(n), N_i)$ is the one-step transition probability matrix of the two-machine Bernoulli line shown in Figure 6.2 during time n.

Step 4 : If $n = T$, terminate the procedure; otherwise, let $n = n + 1$ and return to Step 1 .

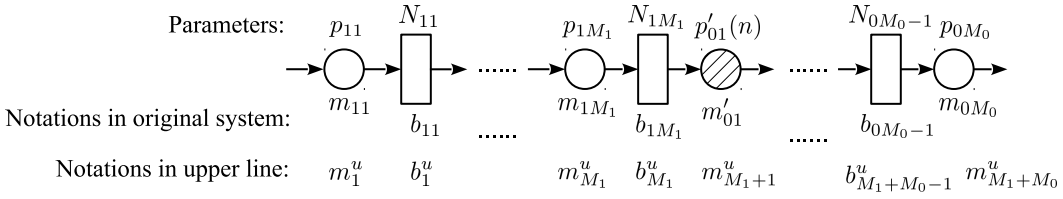
It can be seen that Recursive Procedure 1 and Calculation Procedure 1 share the same initial condition, boundary condition, and aggregation formulas. Specifically, from (6.41) and (6.42), we have

$$x_j^{(i)}(n) = P_j(n-1; \mathbf{p}_i^f(n), \mathbf{p}_{i+1}^b(n), N_i, h_i(0)). \tag{6.43}$$

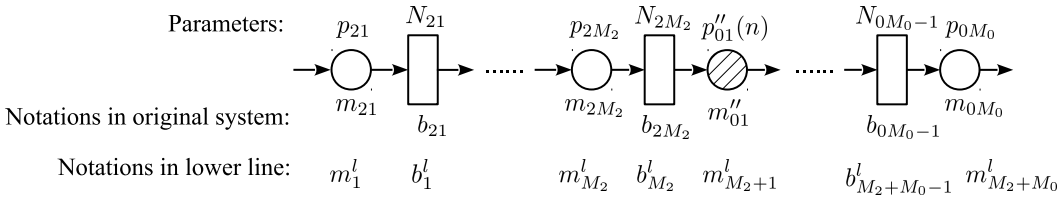
Therefore, the numerical results of the two procedures are completely equivalent. However, since Calculation Procedure 1 requires no recursive iterations, the computational time is reduced to only about 10% or less of Recursive Procedure 1. Next, we apply the idea of Calculation Procedure 1 to the analysis of assembly systems.

6.4.3 Aggregation for transient analysis of assembly systems with Bernoulli machines

To study the steady state performance and improvability of the assembly system shown in Figure 6.1(b), a decomposition method is developed in [113] to transform the system into a pair of serial lines: an upper line with (M_1+M_0) machines and a lower line with (M_2+M_0) machines. In this subsection, the same idea is applied to study its transient performance. Specifically, the upper line is just the original system with all machines and buffers but remove Component Line 2. To account for this modification, the original assembly machine m_{01} is replaced by a virtual machine m'_{01}, represented by the shaded circle in Figure 6.3(a). In addition, since the system operates in transients, the parameter of this virtual machine should be time-varying, denoted as $p'_{01}(n)$. Similarly, the lower line can be constructed by removing Component Line 1 and replacing the assembly operation with virtual machine m''_{01}. We use $p''_{01}(n)$ to denote the parameter of this virtual machine in the lower line.



(a) Upper line



(b) Lower line

Figure 6.3: Transforming an assembly system into upper line and lower line

Clearly, the efficiencies of virtual machines m'_{01} and m''_{01} are not directly known from the original assembly system. To obtain these parameters, i.e., $p'_{01}(n)$ and $p''_{01}(n)$, note that virtual machine m'_{01} is "up" in the upper line if and only if the original assembly operation m_{01} is up and buffer b_{2M_2} is not empty. Similarly, virtual machine m''_{01} is "up" in the lower

line if and only if the original assembly operation m_{01} is up and buffer b_{1M_1} is not empty. Therefore, $p'_{01}(n)$ and $p''_{01}(n)$ may be estimated using the following formulas:

$$p'_{01}(n) \approx p_{01}\left(1 - P[h_{2M_2}(n-1) = 0]\right), \tag{6.44}$$

$$p''_{01}(n) \approx p_{01}\left(1 - P[h_{1M_1}(n-1) = 0]\right). \tag{6.45}$$

Since the probability distributions of buffer occupancy in serial lines can be estimated using Calculation Procedure 1, we can apply this technique to both the lower and upper lines in an iterative fashion and calculate the values of $p'_{01}(n)$ and $p''_{01}(n)$ for each time instant n. Finally, since the machines and buffers downstream of m'_{01} in the upper line and those downstream of m''_{01} in the lower lines are completely the same, the calculation of this part of the system only needs to be performed once: either in the upper line or in the lower line. Based on the ideas elaborated above, the following calculation procedure is proposed:

Calculation Procedure 2:

Step 0 : Let $n = 1$. Define

$$p_i^u = \begin{cases} p_{1i}, & 1 \le i \le M_1, \\ p_{0(i-M_1)}, & M_1 + 1 \le i \le M_1 + M_0, \end{cases} \tag{6.46}$$

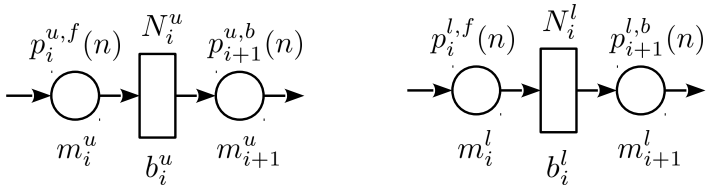
$$N_i^u = \begin{cases} N_{1i}, & 1 \le i \le M_1, \\ N_{0(i-M_1)}, & M_1 + 1 \le i \le M_1 + M_0 - 1, \end{cases} \tag{6.47}$$

$$p_i^l = \begin{cases} p_{2i}, & 1 \le i \le M_2, \\ p_{01}, & i = M_2 + 1, \end{cases} \tag{6.48}$$

$$N_i^l = N_{2i}, \quad 1 \le i \le M_2. \tag{6.49}$$

Note that, in the expressions above, superscripts u and l are used to denote upper line and lower line parameters. Without loss of generality, the parameters of the overlapped portion is calculated through the upper line in this procedure.

Consider the two-machine Bernoulli lines shown in Figure 6.4. Let $\mathbf{x}^{(i,u)}(n) = [x_0^{(i,u)}(n)\ x_1^{(i,u)}(n)\ \ldots\ x_{N_i^u}^{(i,u)}(n)]^T$, where $x_k^{(i,u)}(n)$ is the probability that the buffer has k parts at the end of time slot n in the two-machine Bernoulli line shown in Figure 6.4(a). In addition, let $\mathbf{x}^{(i,l)}(n) = [x_0^{(i,l)}(n)\ x_1^{(i,l)}(n)\ \ldots\ x_{N_i^l}^{(i,l)}(n)]^T$, where $x_k^{(i,l)}(n)$ is the probability that the buffer has k parts at the end of time slot n in the two-machine Bernoulli line shown in Figure 6.4(b). The initial condition is given by:



(a) Upper line aggregation (b) Lower line aggregation

Figure 6.4: Two-machine line representations in upper line and lower line aggregations

$$x_k^{(i,u)}(0) = \begin{cases} 1, & \text{if } h_{1i}(0) = k,\ i \in \{1, \ldots, M_1\}, \\ 1, & \text{if } M_0 > 1 \text{ and } h_{0(i-M_1)}(0) = k, \\ & \quad i \in \{M_1 + 1, \ldots, M_1 + M_0 - 1\}, \\ 0, & \text{otherwise,} \end{cases} \tag{6.50}$$

$$k = 0, 1, \ldots, N_i^u,$$

$$x_k^{(i,l)}(0) = \begin{cases} 1, & \text{if } h_{2i}(0) = k,\ i \in \{1, \ldots, M_2\}, \\ 0, & \text{otherwise,} \end{cases} \tag{6.51}$$

$$k = 0, 1, \ldots, N_i^l.$$

Step 1 : Calculate the parameters of the forward aggregated machines, $p_i^{u,f}(n)$ and $p_i^{l,f}(n)$:

$$p_i^{u,f}(n) = \begin{cases} p_i^u, & \text{if } i = 1, \\ p_i^u \left[1 - x_0^{(i-1,u)}(n-1)\right], & \text{if } i \neq 1 \text{ and } i \neq M_1 + 1, \\ p_i^u \left[1 - x_0^{(i-1,u)}(n-1)\right] \left[1 - x_0^{(M_2,l)}(n-1)\right], & \text{if } i = M_1 + 1, \end{cases} \tag{6.52}$$

$$p_i^{l,f}(n) = \begin{cases} p_i^l, & \text{if } i = 1, \\ p_i^l \left[1 - x_0^{(i-1,l)}(n-1)\right], & \text{if } i \neq 1 \text{ and } i \neq M_2 + 1, \\ p_i^l \left[1 - x_0^{(i-1,l)}(n-1)\right] \left[1 - x_0^{(M_1,u)}(n-1)\right], & \text{if } i = M_2 + 1. \end{cases} \tag{6.53}$$

Step 2 : Calculate the parameters of the backward aggregated machines, $p_i^{u,b}(n)$, in the descending order of i, i.e., first calculate $p_{M_1+M_0}^{u,b}$ and last $p_1^{u,b}$, as follows:

$$p_i^{u,b}(n) = \begin{cases} p_i^u, & \text{if } i = M_1 + M_0,\ M_0 > 1, \\ p_i^u \left[1 - x_0^{(M_2,l)}(n-1)\right], & \text{if } i = M_1 + M_0,\ M_0 = 1, \\ p_i^u \left[1 - (1 - p_{i+1}^{u,b}(n)) x_{N_i^u}^{(i,u)}(n-1)\right] \cdot \left[1 - x_0^{(M_2,l)}(n-1)\right], & \\ & \text{if } i = M_1 + 1,\ M_0 > 1, \\ p_i^u \left[1 - (1 - p_{i+1}^{u,b}(n)) x_{N_i^u}^{(i,u)}(n-1)\right], & \text{otherwise.} \end{cases} \tag{6.54}$$

Calculate $p_i^{l,b}(n)$ in the descending order of i, i.e., first calculate $p_{M_2+1}^{l,b}$ and last $p_1^{l,b}$, as follows:

$$p_i^{l,b}(n) = \begin{cases} p_i^l \left[1 - (1 - p_{M_1+2}^{u,b}(n)) x_{N_{01}}^{(M_1+1,u)}(n-1)\right] \cdot \left[1 - x_0^{(M_1,u)}(n-1)\right], & \\ & \text{if } i = M_2 + 1,\ M_0 > 1, \\ p_i^l \left[1 - x_0^{(M_1,u)}(n-1)\right], & \text{if } i = M_2 + 1,\ M_0 = 1, \\ p_i^l \left[1 - (1 - p_{i+1}^{l,b}(n)) x_{N_i^l}^{(i,l)}(n-1)\right], & \text{if } i \leq M_2. \end{cases} \tag{6.55}$$

Step 3 : Calculate $\mathbf{x}^{(i,u)}(n)$ and $\mathbf{x}^{(i,l)}(n)$ based on

$$\mathbf{x}^{(i,u)}(n) = \mathbf{A}_2(p_i^{u,f}(n), p_{i+1}^{u,b}(n), N_i^u)\mathbf{x}^{(i,u)}(n-1), \tag{6.56}$$

$$\mathbf{x}^{(i,l)}(n) = \mathbf{A}_2(p_i^{l,f}(n), p_{i+1}^{l,b}(n), N_i^l)\mathbf{x}^{(i,l)}(n-1), \tag{6.57}$$

where $\mathbf{A}_2(p_i^{u,f}(n), p_{i+1}^{u,b}(n), N_i^u)$ and $\mathbf{A}_2(p_i^{l,f}(n), p_{i+1}^{l,b}(n), N_i^l)$ are the one-step transition

probability matrices of the two-machine Bernoulli lines shown in Figure 6.4(a) and 6.4(b) during time n, respectively.

Step 4 : If $n = T$, terminate the procedure; otherwise, let $n = n + 1$ and return to Step 1 .

6.4.4 Performance estimates and their accuracy

Based on Calculation Procedure 2, we propose to estimate the transient performance measures of the original assembly system as follows:

$$\widehat{PR}(n) = p^{u,f}_{M_1+M_0}(n+1), \tag{6.58}$$

$$\widehat{CR}_1(n) = p^{u,b}_1(n+1), \tag{6.59}$$

$$\widehat{CR}_2(n) = p^{l,b}_1(n+1), \tag{6.60}$$

$$\widehat{WIP}_{ij}(n) = \begin{cases} \sum_{k=1}^{N_{0j}} \left[kx_k^{(j+M_1,u)}(n) \right], & \text{if } i = 0,\ M_0 > 1 \text{ and } 1 \le j \le M_0 - 1, \\ \sum_{k=1}^{N_{1j}} \left[kx_k^{(j,u)}(n) \right], & \text{if } i = 1 \text{ and } 1 \le j \le M_1, \\ \sum_{k=1}^{N_{2j}} \left[kx_k^{(j,l)}(n) \right], & \text{if } i = 2 \text{ and } 1 \le j \le M_2, \end{cases} \tag{6.61}$$

$$\widehat{ST}_{ij}(n) = \begin{cases} p_{ij} x_0^{(j+M_1-1,u)}(n), & \text{if } i = 0,\ 2 \le j \le M_0, \\ p_{ij} x_0^{(j-1,u)}(n), & \text{if } i = 1,\ 2 \le j \le M_1, \\ p_{ij} x_0^{(j-1,l)}(n), & \text{if } i = 2,\ 2 \le j \le M_2, \end{cases} \tag{6.62}$$

$$\widehat{ST}_{01,1}(n) = p_{01} x_0^{(M_1,u)}(n), \tag{6.63}$$

$$\widehat{ST}_{01,2}(n) = p_{01} x_0^{(M_2,l)}(n), \tag{6.64}$$

$$\widehat{BL}_{ij}(n) = \begin{cases} p_{ij} x_{N_{ij}}^{(j+M_1,u)}(n) \left[1 - p^{u,b}_{j+M_1+1}(n) \right], & \text{if } i = 0 \text{ and } 1 \le j \le M_0 - 1, \\ p_{ij} x_{N_{ij}}^{(j,u)}(n) \left[1 - p^{u,b}_{j+1}(n) \right], & \text{if } i = 1 \text{ and } 1 \le j \le M_1, \\ p_{ij} x_{N_{ij}}^{(j,l)}(n) \left[1 - p^{l,b}_{j+1}(n) \right], & \text{if } i = 2 \text{ and } 1 \le j \le M_2. \end{cases} \tag{6.65}$$

To evaluate the accuracy of these estimates, the accuracy metrics defined in (6.66) are

used,

$$\delta_{PR}(n) = \frac{|PR(n) - \widehat{PR}(n)|}{PR_{ss}} \cdot 100\%, \quad \delta_{CR_i}(n) = \frac{|CR_i(n) - \widehat{CR}_i(n)|}{CR_{ss}} \cdot 100\%,$$

$$\delta_{WIP}(n) = \frac{\left[\sum_{i=1}^{M_0-1} \frac{|WIP_{0i}(n) - \widehat{WIP}_{0i}(n)|}{N_{0i}} + \sum_{i=1}^{M_1} \frac{|WIP_{1i}(n) - \widehat{WIP}_{1i}(n)|}{N_{1i}} + \sum_{i=1}^{M_2} \frac{|WIP_{2i}(n) - \widehat{WIP}_{2i}(n)|}{N_{2i}}\right]}{M_0 + M_1 + M_2 - 1} \cdot 100\%,$$

$$\delta_{ST}(n) = \frac{\left[\sum_{i=1}^{2} |ST_{01,i}(n) - \widehat{ST}_{01,i}(n)| + \sum_{i=0}^{2} \sum_{j=2}^{M_i} |ST_{ij}(n) - \widehat{ST}_{ij}(n)|\right]}{M_0 + M_1 + M_2 - 1}, \tag{6.66}$$

$$\delta_{BL}(n) = \frac{\left[\sum_{i=1}^{M_0-1} |BL_{0i}(n) - \widehat{BL}_{0i}(n)| + \sum_{i=1}^{M_1} |BL_{1i}(n) - \widehat{BL}_{1i}(n)| + \sum_{i=1}^{M_2} |BL_{2i}(n) - \widehat{BL}_{2i}(n)|\right]}{M_0 + M_1 + M_2 - 1}.$$

where PR_{ss} and CR_{ss} are the steady state production rate and consumption rate of the system, respectively. Clearly, for systems defined by assumptions (i)-(viii), both component lines have identical steady state consumption rate CR_{ss}, and $PR_{ss} = CR_{ss}$.

Next, a total of 500,000 assembly systems were generated with M_0, M_1, M_2 randomly selected from $\{1, 2, \ldots, 10\}$. In addition, the machine efficiency, buffer capacity, and initial buffer occupancy of the lines were selected randomly and equiprobably from the following sets:

$$p_{ij} \in (0.7, 1),\ N_{ij} \in \{1, 2, \ldots, 10\},\ h_{ij}(0) \in \{0, 1, \ldots, N_{ij}\}. \tag{6.67}$$

The systems' total operation time T is selected as 3,000 time slots, which is long enough to contain the entire transient period of most systems studied. For each system, thus constructed, Calculation Procedure 2 was performed and the transient performance estimates were calculated using equations (6.58)-(6.65). On the other hand, although the systems' exact performance measures, $PR(n)$, $CR_i(n)$, $WIP_{ij}(n)$, $ST_{ij}(n)$, $ST_{01,i}(n)$, and $BL_{ij}(n)$, can be calculated using (6.16)-(6.26) from Section 6.3, the computational efforts required are far beyond practical. The same problem also exist for exact evaluation of the steady state performance measures PR_{ss} and CR_{ss}. As a result, these terms were evaluated using

simulations based on the following:

Simulation Procedure 1:

(1) Set the initial status of machine m_{ij} to be up with probability p_{ij} and down with probability $1 - p_{ij}$.

(2) Set the initial occupancy of buffer b_{ij} as $h_{ij}(0)$.

(3) For transient performance evaluation, carry out 10,000 runs of the simulation code for each system and calculate the average performance during each time slot.

(4) For steady state performance evaluation, carry out 20 runs of the simulation code for each system. In each run, use the first 20,000 time slots as a warm-up period and the subsequent 400,000 time slots to statistically calculate the average performance.

(5) This results in 95% confidence intervals of less than 0.001 for PR_{ss} and CR_{ss}; 0.005 for $PR(n)$ and $CR_i(n)$; 0.05 for $WIP_{ij}(n)$; and 0.01 for $ST_{ij}(n)$ and $BL_{ij}(n)$.

The results obtained are summarized in Figure 6.5. Specifically, the solid lines indicate the averages of the estimation errors defined in (6.66). Since the accuracy for the performance measures in the same graph is similar, the dashed lines only plot the largest first and third quartiles of the estimation errors for the corresponding performance measure estimates in the same figure. As one can see, the average $\delta_{PR}(n)$ and $\delta_{CR_i}(n)$ are typically within 1.5% with the upper quartile less than 2%. The average $\delta_{WIP}(n)$ is typically within 3% with the upper quartile around 4%. The average $\delta_{ST}(n)$ and $\delta_{BL}(n)$ are typically within 0.01 with the upper quartile less than 0.012. It should be noted that the accuracy of the aggregation procedures is typically low when the buffer capacities in the system are very small. This is because the coupling effect of the machines and buffers in an assembly system becomes stronger when the buffer capacity decreases, which makes it more difficult for the aggregation procedure to *decouple* the machines in the system. Taking into account that the parameters of the machines and buffers are rarely known on the factory floor with accuracy better than

5%-10%, we conclude that Calculation Procedure 2 and (6.58)-(6.65) can be used as an effective tool to estimate the transient performance of Bernoulli assembly systems with good accuracy.

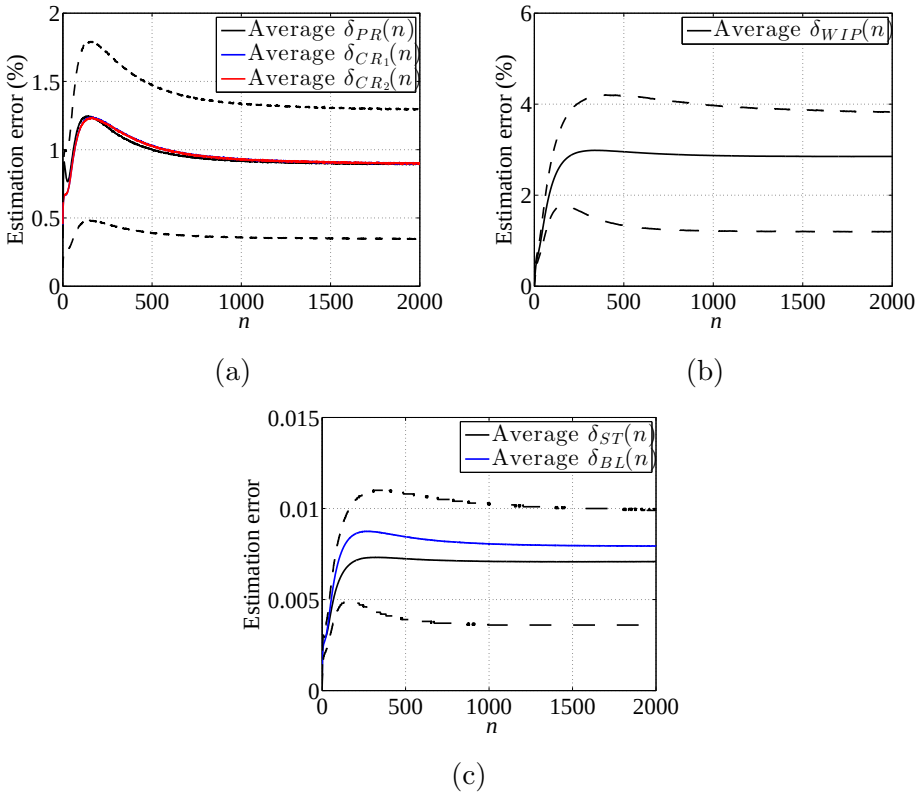


Figure 6.5: Accuracy of performance estimates (6.58)-(6.65)

As an illustration, consider the assembly system shown in Figure 6.6. The number above each machine (circle) represents its efficiency, while the number inside each buffer (rectangle) indicates its capacity. These parameters are randomly generated from (6.67). In this example, all buffers are assumed to be empty at time $n = 0$. The transient performance measures of this system, obtained using simulation and Calculation Procedure 2 are given in Figure 6.7. As one can see from the leftmost column of Figure 6.7, the random error contained in the simulation results is quite strong, especially for PR, CR, ST_{ij} and BL_{ij}. For this example, the computational time required to obtain the simulation results shown in the leftmost column is approximately twice as long as required by Calculation Procedure 2. The middle column of Figure 6.7 shows the results when we increase the simulation

replications such that the total computational time is about 100 times of the one required by the calculation method. Now, the close similarity between the simulation results and the calculation results can be easily observed (see rightmost column of Figure 6.7). Clearly, the calculation-based technique needs much less computational resource and is capable of delivering results that are ready to be used for further applications.

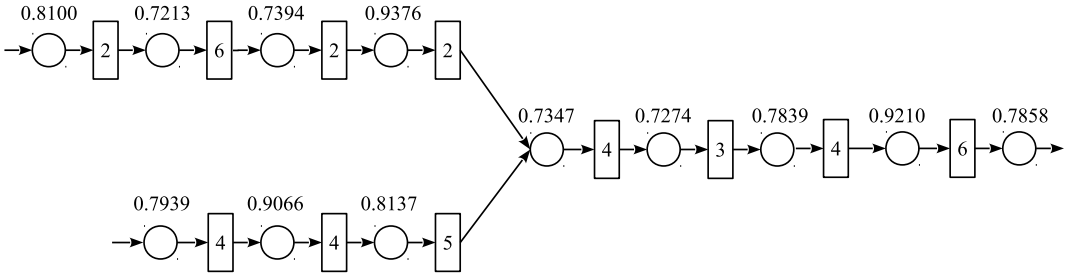


Figure 6.6: Example of transient performance evaluation for Bernoulli assembly system

6.5 Extension to Complex Assembly Systems

6.5.1 Generalized calculation procedure

The system considered in the previous sections contains only two component lines and one assembly operation. In practice, an assembly system may involve multiple component lines and several assembly operations. An example is shown in Figure 6.8. The final product of this system has six components and the system has four assembly operations. One of them ($ao1$) operates to assemble three components into one unit, while the last two assembly operations ($ao3$ and $ao4$) both work with subassemblies.

To analyze the two-component assembly system shown in Figure 6.1(b), we transform the system into a pair of serial lines shown in Figure 6.3. Note that, the upper line contains all and only the machines and buffers that process and hold Component 1, while the lower line contains the machines and buffers that process and hold Component 2. The idea of the transformation can be extended to assembly systems with multiple components and

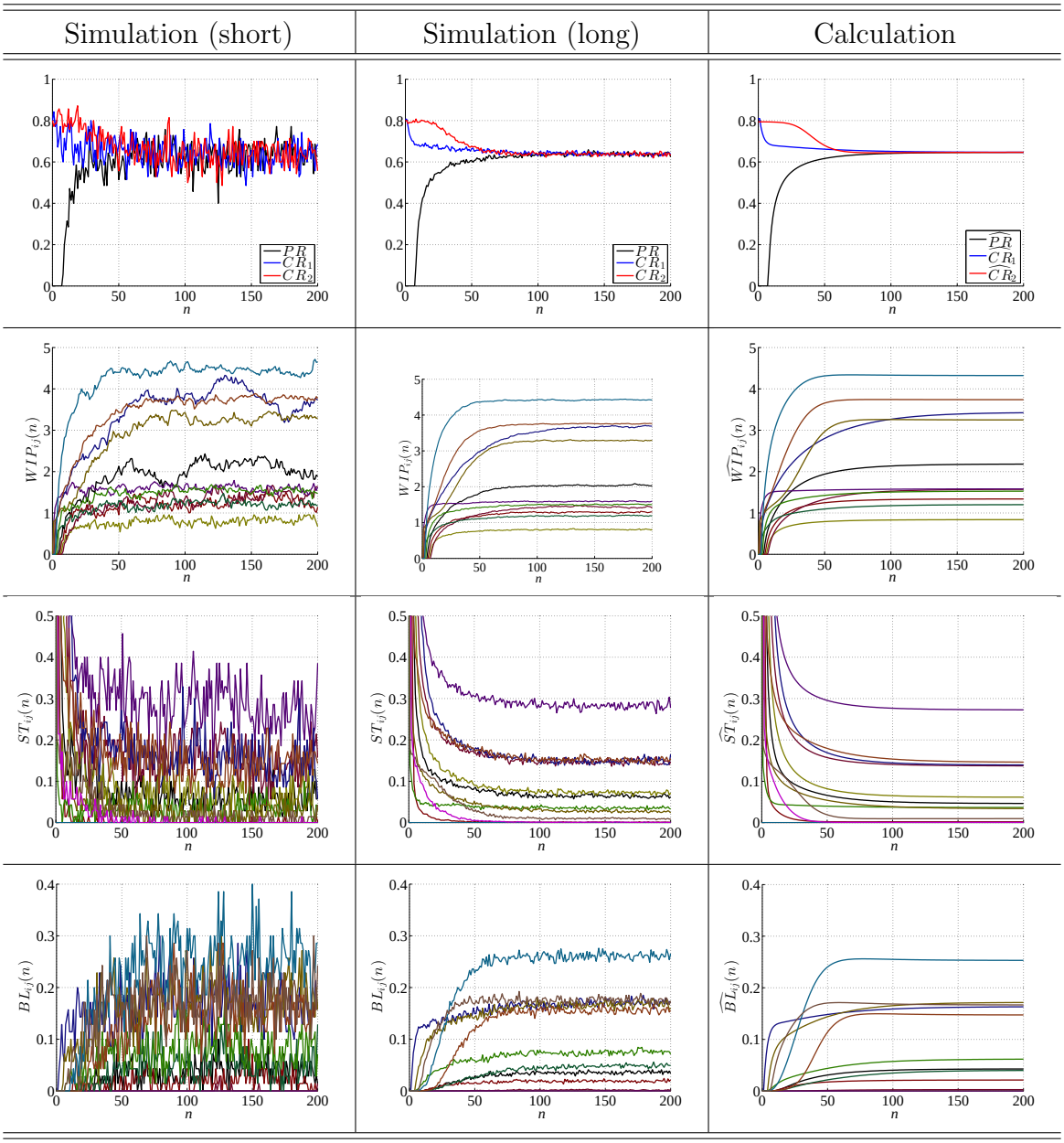


Figure 6.7: Comparison of simulation- and calculation-based methods for transient performance evaluation in Bernoulli assembly system

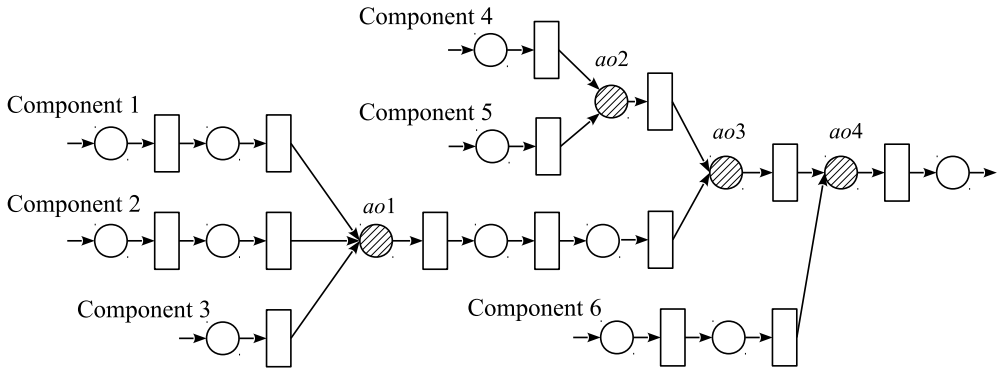


Figure 6.8: Example of assembly system with multiple component lines and assembly operations

multiple assembly operations as well. Specifically, the assembly system will be decomposed into several lines with machine parameters modified to reflect the effects from other lines (i.e., based on starvations and blockages):

Calculation Procedure 3:

Step 1 : Identify the machine-buffer paths for the parts flow of all components.

Step 2 : Construct a virtual serial line based on the parts flow path of each component. Specifically, start with the first machine of a path and continue adding downstream machines and buffers along the path until the last machine of the system is reached or when a machine, which has appeared in previously constructed lines, is reached. We refer to the serial line constructed based on component w's flow path as *Line* w.

Step 3 : The efficiencies (capacities) of the machines (buffers) in each virtual line constructed above are set to the efficiencies (capacities) of the corresponding machines (buffers) in the parts flow path of the component.

Step 4 : Let $n = 1$ and let $p_i^{(w)}$ and $N_i^{(w)}$ denote the efficiency of the i-th machine (m_i^w) and capacity of the i-th buffer (b_i^w) in the virtual serial line for component w constructed above, respectively. Consider the two-machine line shown in Figure 6.9. Let $\mathbf{x}^{(i,w)}(n) = [x_0^{(i,w)}(n)\ x_1^{(i,w)}(n)\ \ldots\ x_{N_i^{(w)}}^{(i,w)}(n)]^T$, where $x_k^{(i,w)}(n)$ is the probability that the buffer has

k parts at the end of time slot n in the two-machine Bernoulli line shown in Figure 6.9 (i.e., around the i-th buffer in Line w).

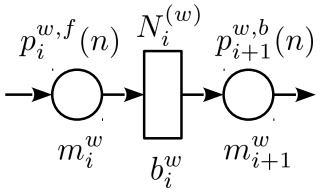


Figure 6.9: Two-machine line representation of the virtual serial line for component w during aggregations

Step 5 : Calculate the parameters of the forward aggregated machines, $p_i^{w,f}(n)$:

$$p_i^{w,f}(n) = \begin{cases} p_i^{(w)}, & \text{if } i = 1, \\ p_i^{(w)} \prod_{(j,v)\in I_{us}(i,w)} \left[1 - x_0^{(j,v)}(n-1)\right], & \\ & \text{otherwise,} \end{cases}$$

where $I_{us}(i,w)$ represents the set of 2-tuples (j,v) such that buffer b_j^v (i.e., the j-th buffer of Line v) corresponds to an immediate upstream buffer of machine m_i^w (i.e., the i-th machine of Line w) in the original assembly system. In other words, if the machine corresponds to the first machine in a line, then its original parameter is retained; otherwise, the parameter should be modified to reflect possible starvations from all its immediate upstream buffers (i.e., $I_{us}(i,w)$).

Step 6 : Calculate the parameters of the backward aggregated machines, $p_i^{w,b}(n)$:

$$p_i^{w,b}(n) = \begin{cases} p_i^{(w)} \prod_{(j,v)\in I_{us}(i,w), v\neq w} \left[1 - x_0^{(j,v)}(n-1)\right], & \\ \text{if } m_i^w \text{ corresponds to the last machine in the original system,} & \\ p_i^{(w)} \prod_{(j,v)\in I_{us}(i,w), v\neq w} \left[1 - x_0^{(j,v)}(n-1)\right] \cdot \left[1 - \left(1 - p_{i^d+1}^{w^d,b}\right) x_{N_{i^d}^{(w^d)}}^{(i^d,w^d)}(n-1)\right], & \\ \quad \text{otherwise,} & \end{cases}$$

where the immediate downstream buffer of machine m_i^w in the original system corresponds to the i^d-th buffer in the serial line for component w^d. Here, the parameter of a backward aggregated machine should include its blockage from downstream (i.e., the (i^d+1)-th machine in line w^d) and all starvations from other lines (i.e., $I_{us}(i,w)$ with $v \neq w$).

Step 7 : Calculate $\mathbf{x}^{(i,w)}(n)$ based on

$$\mathbf{x}^{(i,w)}(n) \quad = \quad \mathbf{A}_2(p_i^{w,f}(n), p_{i+1}^{w,b}(n), N_i^{(w)})\mathbf{x}^{(i,w)}(n-1),$$

where $\mathbf{A}_2(p_i^{w,f}(n), p_{i+1}^{w,b}(n), N_i^{(w)})$ is the one-step transition probability matrix of the two-machine Bernoulli line shown in Figure 6.9 during time n.

Step 8 : If $n = T$, terminate the procedure; otherwise, let $n = n+1$ and return to Step 5.

6.5.2 Example

To illustrate the implementation of Calculation Procedure 3, consider the assembly system shown in Figure 6.8 as an example. During Step 1 through Step 3 of the algorithm, one serial line is constructed for each component. As a result, six lines are obtained (see Figure 6.10). As one can see, Line 1 starts from the first machine in the component path and ends at the last machine of the system. On the other hand, Line 2 and Line 3 both end at operation $ao1$, since the rest of the machines and buffers in the component paths have already been included in Line 1. Similarly, Line 4 ends at $ao3$, Line 5 ends at $ao2$, and Line 6 ends at $ao4$. Note that each buffer appears in one and only one of the serial lines, while each non-assembly operations also appears in one and only one of the serial lines. The number of appearances of each assembly operation is equal to the number of components assembled at this machine.

To carry out Step 5 of the procedure, note that all immediate upstream buffers should be included in the calculation. For example, operation $ao1$ in the original system appears as

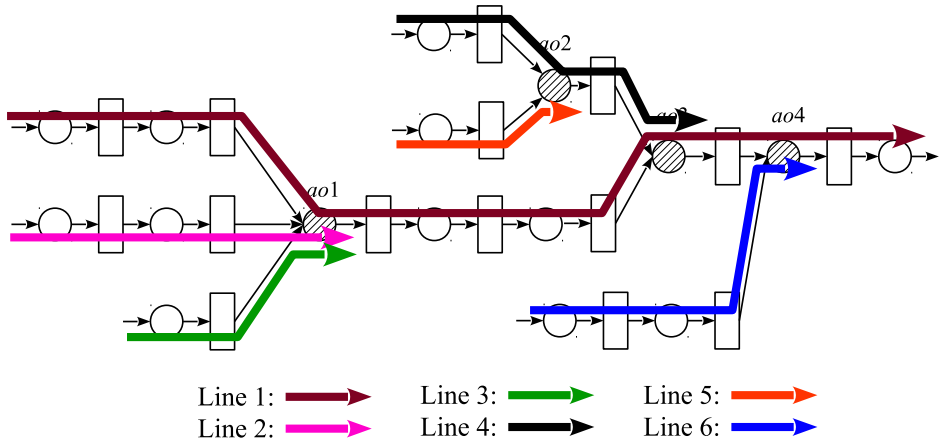


Figure 6.10: Virtual serial line construction in an assembly system with multiple component lines and assembly operations based on component parts flow paths

$m_3^{(1)}$ (3rd machine in Line 1), $m_3^{(2)}$ (3rd machine in Line 2), and $m_2^{(3)}$ (2nd machine in Line 3). Based on Calculation Procedure 3, the calculation of their f parameters should include buffers $b_2^{(1)}$ (2nd buffer in Line 1), $b_2^{(2)}$ (2nd buffer in Line 2), and $b_1^{(3)}$ (1st buffer in Line 3). In other words,

$$I_{us}(3,1) = I_{us}(3,2) = I_{us}(2,3) = \{(2,1),(2,2),(1,3)\}.$$

Let p_{ao1}, p_{ao2}, p_{ao3}, and p_{ao4} denote the efficiencies of assembly operations $ao1$, $ao2$, $ao3$, and $ao4$, respectively. Then, we have

$$p_3^{1,f}(n) = p_3^{2,f}(n) = p_2^{3,f}(n) = p_{ao1}\left[1 - x_0^{(2,1)}(n-1)\right] \cdot \left[1 - x_0^{(2,2)}(n-1)\right] \cdot \left[1 - x_0^{(1,3)}(n-1)\right].$$

Similarly, $ao2$ appears as $m_2^{(4)}$ and $m_2^{(5)}$, $ao3$ appears as $m_6^{(1)}$ and $m_3^{(4)}$, and $ao4$ appears as $m_7^{(1)}$ and $m_3^{(6)}$. Their corresponding f parameters can be calculated as:

$$\begin{aligned} p_2^{4,f}(n) &= p_2^{5,f}(n) = p_{ao2}\left[1 - x_0^{(1,4)}(n-1)\right]\left[1 - x_0^{(1,5)}(n-1)\right], \\ p_6^{1,f}(n) &= p_3^{4,f}(n) = p_{ao3}\left[1 - x_0^{(5,1)}(n-1)\right]\left[1 - x_0^{(2,4)}(n-1)\right], \end{aligned}$$

$$p_7^{1,f}(n) \;=\; p_3^{6,f}(n) = p_{ao4}\left[1 - x_0^{(6,1)}(n-1)\right]\left[1 - x_0^{(2,6)}(n-1)\right].$$

The calculation of the f parameters for the non-assembly operations is straightforward and, thus, is not elaborated here.

To calculate the b parameters of the machines, note that the last machine only appears in Line 1 as $m_8^{(1)}$. Let p_{last} denote the efficiency of the last machine in the assembly system. Then,

$$p_8^{1,b}(n) = p_{last}, \quad \forall n.$$

For all other lines, the last machines in the lines are not the last machine of the original assembly system. Therefore, to calculate their b parameters, we need to locate these machines' immediate downstream buffers and machines in the original system. For instance, the last machine in Line 4 ($m_3^{(4)}$) is an internal machine of the original system, $ao3$. Its immediate downstream buffer and machine appear in Line 1 as $b_6^{(1)}$ and $m_7^{(1)}$. In addition, its immediate upstream buffer, which does not belong to Line 4, is buffer $b_5^{(1)}$ in Line 1. Thus, the b parameter of $m_3^{(4)}$ should be calculated as follows:

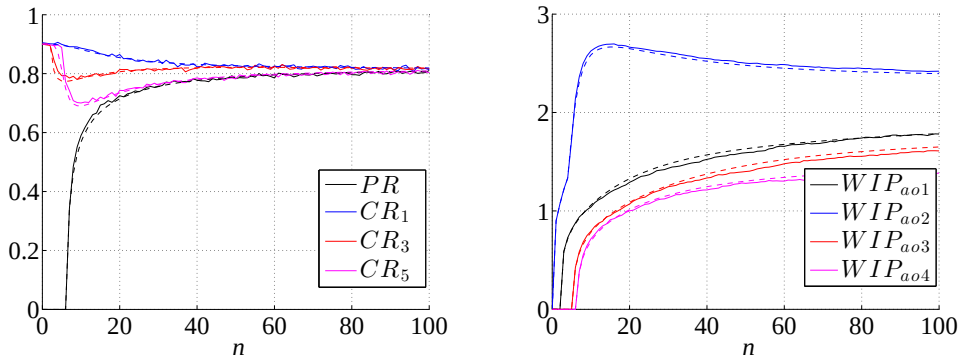
$$p_3^{4,b}(n) \;=\; p_{ao3}\left[1 - x_0^{(5,1)}(n-1)\right] \cdot \left[1 - \left(1 - p_7^{1,b}(n)\right) x_{N_6^{(1)}}^{(6,1)}(n-1)\right].$$

If an assembly operation appears as an internal machine in a line, for instance, $ao3$ appearing as $m_6^{(1)}$ in Line 1, its immediate upstream buffer not belonging to Line 1 is buffer $b_2^{(4)}$ in Line 4. Thus, the b parameter of $m_6^{(1)}$ is calculated as follows:

$$p_6^{1,b}(n) \;=\; p_{ao3}\left[1 - x_0^{(2,4)}(n-1)\right] \cdot \left[1 - \left(1 - p_7^{1,b}(n)\right) x_{N_6^{(1)}}^{(6,1)}(n-1)\right].$$

The calculation of the b parameters for the non-assembly operations is straightforward and, thus, is not elaborated here.

As an illustration, assume that the efficiency of all machines is 0.9 and all buffers have capacity equal to 3. The transient performance measures of the system are evaluated by both simulation and Calculation Procedure 3. Due to space limitation, only part of the results are given below (see Figure 6.11). Specifically, Figure 6.11(a) presents the results of the system production rate of the end product and the raw part consumption rates of Components 1, 3, and 5. In Figure 6.11(b), the work-in-process of the buffers succeeding the assembly operations are provided. In both figures, the solid lines represent simulation results, while the dashed lines represent results from the calculation procedure. As one can see, the generalized calculation procedure is still capable of approximating the system transient performance with high accuracy.



(a) (b)

Figure 6.11: Transient performance evaluation of assembly system with multiple component lines and assembly operations

Similar to Subsection 6.4.4, extensive numerical experiments beyond this example have also been carried out to justify the accuracy of the calculation procedure. The results are very similar to the ones reported in Subsection 6.4.4, and, thus, are not repeated. Finally, it should be noted that the procedure can be further generalized to systems with machines having time-varying efficiencies by simply replacing the time-constant parameters in the current form by the ones at corresponding time instants.

6.6 Summary

In this Chapter, we study the performance evaluation problem of assembly systems with Bernoulli reliability machines and finite capacity buffers. Based on Markovian analysis, we first study systems with two component lines and one assembly operation. Specifically, mathematical model and exact performance evaluation formulas are derived. Then, an improved algorithm for transient performance calculation in Bernoulli serial lines is proposed. Based on this improved algorithm, a computationally-efficient procedure is developed to approximate the transient performance measures of Bernoulli assembly system by transforming the system into a pair of interacting serial lines. The accuracy of the method developed is justified using extensive numerical simulations. Finally, we extend the calculation procedure to complex assembly systems with multiple component lines and assembly operations.

Chapter 7

TRANSIENT PERFORMANCE ANALYSIS FOR SERIAL PRODUCTION LINES WITH BERNOULLI MACHINES AND REAL-TIME WIP-BASED MACHINE SWITCH-ON/OFF CONTROL

7.1 Introduction

Productivity and quality have been the focus of manufacturing systems research over years (see monographs by [115–119]). On the other hand, because of the increasing energy costs and environmental concerns, reducing energy consumption and greenhouse gas emissions have become critical for the manufacturing industry in recent years. To identify potential energy-saving opportunities, an inside-out approach is proposed by [120], while indicators for benchmarking energy use in manufacturing plants are developed by [121, 122]. It is reported that General Motors reaps significant savings from deploying decision support systems based on real-time control methodology [123–125]. To reduce energy consumption for fixed production volume in automotive paint shops, optimal vehicle batching and the design of spot repair capacity problems are studied by [126, 127]. The productivity, quality, and energy performances are closely related. Indeed, continuous improvement and lean design for

productivity and quality can lead to improvements in energy performance [128, 129]. A review of energy efficiency improvement and cost saving opportunities for the vehicle assembly industry is provided by [130].

In a mass production environment, it is identified by [131] that more than 85% of the energy is utilized for functions that are not directly related to the production of parts. Intuitively, to achieve better energy utilization, the production status of the machines should be continuously monitored. Once a machine becomes idle (or close to being idle), it should be temporarily switched off to reserve energy and switched back on when new jobs arrive. Such production control-based shop floor continuous improvement is recognized as one of the most cost-effective ways to achieve energy-efficient production. Unfortunately, however, very few rigorous studies have been carried out aiming to solve this specific problem, especially for large systems, because of the complexity resulted from the interactions among machines and buffers. In fact, most manufacturing execution systems currently used in practice have no module or function to deal with energy management during operation [64, 132]. Among limited results addressing this issue, an analytical model by combining an M/M/1 model with an energy control policy is developed by [65]. Several machine switching strategies using energy saving opportunity windows under random failures are analyzed by [133]. An integrated serial production line with HVAC system is studied with the same approach by [134]. A control policy to switch off machine tools in a pallet-constrained flow shop is proposed by [135]. For the problem of scheduling startup and shutdown of machines in Bernoulli serial lines, a constrained optimization problem is formulated and studied by [66]. In addition, [67] study several switch-off dispatching policies for a non-bottleneck machine in a job shop to minimize its energy consumption. The problem of saving energy in a single machine visited by single part type with stochastic inter-arrival times has been studied by [136, 137]. Several control policies are proposed for switching the machine off and on. This research is extended by [138, 139] to include buffer information when deciding the switch control of the machine.

It should be noted that when considering production systems with machine switch-on/off control, in addition to the energy consumption performance, it is necessary to study the impacts on other system performance measures as well. This will bring important practical insights for continuous improvement and design of production systems. Moreover, controlling the machine operations may incur other complicating phenomena that need to be considered. For example, after a machine is switched on, it usually must undergo a *warm-up* period before the machine becomes available to work; similarly, after being switched off, there is typically a *cool-down* period and the machine cannot be switched on again until the *cool-down* period is finished. Clearly, both *warm-up* and *cool-down* periods should be taken into account when analyzing the system behavior. The effects of *warm-up* and *cool-down* on one-server queueing systems have been discussed in a few early studies by, for instance, [140, 141]. On the other hand, existing results are only applicable to systems with one or two machines, and there still lacks rigorous and systematic analysis of multi-machine systems with switch-on/off control. Therefore, in this Chapter, we study serial production lines with Bernoulli machines and finite capacity buffers, in which some of the machines can be switched on and off during the production process according to a state-based feedback control policy and with switch-on/off associated *warm-up* and *cool-down* periods.

7.2 Model, Control Rules and Performance Measures

7.2.1 Model

Consider a serial production line in Figure 7.1 defined by the following assumptions:

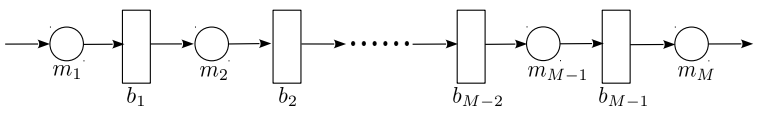


Figure 7.1: Serial production line

(i) The system consists of M machines (represented by circles) and $M-1$ buffers (repre-

sented by rectangles). The arrows indicate the direction of the parts flow.

(ii) The machines have constant and identical cycle time τ. The time axis is slotted with the slot size equal to τ.

(iii) Each buffer, b_i, $i = 1, \ldots, M-1$, is characterized by its capacity, $0 < N_i < \infty$: the largest number of parts that can be stored in b_i.

(iv) A machine has four process status: *warm-up, run, cool-down*, and *sleep*. When machine m_i, $i = 1, \ldots, M$, is in *run* status, it is *up* during a time slot with probability p_i and *down* with probability $1 - p_i$. This is referred to as the *Bernoulli reliability model* and p_i is called the efficiency of m_i.

(v) Machine m_i, $i = 2, \ldots, M$, is starved during a time slot if it is *up* and buffer b_{i-1} is empty at the beginning of this time slot. It is assumed that m_1 is never starved, i.e., infinite raw material supplies.

(vi) Machine m_i, $i = 1, \ldots, M-1$, is blocked during a time slot if it is *up*, buffer b_i has N_i parts at the beginning of the time slot and machine m_{i+1} does not take a part (due to breakdown, blockage, or in *warm-up, cool-down, sleep* status) during that time slot. It is assumed that m_M is never blocked, i.e., infinite finished goods inventory.

(vii) If a machine is *up* during a time slot and *neither starved nor blocked*, then it removes a part from its upstream buffer (raw material supply in case of m_1) at the beginning of the time slot, processes it during the time slot, and places it into its downstream buffer (finished goods inventory in the case of m_M) at the end of the time slot.

(viii) Consider a machine currently in *run* status. The decision of whether or not to switch it off is assessed and actuated between two consecutive time slots, i.e., after all machines have placed the processed parts into the downstream buffers and before any can pick up a new part for the next time slot. If the "switch-off" operation is decided, the machine will enter the *cool-down* status at the beginning the next time slot and remain in *cool-down* status for a total of t_{cd} time slots. After *cool-down* is finished, the machine enters

the *sleep* status if no "switch-on" command is received during this time period. When the machine is waked up from *sleep* by a "switch-on" command, it enters the *warm-up* status of duration t_{wu} time slots before returning to *run* status. Instead of only one machine being controlled in our previous work, let **MC** denote the set of the indices of the controlled machines and let $i^*, j^* \in \mathbf{MC}$. Assume that

$$\begin{aligned} &|i^* - j^*| > 1, \qquad \text{if} \quad 1 < i^*, j^* < M,\ i^* \neq j^*, \\ &|i^* - j^*| > 2, \qquad \text{if} \quad \text{either } i^* \text{ or } j^* \in \{1, M\},\ i^* \neq j^*. \end{aligned} \tag{7.1}$$

In all situations, the machines other than the controlled ones are always in *run* status.

(ix) A machine consumes energy in one of the following six power status: *warm-up, operation, idle, breakdown, cool-down, sleep*. If a machine is in *warm-up, cool-down*, or *sleep* process status, it is also in the same corresponding power status. If a machine is in *run* process status, and neither blocked nor starved, it is in *operation* power status; if it is *up* in *run* status but either blocked or starved, it is in *idle* power status; if a machine is *down* in *run* status, it is in *breakdown* power status. The energy consumption per time slot in the six power status are characterized by e_{wu}, e_{op}, e_{id}, e_{bd}, e_{cd} and e_s, respectively.

Note that both the warm-up and cool-down times are assumed to be deterministic and constant, which is typical is a number of industrial scenarios [66,67]. Moreover, the durations of warm-up and cool-down are assumed to be no shorter than the cycle time, which is also common in manufacturing with, for instance, transitions between hot idle mode and cold idle mode under high thermal inertia [138]. Finally, condition (7.1) in assumption (viii) indicates that it is only possible to control the switch-on/off operations of a subset of all machines in the system. According to this assumption, in two- and three-machine lines, the switch-on/off of only one machine is controlled; while in $M > 3$-machine lines, only non-consecutive internal machines may have controlled switch-on/offs; and if m_1 or m_M is one of the switch-on/off

controlled machines, its closest two downstream or upstream machines cannot be controlled. This condition is introduced to enable analytical study of the system in subsequent sections. When this condition is removed, the switch-on/off control of consecutive machines becomes much more complicated. Although it might be possible to convert such a case by adding a virtual machine in-between, in-depth analysis of systems with this condition relaxed will be carried out in future work.

7.2.2 Control rule

The feedback control rule of machine switch-on/off operations in this Chapter is based on system states. Since the Bernoulli reliability model is memoryless, the production system defined above is characterized by an ergodic Markov chain and the system state consists of all buffers' occupancy and the status of all the controlled machines. Therefore, strictly speaking, the control rule should be determined based on all these information. In this Chapter, for simplicity, consider the case, where the switch-on/off control of a machine is decided based on the occupancy of the buffers and only its own process status, and independent of the status of other machines. More general cases by considering the complete system state will be studied in future work.

To formulate the switch-on/off control rule, let $\mathbf{h}(n) = [h_1(n)\ h_2(n)\ ...\ h_{M-1}(n)]$, where $h_i(n), i = 1,\ ...,\ M-1$, denotes the number of parts in buffer b_i at the end of time slot n. In addition, let $PS^{m_{i^*}}(n) \in \{cd_1, cd_2, ..., cd_{t_{cd}}, sleep, wu_1, wu_2, ..., wu_{t_{wu}}, run\}$, $i^* \in \mathbf{MC}$, denote the state of controlled machine m_{i^*} during time slot n. Note that the state includes the machine's process status as well as the elapsed time during *warm-up* and *cool-down*. Specifically, *sleep* and *run* indicate that the machine is in the corresponding process status, respectively, while cd_k and wu_k represent the states when the machine is in the k-th time slot of *cool-down* and *warm-up*, respectively. Since we assume above that the switch-on/off decisions of machine m_{i^*}, $i^* \in \mathbf{MC}$, are independent of other machines, let $\mathbf{H}_{\text{on}}^{m_{i^*}}$ and $\mathbf{H}_{\text{off}}^{m_{i^*}}$ denote the sets of buffer occupancies, under which machine m_{i^*} is switched on and off,

respectively. Then, machine m_{i^*} is

- switched on at the beginning of time slot $n+1$, if $PS^{m_{i^*}}(n) \in \{cd_{t_{cd}}, sleep\}$ and $\mathbf{h}(n) \in \mathbf{H}_{\text{on}}^{m_{i^*}}$;
- switched off at the beginning of time slot $n+1$, if $PS^{m_{i^*}}(n) = run$ and $\mathbf{h}(n) \in \mathbf{H}_{\text{off}}^{m_{i^*}}$.

In this Chapter we study the systems with a threshold-based feedback control rule formulated as follows to regulate the switch-on/off operations of machine m_{i^*}:

Control Rule 1:

- For $m_{i^*} = m_1$, i.e., $1 \in \mathbf{MC}$,

$$\mathbf{H}_{\text{on}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{on}}^{m_1}, h_2(n) \leqslant h_{2,\text{on}}^{m_1}, ..., h_{M-1}(n) \leqslant h_{M-1,\text{on}}^{m_1}\},$$

$$\mathbf{H}_{\text{off}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{off}}^{m_1}, h_2(n) \geqslant h_{2,\text{off}}^{m_1}, ..., h_{M-1}(n) \geqslant h_{M-1,\text{off}}^{m_1}\};$$

- For $m_{i^*} = m_M$, i.e., $M \in \mathbf{MC}$,

$$\mathbf{H}_{\text{on}}^{m_M} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{on}}^{m_M}, h_2(n) \geqslant h_{2,\text{on}}^{m_M}, ..., h_{M-1}(n) \geqslant h_{M-1,\text{on}}^{m_M}\},$$

$$\mathbf{H}_{\text{off}}^{m_M} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{off}}^{m_M}, h_2(n) \leqslant h_{2,\text{off}}^{m_M}, ..., h_{M-1}(n) \leqslant h_{M-1,\text{off}}^{m_M}\};$$

- For m_{i^*}, $i^* \in \mathbf{MC}$ and $2 \leqslant i^* \leqslant M-1$,

$$\mathbf{H}_{\text{on}}^{m_{i^*}} =$$

$$\{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{on}}^{m_{i^*}}, ..., h_{i^*-1}(n) \geqslant h_{i^*-1,\text{on}}^{m_{i^*}}, h_{i^*}(n) \leqslant h_{i^*,\text{on}}^{m_{i^*}}, ..., h_{M-1}(n) \leqslant h_{M-1,\text{on}}^{m_{i^*}}\},$$

$$\mathbf{H}_{\text{off}}^{m_{i^*}} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{off}}^{m_{i^*}}, ..., h_{i^*-1}(n) \leqslant h_{i^*-1,\text{off}}^{m_{i^*}}\}$$

$$\cup \{\mathbf{h}(n) | h_{i^*}(n) \geqslant h_{i^*,\text{off}}^{m_{i^*}}, ..., h_{M-1}(n) \geqslant h_{M-1,\text{off}}^{m_{i^*}}\},$$

where thresholds $h_{j,\text{on}}^{m_{i^*}}$ and $h_{j,\text{off}}^{m_{i^*}}$ are control parameters.

The intuition of this control rule is as follows: Note that the objective of implementing the switch-on/off control is to reserve energy by eliminating the "idle" periods of certain machines. Clearly, when controlling m_1, for instance, its only "idleness" comes from the

blockage due to lack of available space in downstream buffers. Since all buffers play a role in this regard, we switch it off when the WIPs in all buffers exceed certain levels. This is similar when the switch-on/off of m_M is controlled. On the other hand, an internal machine being idle may imply either starvation or blockage. Therefore, its switch-off should be designed to avoid both. As a result, the switch-off of an internal machine is triggered when it approaches either starvation (due to upstream buffers) *or* blockage (due to downstream buffers). Finally, when the controlled machine is in *cool-down* or *sleep* status, it will not be waked up (i.e., switched on) until its upstream buffers have sufficient parts *and* its downstream buffers have enough storing space to avoid starvations and blockages. It should be noted that similar threshold-based policies are commonly used in production control [143–146], and therefore, are adopted in this study. Other control policies as well as adaptive control will be investigated in future work.

7.2.3 Performance measures

In addition to the common performance measures usually addressed, in this Chapter, the more comprehensive performance measures of the systems are studied:

$Production\ Rate, PR(n)$=the expected number of finished parts produced by m_M in time slot $n+1$;

$Consumption\ Rate, CR(n)$=the expected number of raw parts consumed by m_1 in time slot $n+1$;

$Work\ In\ Process, WIP_i(n)$=the expected number of parts in buffer b_i at the beginning of time slot $n+1$;

$Machine\ Starvation, ST_i(n)$=the probability that machine m_i is starved in time slot $n+1$;

$Machine\ Blockage, BL_i(n)$= the probability that machine m_i is blocked in time slot $n+1$;

$Power, POW_i(n)$= the expected energy consumption of machine m_i in time slot $n+1$.

Note also that in steady state, i.e., when $n \to \infty$, the conservation law holds:

$$\lim_{n\to\infty} PR(n) = \lim_{n\to\infty} CR(n).$$

The energy performance of the system is defined and calculated during steady state as:

Average Energy Consumption, AE = the average energy consumption per finished part

$$= \lim_{n\to\infty} \frac{\sum_{i=1}^{M} POW_i(n)}{PR(n)}.$$

In the following sections, we first apply Markovian analysis to two- and three-machine lines to derive equations for calculating these performance measures. Then, for longer lines, an aggregation-based approximation algorithm is proposed and validated through numerical experiments.

7.3 Exact Performance Analysis for Two- and Three-machine Lines

According to the model assumptions, in two- and three-machine lines, we only consider the cases where the switch-on/off of just one machine is controlled. Moreover, the controlled machine must be in *cool-down* or *sleep* (*run*) status to activate switch-on (switch-off) operations. For two- and three-machine lines, the control rule becomes:

Control Rule 2 (for two-machine lines):

- If $\mathbf{MC} = \{1\}$, i.e., $m_{i^*} = m_1$,
  $\mathbf{H}_{\text{on}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{on}}^{m_1}\}$, $\mathbf{H}_{\text{off}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{off}}^{m_1}\}$;

- If $\mathbf{MC} = \{2\}$, i.e., $m_{i^*} = m_2$,
  $\mathbf{H}_{\text{on}}^{m_2} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{on}}^{m_2}\}$, $\mathbf{H}_{\text{off}}^{m_2} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{off}}^{m_2}\}$.

Control Rule 3 (for three-machine lines):

- If $\mathbf{MC} = \{1\}$, i.e., $m_{i^*} = m_1$,

  $\mathbf{H}_{\text{on}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{on}}^{m_1}, h_2(n) \leqslant h_{2,\text{on}}^{m_1}\}$,
  $\mathbf{H}_{\text{off}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{off}}^{m_1}, h_2(n) \geqslant h_{2,\text{off}}^{m_1}\}$;

- If $\mathbf{MC} = \{2\}$, i.e., $m_{i^*} = m_2$,

  $\mathbf{H}_{\text{on}}^{m_2} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{on}}^{m_2}, h_2(n) \leqslant h_{2,\text{on}}^{m_2}\}$,

  $\mathbf{H}_{\text{off}}^{m_2} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{off}}^{m_2}\} \cup \{\mathbf{h}(n) | h_2(n) \geqslant h_{2,\text{off}}^{m_2}\}$;

- If $\mathbf{MC} = \{3\}$, i.e., $m_{i^*} = m_3$,

  $\mathbf{H}_{\text{on}}^{m_3} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{on}}^{m_3}, h_2(n) \geqslant h_{2,\text{off}}^{m_3}\}$,
  $\mathbf{H}_{\text{off}}^{m_3} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{off}}^{m_3}, h_2(n) \leqslant h_{2,\text{off}}^{m_3}\}$.

7.3.1 Two-machine line case

(i) Switch-on/off Control of m_1 in Two-Machine Line Case

According to Control Rule 2, when the switch-on/off of m_1 is controlled, we

- switch m_1 on at the beginning of time slot $n+1$ if $PS^{m_1}(n) \in \{cd_{t_{cd}}, sleep\}$ and $h_1(n) \leqslant h_{1,\text{on}}^{m_1}$;

- switch m_1 off at the beginning of time slot $n+1$ if $PS^{m_1}(n) = run$ and $h_1(n) \geqslant h_{1,\text{off}}^{m_1}$.

Clearly, the buffer occupancy h_1 can never exceed $h_{1,\text{off}}^{m_1}$ based on the control rule. Thus, without loss of generality, assume $0 \leqslant h_{1,\text{on}}^{m_1} \leqslant h_{1,\text{off}}^{m_1} \leqslant N_1$. Define one *control cycle* as the time interval starting from a switch-off command until the next switch-off command.

To illustrate the operation of such a system, consider a two-machine line with the parameters as follows: $p_1 = 0.91$, $p_2 = 0.79$, $N_1 = 10$. In addition, assume $m_{i^*} = m_1$, $PS^{m_1}(0) = run$, $h_1(0) = 10$, $t_{wu} = 2$, $t_{cd} = 2$ and $h^{m_1}_{1,\text{off}} = 10$, $h^{m_1}_{1,\text{on}} = 5$. The evolution of the buffer occupancy in one typical *control cycle* is shown in Figure 7.2. Since it starts with

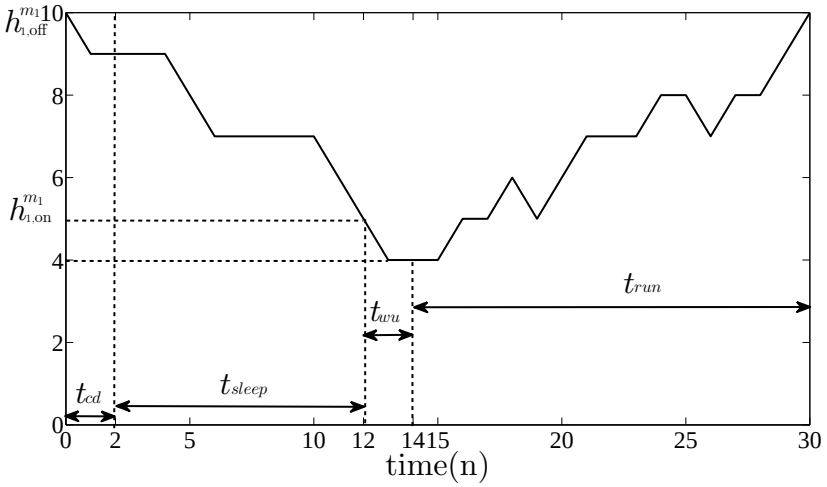


Figure 7.2: Evolution of buffer occupancy in one control cycle ($m_{i^*} = m_1$)

the buffer containing 10 parts and machine m_1 in *run* status, m_1 is switched off according to the control policy at the beginning of the first time slot. Then, m_1 goes into *cool-down* and only m_2 is in *run* status so that the parts in the buffer are consumed by m_2. After the next $t_{cd} = 2$ time slots, m_1 finishes *cool-down*, goes into *sleep* and stays in *sleep* until the buffer occupancy decreases to $h^{m_1}_{1,\text{on}} = 5$ parts. In this example, it occurs at the end of time slot 12 (see Figure 7.2). As a result, m_1 is switched on at the beginning of time slot 13 and goes into *warm-up* for $t_{wu} = 2$ time slots before it can process any part. During its *warm-up* period, m_2 is up during time slot 13 and down during time slot 14. Therefore, when m_1 enters *run* status in time slot 15, the buffer has 4 parts remained. From this point on, the buffer occupancy evolves based on the reliability status of both machines. The current *control cycle* finishes at the end of time slot 30 and a new *control cycle* will follow.

To proceed with the analysis, note that, the system state consists of two components: the buffer occupancy h_1 and the state of the controlled machine PS^{m_1} defined in Subsection 7.2.2. The possible range of the system state is given by:

$$(h_1, PS^{m_1}) \in \{0, 1, ..., N_1\} \times \{cd_1, cd_2, ..., cd_{t_{cd}}, sleep, wu_1, wu_2, ..., wu_{t_{wu}}, run\}. \tag{7.2}$$

Therefore, the total number of the system states is:

$$S = (N_1 + 1) \times (t_{cd} + t_{wu} + 2).$$

Then, one can construct bijections $\alpha_2(\cdot)$ between the set of all system states to the set of positive integers $\{1, 2, ..., S\}$ to assign a specific state number to each system state. The bijection $\alpha_2(\cdot)$ from state (h_1, PS^{m_1}) to its assigned state number can be defined as follows:

$$\begin{aligned}
&\alpha_2(h_1, cd_k) = (N_1 + 1) \times (k - 1) + h_1 + 1, \quad (1 \leqslant k \leqslant t_{cd});\\
&\alpha_2(h_1, sleep) = (N_1 + 1) \times t_{cd} + h_1 + 1;\\
&\alpha_2(h_1, wu_k) = (N_1 + 1) \times (t_{cd} + k) + h_1 + 1, \quad (1 \leqslant k \leqslant t_{wu});\\
&\alpha_2(h_1, run) = (N_1 + 1) \times (t_{cd} + t_{wu}) + h_1 + 1.
\end{aligned} \tag{7.3}$$

Let $s(n)$ denote the system state at time slot n indicated based on state number assignment (7.3). The transition probabilities among the system states can be obtained as follows:

$$\begin{aligned}
&P[s(n+1) = i | s(n) = j] = 1 - p_2, \quad (i, j) \in \{(\alpha_2(h_1, cd_{k1+1}), \alpha_2(h_1, cd_{k1})),\\
&\quad (\alpha_2(h_1, wu_{k2+1}), \alpha_2(h_1, wu_{k2})), (\alpha_2(h_1, run), \alpha_2(h_1, wu_{t_{wu}}))\},\\
&\quad h_1 > 0,\ 0 < k1 < t_{cd}, 0 < k2 < t_{wu};\\
&P[s(n+1) = i | s(n) = j] = p_2, \quad (i, j) \in \{(\alpha_2(h_1 - 1, cd_{k1+1}), \alpha_2(h_1, cd_{k1})),\\
&\quad (\alpha_2(h_1 - 1, wu_{k2+1}), \alpha_2(h_1, wu_{k2})),\ (\alpha_2(h_1 - 1, run), \alpha_2(h_1, wu_{t_{wu}}))\},\\
&\quad h_1 > 0,\ 0 < k1 < t_{cd}, 0 < k2 < t_{wu};\\
&P[s(n+1) = i | s(n) = j] = 1, \quad (i, j) \in \{(\alpha_2(0, cd_{k1+1}), \alpha_2(0, cd_{k1})),
\end{aligned}$$

$$
\begin{aligned}
&(\alpha_2(0, wu_{k2+1}), \alpha_2(0, wu_{k2})), (\alpha_2(0, run), \alpha_2(0, wu_{t_{wu}}))\},\ 0 < k1 < t_{cd}, 0 < k2 < t_{wu};\\
&P[s(n+1) = i | s(n) = j] = 1 - p_2, \quad (i,j) \in \{(\alpha_2(h_1, wu_1), \alpha_2(h_1, cd_{t_{cd}}))\}, \quad h_1 \leqslant h_{1,\text{on}}^{m_1};\\
&P[s(n+1) = i | s(n) = j] = 1 - p_2, \quad (i,j) \in \{(\alpha_2(h_1, sleep), \alpha_2(h_1, sleep)),\\
&\quad (\alpha_2(h_1, sleep), \alpha_2(h_1, cd_{t_{cd}}))\}, \quad h_1 > h_{1,\text{on}}^{m_1};\\
&P[s(n+1) = i | s(n) = j] = p_2, \quad (i,j) \in \{(\alpha_2(h_1, wu_1), \alpha_2(h_1+1, cd_{t_{cd}})),\\
&\quad (\alpha_2(h_1, wu_1), \alpha_2(h_1+1, sleep))\}, \quad h_1 \leqslant h_{1,\text{on}}^{m_1};\\
&P[s(n+1) = i | s(n) = j] = p_2, \quad (i,j) \in \{(\alpha_2(h_1, sleep), \alpha_2(h_1+1, sleep))\}, \quad h_1 > h_{1,\text{on}}^{m_1};\\
&P[s(n+1) = i | s(n) = j] = 1 - p_1, \quad (i,j) \in \{(\alpha_2(0, run), \alpha_2(0, run))\};\\
&P[s(n+1) = i | s(n) = j] = p_1, \quad (i,j) \in \{(\alpha_2(1, run), \alpha_2(0, run))\};\\
&P[s(n+1) = i | s(n) = j] = p_1 p_2 + (1-p_1)(1-p_2),\\
&\quad (i,j) \in \{\alpha_2(h_1, run), \alpha_2(h_1, run)\}, 1 \leqslant h_1 \leqslant h_{1,\text{off}}^{m_1} - 1;\\
&P[s(n+1) = i | s(n) = j] = (1-p_1)p_2,\\
&(i,j) \in \{(\alpha_2(h_1 - 1, run), \alpha_2(h_1, run))\}, 1 \leqslant h_1 \leqslant h_{1,\text{off}}^{m_1} - 1;\\
&P[s(n+1) = i | s(n) = j] = p_1(1-p_2),\\
&(i,j) \in \{(\alpha_2(h_1 + 1, run), \alpha_2(h_1, run))\}, 1 \leqslant h_1 \leqslant h_{1,\text{off}}^{m_1} - 2;\\
&P[s(n+1) = i | s(n) = j] = p_1(1-p_2),\ (i,j) \in \{(\alpha_2(h_{1,\text{off}}^{m_1}, cd_1), \alpha_2(h_{1,\text{off}}^{m_1} - 1, run))\}.
\end{aligned}
\tag{7.4}
$$

Let $\mathbf{x}(n) = [x_1(n)\ x_2(n)\ ...\ x_S(n)]^T$, where $x_i(n) = P[s(n) = i]$, denote the probability distribution of the system states at the end of time slot n. The evolution of $\mathbf{x}(n)$, can be described by the following linear time-invariant equation with initial condition:

$$
\mathbf{x}(n+1) = \mathbf{A}_2^{m_1}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \quad x_{\alpha_2(\mu,\nu)}(0) = \begin{cases} 1, & \mu = h_1(0),\ \nu = PS^{m_1}(0), \\ 0, & \text{otherwise}, \end{cases}
\tag{7.5}
$$

where $\mathbf{A}_2^{m_1}$ is the transition probability matrix calculated based on equation (7.4). Finally, the performance measures of the system can be calculated as:

$$PR(n) = P[m_2 \text{ is } up, b_1 \text{ is not empty in time slot } n] = p_2 P[h_1(n) > 0] = \mathbf{V}_1^{2,m_1}\mathbf{x}(n), \tag{7.6}$$

$$\begin{aligned} CR(n) &= P[m_1 \text{ is } up \text{ in } run \text{ status and not blocked in time slot } n] \\ &= p_1 P[m_1 \text{ is in } run \text{ status in time slot } n] P[h_1(n) < N_1] \\ &\quad + p_1 P[m_1 \text{ is in } run \text{ status in time slot } n] p_2 P[h_1(n) = N_1] = \mathbf{V}_2^{2,m_1}\mathbf{x}(n), \end{aligned} \tag{7.7}$$

$$WIP(n) = \sum_{k=0}^{N_1} kP[h_1(n) = k] = \mathbf{V}_3^{2,m_1}\mathbf{x}(n), \tag{7.8}$$

$$\begin{aligned} BL_1(n) &= P[m_1 \text{ is } up \text{ in } run \text{ status}, b_1 \text{ is full}, \ m_2 \text{ is } down \text{ in time slot } n] \\ &= p_1 P[m_1 \text{ is } up \text{ in } run \text{ status}](1 - p_2) P[h_1(n) = N_1] = \mathbf{V}_4^{2,m_1}\mathbf{x}(n), \end{aligned} \tag{7.9}$$

$$ST_2(n) = P[m_2 \text{ is } up, b_1 \text{ is empty in time slot } n] = p_2 P[h_1(n) = 0] = \mathbf{V}_5^{2,m_1}\mathbf{x}(n), \tag{7.10}$$

$$\begin{aligned} POW_1(n) &= e_{wu} P[m_1 \text{ is in } warm\text{-}up \text{ status in time slot } n] \\ &\quad + e_{op} P[m_1 \text{ is } up \text{ in } run \text{ status and not blocked in time slot } n] \\ &\quad + e_{id} P[m_1 \text{ is } up \text{ in } run \text{ status and blocked in time slot } n] \\ &\quad + e_{bd} P[m_1 \text{ is } down \text{ in } run \text{ status in time slot } n] \\ &\quad + e_{cd} P[m_1 \text{ is in } cool\text{-}down \text{ status in time slot } n] \\ &\quad + e_s P[m_1 \text{ is in } sleep \text{ status in time slot } n] \\ &= \mathbf{V}_{6,1}^{2,m_1}\mathbf{x}(n), \end{aligned} \tag{7.11}$$

$$\begin{aligned} POW_2(n) &= e_{op} P[m_2 \text{ is } up \text{ and not starved in time slot } n] \\ &\quad + e_{id} P[m_2 \text{ is } up \text{ and starved in time slot } n] + e_{bd} P[m_2 \text{ is } down \text{ in time slot } n] \\ &= \mathbf{V}_{6,2}^{2,m_1}\mathbf{x}(n), \end{aligned} \tag{7.12}$$

where

$$
\begin{aligned}
&\mathbf{V}_1^{2,m_1} = [0 \quad p_2\mathbf{J}_{1,N_1}]\mathbf{C}_{(N_1+1)\times S}, && \mathbf{V}_2^{2,m_1} = [\mathbf{0}_{1,S-N_1-1} \quad p_1\mathbf{J}_{1,N_1} \quad p_1p_2],\\
&\mathbf{V}_3^{2,m_1} = [0 \quad 1 \quad ... \quad N_1]\mathbf{C}_{(N_1+1)\times S}, && \mathbf{V}_4^{2,m_1} = [\mathbf{0}_{1,S-1} \quad p_1(1-p_2)],\\
&\mathbf{V}_5^{2,m_1} = [p_2 \quad \mathbf{0}_{1,N_1}]\mathbf{C}_{(N_1+1)\times S},\\
&\mathbf{V}_{6,1}^{2,m_1} = [e_{cd}\mathbf{J}_{1,(N_1+1)t_{cd}} \quad e_s\mathbf{J}_{1,N_1+1} \quad e_{wu}\mathbf{J}_{1,(N_1+1)t_{wu}} \quad (p_1e_{op}+(1-p_1)e_{bd})\mathbf{J}_{1,N_1}\\
&\qquad\qquad p_1p_2e_{op}+(1-p_1)e_{bd}+p_1(1-p_2)e_{id})],\\
&\mathbf{V}_{6,2}^{2,m_1} = [p_2e_{id}+(1-p_2)e_{bd} \quad (p_2e_{op}+(1-p_2)e_{bd})\mathbf{J}_{1,N_1}]\mathbf{C}_{(N_1+1)\times S},
\end{aligned}
\tag{7.13}
$$

$\mathbf{0}_{1,k}$ and $\mathbf{J}_{1,k}$ denote the 1-by-k matrices of zeros and ones, respectively. Moreover, for j divisible by i, i-by-j matrix $\mathbf{C}_{i\times j} = [\mathbf{I}_i \cdots \mathbf{I}_i]$ represents the matrix consisting of j/i identity matrices $\mathbf{I}_i$.

(ii) Switch-on/off Control of m_2 in Two-Machine Line Case

When the switch-on/off of m_2 is controlled, the control policy becomes:

- switch m_2 on at the beginning of time slot $n+1$ if $PS^{m_2}(n) \in \{cd_{t_{cd}}, sleep\}$ and $h_1(n) \geqslant h_{1,\text{on}}^{m_2}$;

- switch m_2 off at the beginning of time slot $n+1$ if $PS^{m_2}(n) = run$ and $h_1(n) \leqslant h_{1,\text{off}}^{m_2}$.

Clearly, the buffer occupancy h_1 can never be lower than $h_{1,\text{off}}^{m_2}$ based on the control rule. Therefore, assume $0 \leqslant h_{1,\text{off}}^{m_2} \leqslant h_{1,\text{on}}^{m_2} \leqslant N_1$. To illustrate the operation of such a system, consider a two-machine line with the parameters as follows: $p_1 = 0.76,\ p_2 = 0.89,\ N_1 = 10$. In addition, assume $m_{i^*} = m_2$, $PS^{m_2}(0) = run$, $h_1(0) = 0$, $t_{wu} = 2,\ t_{cd} = 2$, and $h_{1,\text{off}}^{m_2} = 0$, $h_{1,\text{on}}^{m_2} = 5$, which conform to the inequality $0 \leqslant h_{1,\text{off}}^{m_2} \leqslant h_{1,\text{on}}^{m_2} \leqslant N_1$. The evolution of the buffer occupancy in a typical *control cycle* is shown in Figure 7.3.

Since it starts with an empty buffer at time 0 and machine m_2 is in *run* status, m_2 is switched off according to the control policy at the beginning of the control cycle. After the switch-off operation, m_2 goes into *cool-down* status and only m_1 is in *run* status so that

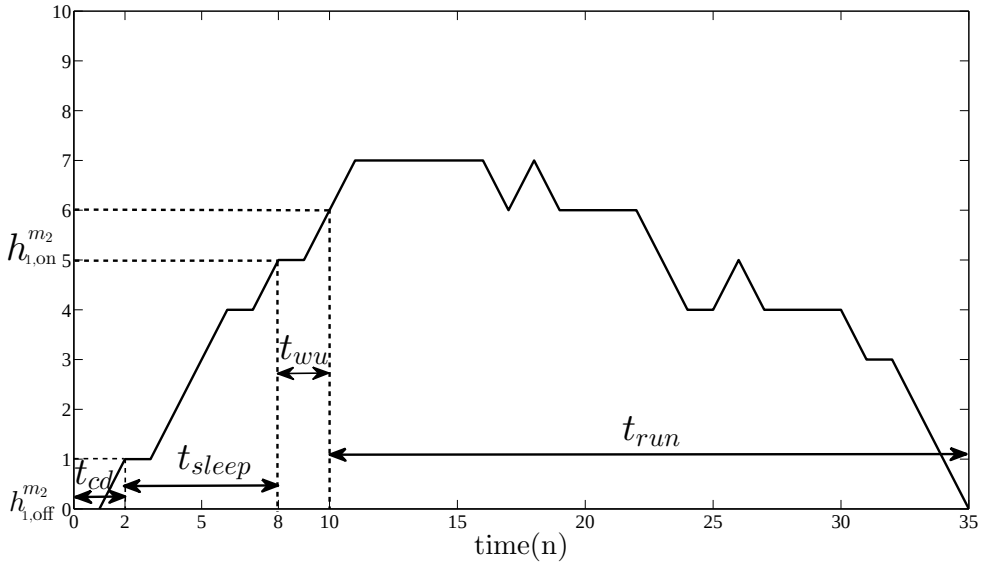


Figure 7.3: Evolution of buffer occupancy in one control cycle ($m_{i^*} = m_2$)

the parts will be produced and put into the buffer by m_1. After the next $t_{cd} = 2$ time slots following the switch-off, m_2 finishes *cool-down* and goes into the *sleep* status and will stay in the *sleep* status until buffer occupancy increases to $h_{1,\text{on}}^{m_2} = 5$ parts. In this example, it takes place at the end of time slot 8 as shown in Figure 7.3. As a result, m_2 is switched on at the beginning of time slot 9 and goes into *warm-up* for $t_{wu} = 2$ time slots before it can process any part. During its *warm-up* period, m_1 is down during time slot 9 and up during time slot 10. Therefore, when m_2 enters *run* status in time slot 11, the buffer has 6 parts. From this point on, the buffer occupancy evolves based on the reliability status of the two Bernoulli machines. Finally, the current *control cycle* finishes when the buffer occupancy hits $h_{1,\text{off}}^{m_2}$ again at the end of time slot 35 and a new *control cycle* will start.

To analyze such a system, the possible range of the system state is given by:

$$(h_1, PS^{m_2}) \in \{0, 1, ..., N_1\} \times \{cd_1, cd_2, ..., cd_{t_{cd}}, sleep, wu_1, wu_2, ..., wu_{t_{wu}}, run\}. \quad (7.14)$$

Therefore, the total number of the system states is:

$$S = (N_1 + 1) \times (t_{cd} + t_{wu} + 2).$$

Then, one can assign a unique state number from the set $\{1, 2, \ldots, S\}$ to each of the system state based on bijection $\alpha_2(\cdot)$ defined by equation (7.3). Then, under this state number assignment, the transition probabilities among the system states in such case can be calculated as:

$$P_{i,j}[s(n+1) = i | s(n) = j] = 1 - p_1, \quad (i,j) = \{(\alpha_2(h_1, cd_{k1+1}), \alpha_2(h_1, cd_{k1})),$$
$$(\alpha_2(h_1, wu_{k2+1}), \alpha_2(h_1, wu_{k2})), (\alpha_2(h_1, run), \alpha_2(h_1, wu_{t_{wu}}))\},$$
$$h_1 < N_1, 0 < k1 < t_{cd}, 1 < k2 < t_{wu};$$

$$P[s(n+1) = i | s(n) = j] = p_1, \quad (i,j) \in \{(\alpha_2(h_1 + 1, cd_{k1+1}), \alpha_2(h_1, cd_{k1})),$$
$$(\alpha_2(h_1 + 1, wu_{k2+1}), \alpha_2(h_1, wu_{k2})), (\alpha_2(h_1 + 1, run), \alpha_2(h_1, wu_{t_{wu}}))\},$$
$$h_1 < N_1, 0 < k1 < t_{cd}, 1 < k2 < t_{wu};$$

$$P_{i,j}[s(n+1) = i | s(n) = j] = 1, \quad (i,j) = \{(\alpha_2(N_1, cd_{k1+1}), \alpha_2(N_1, cd_{k1})),$$
$$(\alpha_2(N_1, wu_{k2+1}), \alpha_2(N_1, wu_{k2})), (\alpha_2(N_1, run), \alpha_2(N_1, wu_{t_{wu}}))\}, 0 < k1 < t_{cd}, 1 < k2 < t_{wu};$$

$$P[s(n+1) = i | s(n) = j] = 1 - p_1, \quad (i,j) \in \{(\alpha_2(h_1, wu_1), \alpha_2(h_1, cd_{t_{cd}}))\}, \quad h_1 \geqslant h_{1,\text{on}}^{m_2};$$

$$P[s(n+1) = i | s(n) = j] = 1 - p_1, \quad (i,j) \in \{(\alpha_2(h_1, sleep), \alpha_2(h_1, cd_{t_{cd}})),$$
$$(\alpha_2(h_1, sleep), \alpha_2(h_1, sleep))\}, \quad h_1 < h_{1,\text{on}}^{m_2};$$

$$P[s(n+1) = i | s(n) = j] = p_1, \quad (i,j) \in \{(\alpha_2(h_1, wu_1), \alpha_2(h_1 - 1, cd_{t_{cd}})),$$
$$(\alpha_2(h_1, wu_1), \alpha_2(h_1 - 1, sleep))\}, \quad h_1 \geqslant h_{1,\text{on}}^{m_2};$$

$$P[s(n+1) = i | s(n) = j] = p_1, \quad (i,j) \in \{(\alpha_2(h_1, sleep), \alpha_2(h_1 - 1, sleep))\}, \quad h_1 < h_{1,\text{on}}^{m_2};$$

$$P[s(n+1) = i | s(n) = j] = p_1 p_2 + (1 - p_1)(1 - p_2),$$
$$(i,j) \in \{\alpha_2(h_1, run), \alpha_2(h_1, run))\}, h_{1,\text{off}}^{m_2} + 1 \leqslant h_1 \leqslant N_1 - 1;$$

$$P[s(n+1) = i | s(n) = j] = (1 - p_1)p_2,$$
$$(i,j) \in \{(\alpha_2(h_1, run), \alpha_2(h_1 + 1, run)\}, h_{1,\text{off}}^{m_2} + 1 \leqslant h_1 \leqslant N_1 - 1;$$

$$
\begin{aligned}
&P_{i,j}[s(n+1)=i|s(n)=j]=p_1(1-p_2),\\
&\quad (i,j)=\{\alpha_2(h_1+1,run),\alpha_2(h_1,run)\}, h_{1,\text{off}}^{m_2}+1\leqslant h_1\leqslant N_1-1;\\
&P[s(n+1)=i|s(n)=j]=p_1p_2+1-p_2,\quad (i,j)\in\{\alpha_2(N_1,run),\alpha_2(N_1,run)\};\\
&P[s(n+1)=i|s(n)=j]=(1-p_1)p_2,\quad (i,j)\in\{\alpha_2(h_{1,\text{off}}^{m_2},cd_1),\alpha_2(h_{1,\text{off}}^{m_2}+1,run)\}.
\end{aligned}
\tag{7.15}
$$

Let $\mathbf{x}(n)=[x_1(n)\ x_2(n)\ ...\ x_S(n)]^T$, where $x_i(n)=P[s(n)=i]$, denote the probability distribution of the system states at the end of time slot n. Let $\mathbf{0}_{1,k}$ and $\mathbf{J}_{1,k}$ denote the 1-by-k matrices of zeros and ones, respectively. In addition, for j divisible by i, let $\mathbf{C}_{i\times j}=[\mathbf{I}_i\cdots\mathbf{I}_i]$ represent the matrix consisting of j/i identity matrices $\mathbf{I}_i$. The evolution of $\mathbf{x}(n)$ can be described by the following linear time-invariant equation with initial condition:

$$
\mathbf{x}(n+1)=\mathbf{A}_2^{m_2}\mathbf{x}(n),\quad \sum_{i=1}^{S}x_i(n)=1,\quad x_{\alpha_2(\mu,\nu)}(0)=\begin{cases}1, & \mu=h_1(0),\ \nu=PS^{m_2}(0),\\ 0, & \text{otherwise,}\end{cases}
\tag{7.16}
$$

where $\mathbf{A}_2^{m_2}$ is the transition probability matrix calculated based on equation (7.15). The performance measures can be calculated as:

$$
\begin{aligned}
&PR(n)=\mathbf{V}_1^{2,m_2}\mathbf{x}(n),\quad CR(n)=\mathbf{V}_2^{2,m_2}\mathbf{x}(n),\quad WIP(n)=\mathbf{V}_3^{2,m_2}\mathbf{x}(n),\\
&BL_1(n)=\mathbf{V}_4^{2,m_2}\mathbf{x}(n),\quad ST_2(n)=\mathbf{V}_5^{2,m_2}\mathbf{x}(n),\quad POW_i(n)=\mathbf{V}_{6,i}^{2,m_2}\mathbf{x}(n),\quad i=1,2,
\end{aligned}
\tag{7.17}
$$

where

$$
\begin{aligned}
\mathbf{V}_1^{2,m_2} &= [\mathbf{0}_{1,S-N_1} \quad p_2\mathbf{J}_{1,N_1}],\\
\mathbf{V}_2^{2,m_2} &= [[p_1\mathbf{J}_{1,N_1} \quad 0]\mathbf{C}_{(N_1+1)\times(N_1+1)(t_{cd}+t_{wu}+1)} \quad p_1\mathbf{J}_{1,N_1} \quad p_1p_2],\\
\mathbf{V}_3^{2,m_2} &= [0 \quad 1 \quad ... \quad N_1]\mathbf{C}_{(N_1+1)\times S}\\
\mathbf{V}_4^{2,m_2} &= [[\mathbf{0}_{1,N_1} \quad p_1]\mathbf{C}_{(N_1+1)\times(N_1+1)(t_{cd}+t_{wu}+1)} \quad \mathbf{0}_{1,N_1} \quad p_1(1-p_2)],\\
\mathbf{V}_5^{2,m_2} &= [\mathbf{0}_{1,S-N_1-1} \quad p_2 \quad \mathbf{0}_{1,N_1}],\\
\mathbf{V}_{6,1}^{2,m_2} &= [[(p_1e_{op}+(1-p_1)e_{bd})\mathbf{J}_{1,N_1} \quad p_1e_{id}+(1-p_1)e_{bd}]\mathbf{C}_{(N_1+1)\times(N_1+1)(t_{cd}+t_{wu}+1)}\\
&\qquad (p_1e_{op}+(1-p_1)e_{bd})\mathbf{J}_{1,N_1} \quad p_1p_2e_{op}+p_1(1-p_2)e_{id}+(1-p_1)e_{bd}],\\
\mathbf{V}_{6,2}^{2,m_2} &= [e_{cd}\mathbf{J}_{1,(N_1+1)t_{cd}} \quad e_s\mathbf{J}_{1,N_1+1} \quad e_{wu}\mathbf{J}_{1,(N_1+1)t_{wu}} \quad p_2e_{id}+(1-p_2)e_{bd}\\
&\qquad (p_2e_{op}+(1-p_2)e_{bd})\mathbf{J}_{1,N_1}].
\end{aligned}
\tag{7.18}
$$

7.3.2 Three-machine line case

(i) Switch-on/off Control of m_2 in Three-Machine Line Case

Consider a three-machine line where the switch-on/off operations of m_2 is controlled. Then, Control Rule 3 formulated above can be rewritten as follows:

- switch m_2 on at the beginning of time slot $n+1$ if $PS^{m_2}(n) \in \{cd_{t_{cd}}, sleep\}$, $h_1(n) \geqslant h_{1,\text{on}}^{m_2}$ and $h_2(n) \leqslant h_{2,\text{off}}^{m_2}$;

- switch m_2 off at the beginning of time slot $n+1$ if $PS^{m_2}(n) = run$, and either $h_1(n) \leqslant h_{1,\text{off}}^{m_2}$ or $h_2(n) \geqslant h_{2,\text{off}}^{m_2}$.

To study such a system, note that, all possible combinations of the buffers occupancy can be arranged in Table 7.1, where $S_0 = (N_1+1)\times(N_2+1)$.
The system state can be expressed as a 3-tuple: (h_1, h_2, PS^{m_2}), where $h_1 \in \{0,1,...,N_1\}$, $h_2 \in \{0,1,...,N_2\}$, and $PS^{m_2} \in \{cd_1, cd_2, ..., cd_{t_{cd}}, sleep, wu_1, ..., wu_{t_{wu}}, run\}$. The total number of the system states is $S = S_0 \times (t_{cd}+t_{wu}+2)$.

Table 7.1: Combinations of the buffers occupancy

Number	1	2	...	N_2+1	N_2+2	N_2+3	...	S_0-1	S_0
h_1	0	0	...	0	1	1	...	N_1	N_1
h_2	0	1	...	N_2	0	1	...	N_2-1	N_2

Next, we arrange the states from state 1 to state S based on a bijection $\alpha_3(\cdot)$ from system state (h_1, h_2, PS^{m_2}) to its assigned state number as follows:

$$\begin{aligned}
\alpha_3(h_1,h_2,cd_k) &= (N_1+1)\times(N_2+1)\times(k-1)+h_1\times(N_2+1)+h_2+1,\ (1\leqslant k\leqslant t_{cd}),\\
\alpha_3(h_1,h_2,sleep) &= (N_1+1)\times(N_2+1)\times t_{cd}+h_1\times(N_2+1)+h_2+1,\\
\alpha_3(h_1,h_2,wu_k) &= (N_1+1)\times(N_2+1)\times(t_{cd}+k)+h_1\times(N_2+1)+h_2+1,\ (1\leqslant k\leqslant t_{wu}),\\
\alpha_3(h_1,h_2,run) &= (N_1+1)\times(N_2+1)\times(t_{cd}+t_{wu}+1)+h_1\times(N_2+1)+h_2+1.
\end{aligned} \tag{7.19}$$

According to the assumptions (i)-(viii), the dynamics of the buffer occupancies are given as follows:

$$\begin{aligned}
h_2'(n+1) &= h_2(n)-\beta_3(n+1)\min\{h_2(n),1\},\\
h_1'(n+1) &= h_1(n)-\beta_2(n+1)\cdot\min\{h_1(n),N_2-h_2'(n+1),1\},\\
h_2(n+1) &= h_2'(n+1)+\beta_2(n+1)\cdot\min\{h_1(n),N_2-h_2'(n+1),1\},\\
h_1(n+1) &= h_1'(n+1)+\beta_1(n+1)\cdot\min\{N_1-h_1'(n+1),1\},
\end{aligned} \tag{7.20}$$

where

$$\beta_2(n)=\begin{cases} 1,\ \text{if } m_2 \text{ is } up \text{ in } run \text{ status},\\ 0,\ \text{if } m_2 \text{ is in } warm\text{-}up, cool\text{-}down, sleep \text{ status, or } down \text{ in } run \text{ status}, \end{cases}$$

$$\beta_i(n)\in\begin{cases} 1,\ \text{if } m_i \text{ is } up, \\ 0,\ \text{if } m_i \text{ is } down, \end{cases} \qquad i\in\{1,3\},$$

and $h_i'(n)$ represents the occupancy of buffer b_i as soon as its downstream machine removes

a part from the buffer at the beginning of time slot n.

Note that if the controlled machine m_2 is in *run* status, then there is a total of $2^3 = 8$ possible combinations of the three machines' status (either *up* or *down* for each machine). The probabilities of these combinations are given by

$$P[\beta_1(n), \beta_2(n), \beta_3(n)] = \prod_{i=1}^{3} p_i^{\beta_i(n)}(1-p_i)^{1-\beta_i(n)}, \quad \beta_i(n) \in \{0,1\}. \tag{7.21}$$

If m_2 is in *sleep*, *warm-up* or *cool-down* status, the total number of possible combinations of machines status reduces to $2^2 = 4$, since the state of m_1 is fixed. The probabilities of these combinations are given by

$$P[\beta_1(n), 0, \beta_3(n)] = p_1^{\beta_1(n)}(1-p_1)^{1-\beta_1(n)} p_3^{\beta_3(n)}(1-p_3)^{1-\beta_3(n)}, \quad \beta_i(n) \in \{0,1\}. \tag{7.22}$$

Thus, from any given system state, we can enumerate all possible combinations of machines status and determine the corresponding outcome states using equations (7.20)-(7.22). Then, the combinations of machines status leading to the same outcome state are identified and the probabilities of these combinations are summed to obtain the transition probability from the original state to this particular outcome state. Use $\mathbf{A}_3^{m_2}$ to represent the transition matrix for the three-machine line. Let $\mathbf{x}(n) = [x_1(n)\ x_2(n)\ ...\ x_S(n)]^T$ denote the probability distribution of the system states, where $x_i(n)$=P[system in state i at the end of time slot n]. Then,

$$\begin{aligned} &\mathbf{x}(n+1) = \mathbf{A}_3^{m_2}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \\ &x_{\alpha_3(\mu,\upsilon,\nu)}(0) = \begin{cases} 1, & \mu = h_1(0), \upsilon = h_2(0),\ \nu = PS^{m_2}(0), \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{7.23}$$

Define

$$\mathbf{K}^{3M} = \begin{bmatrix} 0 & 0 & ... & 0 & 1 & 1 & ... & N_1 & N_1 \\ 0 & 1 & ... & N_2 & 0 & 1 & ... & N_2-1 & N_2 \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix}. \tag{7.24}$$

Based on these notations, the performance measures of the three-machine line can be calculated as follows:

$$\begin{aligned} &PR(n) = \mathbf{V}_1^{3,m_2}\mathbf{x}(n), \qquad CR(n) = \mathbf{V}_2^{3,m_2}\mathbf{x}(n), \quad WIP_i(n) = \mathbf{V}_{3,i}^{3,m_2}\mathbf{x}(n), \\ &BL_i(n) = \mathbf{V}_{4,i}^{3,m_2}\mathbf{x}(n), \quad ST_i(n) = \mathbf{V}_{5,i}^{3,m_2}\mathbf{x}(n), \qquad POW_i(n) = \mathbf{V}_{6,i}^{3,m_2}\mathbf{x}(n), \end{aligned} \tag{7.25}$$

where

$$\begin{aligned}
\mathbf{V}_1^{3,m_2} &= [0 \quad p_3\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S}, \\
\mathbf{V}_2^{3,m_2} &= [[p_1\mathbf{J}_{1,N_1(N_2+1)} \ \mathbf{0}_{1,N_2+1}]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad p_1\mathbf{J}_{1,N_1(N_2+1)} \quad p_1p_2\mathbf{J}_{1,N_2} \quad p_1p_2p_3], \\
\mathbf{V}_{3,i}^{3,m_2} &= [\mathbf{K}_i]\mathbf{C}_{S_0\times S}, \quad i = 1, 2, \\
\mathbf{V}_{4,1}^{3,m_2} &= [[\mathbf{0}_{1,N_1(N_2+1)} \quad p_1\mathbf{J}_{1,(N_2+1)}]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad \mathbf{0}_{1,N_1(N_2+1)} \quad p_1(1-p_2)\mathbf{J}_{1,N_2} \\
&\qquad p_1(1-p_2) + p_1p_2(1-p_3)], \\
\mathbf{V}_{4,2}^{3,m_2} &= [\mathbf{0}_{1,S_0(t_{cd}+t_{wu}+1)} \quad [\mathbf{0}_{1,N_2} \quad p_2(1-p_3)]\mathbf{C}_{(N_2+1)\times S_0}], \\
\mathbf{V}_{4,3}^{3,m_2} &= [\mathbf{0}_{1,S}], \\
\mathbf{V}_{5,1}^{3,m_2} &= [\mathbf{0}_{1,S}], \\
\mathbf{V}_{5,2}^{3,m_2} &= [\mathbf{0}_{1,S_0(t_{cd}+t_{wu}+1)} \quad p_2\mathbf{J}_{1,(N_2+1)} \quad \mathbf{0}_{1,N_1(N_2+1)}], \\
\mathbf{V}_{5,3}^{3,m_2} &= [p_3 \quad \mathbf{0}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S}, \\
\mathbf{V}_{6,1}^{3,m_2} &= [[(p_1e_{op} + (1-p_1)e_{bd})\mathbf{J}_{1,N_1(N_2+1)} \quad (p_1e_{id} + (1-p_1)e_{bd})\mathbf{J}_{1,N_2+1}]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \\
&\qquad (p_1e_{op} + (1-p_1)e_{bd})\mathbf{J}_{1,N_1(N_2+1)} \quad (p_1p_2e_{op} + p_1(1-p_2)e_{id} + (1-p_1)e_{bd})\mathbf{J}_{1,N_2} \\
&\qquad (p_1p_2e_{op} + (p_1(1-p_2) + p_1p_2(1-p_3))e_{id} + (1-p_1)e_{bd}], \\
\mathbf{V}_{6,2}^{3,m_2} &= [e_{cd}\mathbf{J}_{1,S_0t_{cd}} \quad e_s\mathbf{J}_{1,S_0} \quad e_{wu}\mathbf{J}_{1,S_0t_{wu}} \quad (p_2e_{id} + (1-p_2)e_{bd})\mathbf{J}_{1,(N_2+1)} \\
&\qquad [(p_2e_{op} + (1-p_2)e_{bd})\mathbf{J}_{1,N_2} \quad p_2p_3e_{op} + p_2(1-p_3)e_{id} + (1-p_2)e_{bd}]\mathbf{C}_{(N_2+1)\times N_1(N_2+1)}], \\
\mathbf{V}_{6,3}^{3,m_2} &= [(p_3e_{id} + (1-p_3)e_{bd} \quad (p_3e_{op} + (1-p_3)e_{bd})\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S}.
\end{aligned} \tag{7.26}$$

The above analysis can be extended to the cases where $\mathbf{MC} = \{1\}$ or $\{3\}$.

(ii) Switch-on/off Control of m_1 or m_3 in Three-Machine Line Case

The analysis presented above can be applied to the cases where $\mathbf{MC} = \{1\}$ or $\{3\}$. First, the system states can be constructed using a similar approach. We, again, arrange the states from state 1 to state S based on bijection $\alpha_3(\cdot)$ defined in equation (7.19) for all system states (h_1, h_2, PS^{m_i}). According to the descriptive model, the dynamics of the system are still defined by equation (7.20), similar to the $\mathbf{MC} = \{2\}$ case. However, the definitions of β_i's need to be modified to account for the change of the controlled machine from m_2 to m_1 or m_3:

- If $m_{i^*} = m_1$,

$$\beta_1(n) = \begin{cases} 1, \text{ if } m_1 \text{ is } up \text{ in } run \text{ status,} \\ 0, \text{ if } m_1 \text{ is in } warm\text{-}up, cool\text{-}down, sleep \text{ status, or } down \text{ in } run \text{ status,} \end{cases}$$

$$\beta_i(n) \in \begin{cases} 1, \text{ if } m_i \text{ is } up, \\ 0, \text{ if } m_i \text{ is } down, \end{cases} \qquad i \in \{2, 3\};$$

- If $m_{i^*} = m_3$,

$$\beta_3(n) = \begin{cases} 1, \text{ if } m_3 \text{ is } up \text{ in } run \text{ status,} \\ 0, \text{ if } m_3 \text{ is in } warm\text{-}up, cool\text{-}down, sleep \text{ status, or } down \text{ in } run \text{ status,} \end{cases}$$

$$\beta_i(n) \in \begin{cases} 1, \text{ if } m_i \text{ is } up, \\ 0, \text{ if } m_i \text{ is } down, \end{cases} \qquad i \in \{1, 2\}.$$

Then, the transition probability matrices for $\mathbf{MC} = \{1\}$ and $\mathbf{MC} = \{3\}$ can be derived using the same technique described in this Subsection. Finally, let $S = S_0 \times (t_{cd} + t_{wu} + 2)$ and $S_0 = (N_1 + 1) \times (N_2 + 1)$. The resulting state evolution equation as well as the formulas for system performance evaluation are given below:

- If $m_{i^*} = m_1$,

$$\begin{aligned} &\mathbf{x}(n+1) = \mathbf{A}_3^{m_1}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \\ &x_{\alpha_3(\mu,\upsilon,\nu)}(0) = \begin{cases} 1, & \mu = h_1(0), \upsilon = h_2(0),\ \nu = PS^{m_1}(0), \\ 0, & \text{otherwise}, \end{cases} \end{aligned} \tag{7.27}$$

where $\mathbf{A}_3^{m_1}$ represents the transition probability matrix in a three-machine line with m_1's switch-on/off being controlled.

The performance measures of the line can be calculated as follows:

$$\begin{aligned} &PR(n) = \mathbf{V}_1^{3,m_1}\mathbf{x}(n), \quad CR(n) = \mathbf{V}_2^{3,m_1}\mathbf{x}(n) \quad WIP_i(n) = \mathbf{V}_{3,i}^{3,m_1}\mathbf{x}(n), \\ &BL_i(n) = \mathbf{V}_{4,i}^{3,m_1}\mathbf{x}(n) \quad ST_i(n) = \mathbf{V}_{5,i}^{3,m_1}\mathbf{x}(n), \quad POW_i(n) = \mathbf{V}_{6,i}^{3,m_1}\mathbf{x}(n), \end{aligned} \tag{7.28}$$

where

$$\begin{aligned} \mathbf{V}_1^{3,m_1} &= [0 \quad p_3\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S}, \\ \mathbf{V}_2^{3,m_1} &= [\mathbf{0}_{1,S-S_0} \quad p_1\mathbf{J}_{1,N_1(N_2+1)} \quad p_1p_2\mathbf{J}_{1,N_2} \quad p_1p_2p_3\ ], \\ \mathbf{V}_{3,i}^{3,m_1} &= [\mathbf{K}_i]\mathbf{C}_{S_0\times S}, \quad i = 1, 2, \\ \mathbf{V}_{4,1}^{3,m_1} &= [\mathbf{0}_{1,S-N_2-1} \quad p_1(1-p_2)\mathbf{J}_{1,N_2} \quad p_1p_2(1-p_3)\ ], \\ \mathbf{V}_{4,2}^{3,m_1} &= [\mathbf{0}_{1,N_2} \quad p_2(1-p_3)]\mathbf{C}_{(N_2+1)\times S}, \\ \mathbf{V}_{4,3}^{3,m_1} &= [\mathbf{0}_{1,S}], \\ \mathbf{V}_{5,1}^{3,m_1} &= [\mathbf{0}_{1,S}], \\ \mathbf{V}_{5,2}^{3,m_1} &= [p_2\mathbf{J}_{1,N_2+1} \quad \mathbf{0}_{1,N_1(N_2+1)}]\mathbf{C}_{S_0\times S}, \\ \mathbf{V}_{5,3}^{3,m_1} &= [p_3 \quad \mathbf{0}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S}, \\ \mathbf{V}_{6,1}^{3,m_1} &= [e_{cd}\mathbf{J}_{1,S_0t_{cd}} \quad e_s\mathbf{J}_{1,S_0} \quad e_{wu}\mathbf{J}_{1,S_0t_{wu}} \quad (p_1e_{op} + (1-p_1)e_{bd})\mathbf{J}_{1,(N_2+1)\times N_1} \\ &\qquad (p_1p_2e_{op} + p_1(1-p_2)e_{id} + (1-p_1)e_{bd})\mathbf{J}_{1,N_2} \\ &\qquad p_1p_2p_3e_{op} + p_1p_2(1-p_3)e_{id} + p_1(1-p_2)e_{id} + (1-p_1)e_{bd}], \end{aligned}$$

$$\begin{aligned}
\mathbf{V}_{6,2}^{3,m_1} &= [(p_2 e_{id} + (1-p_2)e_{bd})\mathbf{J}_{1,N_2+1} \quad [(p_2 e_{op} + (1-p_2)e_{bd})\mathbf{J}_{1,N_2} \\
&\qquad p_2 p_3 e_{op} + p_2(1-p_3)e_{id} + (1-p_2)e_{bd}]\mathbf{C}_{(N_2+1)\times N_1(N_2+1)}]\mathbf{C}_{S_0\times S}, \\
\mathbf{V}_{6,3}^{3,m_1} &= [(p_3 e_{id} + (1-p_3)e_{bd} \quad (p_3 e_{op} + (1-p_3)e_{bd})\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S},
\end{aligned} \tag{7.29}$$

and $\mathbf{K}_i$, $i = 1, 2$, is defined in equation (7.24).

- If $m_{i^*} = m_3$,

$$\begin{aligned}
&\mathbf{x}(n+1) = \mathbf{A}_3^{m_3}\mathbf{x}(n), \quad \sum_{i=1}^{S} x_i(n) = 1, \\
&x_{\alpha_3(\mu,\upsilon,\nu)}(0) = \begin{cases} 1, & \mu = h_1(0), \upsilon = h_2(0),\ \nu = PS^{m_3}(0), \\ 0, & \text{otherwise.} \end{cases}
\end{aligned} \tag{7.30}$$

  The performance measures can be calculated as follows:

$$\begin{aligned}
&PR(n) = \mathbf{V}_1^{3,m_3}\mathbf{x}(n), \qquad CR(n) = \mathbf{V}_2^{3,m_3}\mathbf{x}(n), \qquad WIP_i(n) = \mathbf{V}_{3,i}^{3,m_3}\mathbf{x}(n), \\
&BL_i(n) = \mathbf{V}_{4,i}^{3,m_3}\mathbf{x}(n), \qquad ST_i(n) = \mathbf{V}_{5,i}^{3,m_3}\mathbf{x}(n), \qquad POW_i(n) = \mathbf{V}_{6,i}^{3,m_3}\mathbf{x}(n),
\end{aligned}$$

where

$$\begin{aligned}
\mathbf{V}_1^{3,m_3} &= [\mathbf{0}_{1,S_0(t_{wu}+t_{cd}+1)} \quad [0 \quad p_3\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S_0}], \\
\mathbf{V}_2^{3,m_3} &= [[p_1\mathbf{J}_{1,N_1(N_2+1)} \quad p_1p_2\mathbf{J}_{1,N_2} \quad 0]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad p_1\mathbf{J}_{1,N_1(N_2+1)} \quad p_1p_2\mathbf{J}_{1,N_2} \quad p_1p_2p_3\ ], \\
\mathbf{V}_{3,i}^{3,m_3} &= [\mathbf{K}_i]\mathbf{C}_{S_0\times S}, \quad i = 1, 2, \\
\mathbf{V}_{4,1}^{3,m_3} &= [[\mathbf{0}_{1,N_1(N_2+1)} \quad p_1(1-p_2)\mathbf{J}_{1,N_2} \quad p_1]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad \mathbf{0}_{1,N_1(N_2+1)} \quad p_1(1-p_2)\mathbf{J}_{1,N_2} \\
&\qquad p_1(1-p_2) + p_1p_2(1-p_3)], \\
\mathbf{V}_{4,2}^{3,m_3} &= [[\mathbf{0}_{1,N_2} \quad p_2]\mathbf{C}_{(N_2+1)\times S_0(t_{cd}+t_{wu}+1)} \quad [\mathbf{0}_{1,N_2} \quad p_2(1-p_3)]\mathbf{C}_{(N_2+1)\times S_0}], \\
\mathbf{V}_{4,3}^{3,m_3} &= [\mathbf{0}_{1,S}], \\
\mathbf{V}_{5,1}^{3,m_3} &= [\mathbf{0}_{1,S}], \\
\mathbf{V}_{5,2}^{3,m_3} &= [p_2\mathbf{J}_{1,(N_2+1)} \quad \mathbf{0}_{1,N_1(N_2+1)}]\mathbf{C}_{S_0\times S},
\end{aligned}$$

$$
\begin{aligned}
\mathbf{V}_{5,3}^{3,m_3} &= [\mathbf{0}_{1,S_0(t_{cd}+t_{wu}+1)} \quad [p_3 \quad \mathbf{0}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S_0}], \\
\mathbf{V}_{6,1}^{3,m_3} &= [[(p_1e_{op}+(1-p_1)e_{bd})\mathbf{J}_{1,N_1(N_2+1)} \quad (p_1p_2e_{op}+p_1(1-p_2)e_{id}+(1-p_1)e_{bd})\mathbf{J}_{1,N_2} \\
&\quad p_1e_{id}]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad (p_1e_{op}+(1-p_1)e_{bd})\mathbf{J}_{1,N_1(N_2+1)} \\
&\quad (p_1p_2e_{op}+p_1(1-p_2)e_{id}+(1-p_1)e_{bd})\mathbf{J}_{1,N_2} \\
&\quad (p_1p_2p_3e_{op}+(p_1(1-p_2)+p_1p_2(1-p_3))e_{id}+(1-p_1)e_{bd}], \\
\mathbf{V}_{6,2}^{3,m_2} &= [[(p_2e_{id}+(1-p_2)e_{bd})\mathbf{J}_{1,N_2+1} \quad [(p_2e_{op}+(1-p_2)e_{bd})\mathbf{J}_{1,N_2} \\
&\quad p_2e_{id}+(1-p_2)e_{bd}]\mathbf{C}_{(N_2+1)\times N_1(N_2+1)}]\mathbf{C}_{S_0\times S_0(t_{cd}+t_{wu}+1)} \quad (p_2e_{id}+(1-p_2)e_{bd})\mathbf{J}_{1,N_2+1}) \\
&\quad [(p_2e_{op}+(1-p_2)e_{bd})\mathbf{J}_{1,N_2} \quad p_2p_3e_{op}+p_2(1-p_3)e_{id}+(1-p_2)e_{bd}]\mathbf{C}_{(N_2+1)\times N_1(N_2+1)}], \\
\mathbf{V}_{6,3}^{3,m_3} &= [e_{cd}\mathbf{J}_{1,S_0t_{cd}} \quad e_s\mathbf{J}_{1,S_0} \quad e_{wu}\mathbf{J}_{1,S_0t_{wu}} \\
&\quad [p_3e_{id}+(1-p_3)e_{bd} \quad (p_3e_{op}+(1-p_3)e_{bd})\mathbf{J}_{1,N_2}]\mathbf{C}_{(N_2+1)\times S_0}],
\end{aligned}
\tag{7.31}
$$

and $\mathbf{K}_i$, $i = 1, 2$, is defined in equation (7.24).

Finally, it should be noted that, the analysis presented in this section can be extended to systems with machines having time-varying efficiencies by replacing the p_i in the equations by $p_i(n)$, the efficiency of the machine during time slot n.

7.4 Aggregation-based Approximate Analysis for M>3-machine Lines

Although the exact Markovian analysis approach for two- and three-machine lines can be extended to M>3-machine lines, it is not practical because of the large number of system states involved. Therefore, in this section, a computationally efficient method based on recursive aggregation is developed to approximate the system performance measures.

Moreover, note that it is intuitive that the control decisions should rely more on the

buffers that are closer to the controlled machine. Therefore, we further simplify the control rule such that the switch-on/off of machine m_{i^*} only depends on the occupancy of its closest two buffers:

Control Rule 4 (simplified for $M > 3$-machine lines):

- For $m_{i^*} = m_1$, i.e., $1 \in \mathbf{MC}$,

  $\mathbf{H}_{\text{on}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \leqslant h_{1,\text{on}}^{m_1}, h_2(n) \leqslant h_{2,\text{off}}^{m_1}\}$,

  $\mathbf{H}_{\text{off}}^{m_1} = \{\mathbf{h}(n) | h_1(n) \geqslant h_{1,\text{off}}^{m_1}, h_2(n) \geqslant h_{2,\text{off}}^{m_1}\}$;

- For $m_{i^*} = m_M$, i.e., $M \in \mathbf{MC}$,

  $\mathbf{H}_{\text{on}}^{m_M} = \{\mathbf{h}(n) | h_{M-2}(n) \geqslant h_{M-2,\text{on}}^{m_M}, h_{M-1}(n) \geqslant h_{M-1,\text{off}}^{m_M}\}$,

  $\mathbf{H}_{\text{off}}^{m_M} = \{\mathbf{h}(n) | h_{M-2}(n) \leqslant h_{M-2,\text{off}}^{m_M}, h_{M-1}(n) \leqslant h_{M-1,\text{off}}^{m_M}\}$;

- For m_{i^*}, $i^* \in \mathbf{MC}$ and $2 \leqslant i^* \leqslant M-1$,

  $\mathbf{H}_{\text{on}}^{m_{i^*}} = \{\mathbf{h}(n) | h_{i^*-1}(n) \geqslant h_{i^*-1,\text{on}}^{m_{i^*}}, h_{i^*}(n) \leqslant h_{i^*,\text{on}}^{m_{i^*}}\}$,

  $\mathbf{H}_{\text{off}}^{m_{i^*}} = \{\mathbf{h}(n) | h_{i^*-1}(n) \leqslant h_{i^*-1,\text{off}}^{m_{i^*}}\} \cup \{\mathbf{h}(n) | h_{i^*}(n) \geqslant h_{i^*,\text{off}}^{m_{i^*}}\}$.

This simplification, along with condition (7.1), also allows for the *decoupling* of the switch-on/off control among the machines, in the sense that no buffer is involved in the switch-on/off decision of more than one machine.

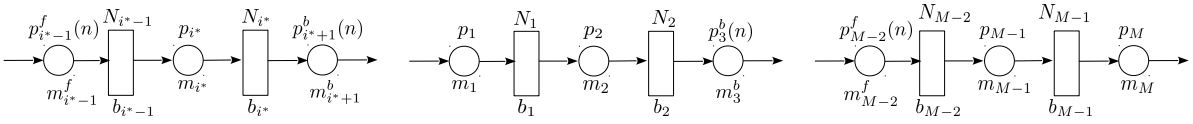
7.4.1 Idea and implementation of the aggregation procedure

To calculate the steady state performance of Bernoulli serial lines without switch-on/off control, an aggregation-based recursive calculation procedure was proposed by [118]. Based on this idea, an aggregation procedure for transient performance evaluation in such systems is developed by [147]. This procedure is improved by [101] and then extended to Bernoulli assembly systems. The issue of machine switch-on/off control, however, has not been included

in these investigations. In this dissertation, a novel aggregation procedure is proposed to contribute to this end.

Let $|\mathbf{MC}|$ denote the total number of machines being controlled. To study such a system, the original M-machine line is represented by a total of $(M-1-|\mathbf{MC}|)$ virtual lines. Specifically, for each machine with switch-on/off control, a virtual three-machine line is constructed, leading to a total of $|\mathbf{MC}|$ virtual lines. If a controlled machine m_{i^*} is an internal machine, i.e., $i^* \in \mathbf{MC}$ and $1 < i^* < M$, a virtual three-machine line is constructed with the middle machine being m_{i^*} and the two surrounding buffers being b_{i^*-1} and b_{i^*}, i.e., the immediate upstream and downstream buffers of m_{i^*} in the original line. The middle machine of the line has the same efficiency as m_{i^*}, i.e., p_{i^*}, while the first and third machines have time-varying efficiencies, denoted as $p^f_{i^*-1}(n)$ and $p^b_{i^*+1}(n)$ during time slot n, respectively (see Figure 7.4(a)). Here, superscripts f and b stand for *forward* and *backward*, and $p^f_{i^*-1}(n)$ and $p^b_{i^*+1}(n)$ represent the *aggregated* effects of part producing and consuming in the original line from up- and downstream of buffers b_{i^*-1} and b_{i^*}, respectively. If either m_1 or m_M is being controlled, then the virtual three-machine line is constructed as illustrated in Figure 7.4(b) and 7.4(c): if $m_{i^*} = m_1$, the efficiency of the first machine in the virtual line is simply p_1, since m_1 is the only component upstream of b_1. Similarly, if m_{i^*}=m_M, the efficiency of the third machine in the virtual line is just p_M. It should be noted that, due to the assuption (viii), no buffer is shared by any two virtual three-machine lines constructed above. Thus, the number of buffers not used by the virtual lines becomes $M-1-2|\mathbf{MC}|$. Next, we construct a virtual two-machine line around each of the unused buffers. For each of these two-machine lines, the up- and downstream machines have time-varying efficiencies $p^f_i(n)$ and $p^b_{i+1}(n)$ representing the *aggregated* effects from up- and downstream of buffer b_i in the original line, respectively (see Figure 7.5).

Clearly, if the parameters of the virtual machines are known (i.e., the ones with superscripts f and b), it is possible to apply the approach described in Subsection 7.3.2 and the method developed by [147] to analyze these virtual lines. Therefore, to identify the pa-



(a) $m_{i^*}, 1 < i^* < M$ (b) $m_{i^*} = m_1$ (c) $m_{i^*} = m_M$

Figure 7.4: Virtual three-machine lines

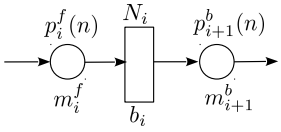


Figure 7.5: Virtual two-machine line

rameters of the virtual machines and to approximate the system evolution and transient performance using the virtual lines, the following calculation procedure is proposed:

Step 0: Identify the virtual three-machine and two-machine lines based on the descriptions above.

Step 1: Define two sets of indices, $\mathbf{VL}_{2M}$ and $\mathbf{VL}_{3M}$, where $\mathbf{VL}_{2M}$ contains indices i for all buffers b_i in the virtual two-machine lines, and $\mathbf{VL}_{3M}$ contains index pairs (j_1, j_2) for all buffer pairs (b_{j_1}, b_{j_2}) in the virtual three-machine lines. Apparently, $j_1 + 1 = j_2$. Since the virtual two-machine lines do not contain the machines with switch-on/off control, the state of each two-machine line shown in Figure 7.5 only includes the occupancy of the buffer. Therefore, for $i \in \mathbf{VL}_{2M}$, let $\widehat{\mathbf{x}}^{(i)}(n) = [\widehat{x}^{(i)}_0(n)\ \widehat{x}^{(i)}_1(n)\ ...\ \widehat{x}^{(i)}_{N_i}(n)]^T$ denote the probability distribution of the buffer occupancy in the virtual two-machine line involving buffer b_i at the end of time slot n. In addition, for the three-machine lines involving buffers b_{j_1} and b_{j_2}, i.e., $(j_1, j_2) \in \mathbf{VL}_{3M}$, let $\widehat{\mathbf{x}}^{(j_1,j_2)}(n) = [\widehat{x}^{(j_1,j_2)}_0(n)\ \widehat{x}^{(j_1,j_2)}_1(n)\ ...\ \widehat{x}^{(j_1,j_2)}_S(n)]^T$ denote the probability distribution of the states of the virtual line constructed based on the methods given in Subsection 7.3.2. The boundary condition of the procedure is $p^b_M(n) = p_M$ and $p^f_1(n) = p_1$. The initial condition is:

$$\widehat{x}_j^{(i)}(0) = \begin{cases} 1, & j = h_i(0), \\ 0, & \text{otherwise}, \end{cases} \qquad i \in \mathbf{VL}_{2M},$$

$$\widehat{x}_{\alpha_3(\mu,\upsilon,\nu)}^{(j_1,j_2)}(0) = \begin{cases} 1, \ \mu = h_{j_1}(0), \upsilon = h_{j_2}(0), \ \nu = PS^{m_l}(0), \\ 0, \ \text{otherwise}, \end{cases} \qquad (j_1, j_2) \in \mathbf{VL}_{3M},$$

where m_l indicates the controlled machine in the virtual three-machine line with buffers b_{j_1} and b_{j_2}. Let $n = 0$.

Step 2: For each $i \in \mathbf{VL}_{2M}$, and $(j_1, j_2) \in \mathbf{VL}_{3M}$, calculate $p_{i+1}^f(n+1)$ and $p_{j_2+1}^f(n+1)$ as follows:

$$p_{i+1}^f(n+1) = p_{i+1}[1 - \widehat{x}_0^{(i)}(n)], \qquad i \in \mathbf{VL}_{2M},$$

$$p_{j_2+1}^f(n+1) = p_{j_2+1} \cdot \frac{\mathbf{V}_1^{3,m_k}(p_{j_1}^f(n+1), p_{j_2}, p_{j_2+1}^b(n+1))\widehat{\mathbf{x}}^{(j_1,j_2)}(n)}{p_{j_2+1}^b(n+1)}, \quad (j_1, j_2) \in \mathbf{VL}_{3M},$$

$$k = \sum_{j=j_1}^{j_1+2} j \cdot \mathbf{1}_{\mathbf{MC}}(j) - (j_1 - 1), \quad \mathbf{1}_{\mathbf{MC}}(j) := \begin{cases} 1 & \text{if } j \in \mathbf{MC}, \\ 0 & \text{if } j \notin \mathbf{MC}. \end{cases}$$

where $k \in \{1, 2, 3\}$ indicates the position of the controlled machine in the virtual three-machine line and $\mathbf{V}_1^{3,m_k}$ is given in Subsection 7.3.2 with p_1, p_2, p_3 replaced by $p_{j_1}^f(n+1), p_{j_2}, p_{j_2+1}^b(n+1)$ and N_1, N_2 replaced by N_{j_1}, N_{j_2}, respectively.

Step 3: For each $i \in \mathbf{VL}_{2M}$, and $(j_1, j_2) \in \mathbf{VL}_{3M}$, calculate $p_i^b(n+1)$ and $p_{j_1}^b(n+1)$ in the descending order of i and j_1 as follows:

$$p_i^b(n+1) = p_i[1 - (1 - p_{i+1}^b(n))\widehat{x}_{N_i}^{(i)}(n)], \quad i \in \mathbf{VL}_{2M},$$

$$p_{j_1}^b(n+1) = p_{j_1} \cdot \frac{\mathbf{V}_2^{3,m_k}(p_{j_1}^f(n+1), p_{j_2}, p_{j_2+1}^b(n+1))\widehat{\mathbf{x}}^{(j_1,j_2)}(n)}{p_{j_1}^f(n+1)}, \quad (j_1, j_2) \in \mathbf{VL}_{3M},$$

$$k = \sum_{j=j_1}^{j_1+2} j \cdot \mathbf{1}_{\mathbf{MC}}(j) - (j_1 - 1),$$

where $\mathbf{V}_2^{3,m_k}$ is also given in Subsection 7.3.2 with p_1, p_2, p_3 replaced by $p_{j_1}^f(n+1), p_{j_2}, p_{j_2+1}^b(n+$

1) and N_1, N_2 replaced by N_{j_1}, N_{j_2}, respectively.

Step 4: Update all $\widehat{\mathbf{x}}^{(i)}(n+1)$'s and $\widehat{\mathbf{x}}^{(j_1,j_2)}(n+1)$'s based on

$$\begin{aligned}
\widehat{\mathbf{x}}^{(i)}(n+1) &= \mathbf{A}_2(p_i^f(n+1), p_{i+1}^b(n+1), N_i)\widehat{\mathbf{x}}^{(i)}(n), \quad i \in \mathbf{VL}_{2M}, \\
\widehat{\mathbf{x}}^{(j_1,j_2)}(n+1) &= \mathbf{A}_3^{m_k}(p_{j_1}^f(n+1), p_{j_2}, p_{j_2+1}^b(n+1))\widehat{\mathbf{x}}^{(j_1,j_2)}(n), \quad (j_1,j_2) \in \mathbf{VL}_{3M}, \\
k &= \sum_{j=j_1}^{j_1+2} j \cdot \mathbf{1}_{\mathbf{MC}}(j) - (j_1 - 1),
\end{aligned}$$

where $\mathbf{A}_2(p_i^f(n+1), p_{i+1}^b(n+1), N_i)$ represents the one-step transition probability matrix of a two-machine Bernoulli line without switch-on/off control [147] and $\mathbf{A}_3^{m_k}$ is the transition probability matrix of the three-machine line calculated based on the methods described in Subsection 7.3.2.

Step 5: The performance measures of the original system for time $n+1$ are approximated based on the virtual two- and three-machine lines constructed. Specifically,

$$\begin{aligned}
\widehat{PR}(n+1) &= p_M^f(n+1), \qquad \widehat{CR}(n+1) = p_1^b(n+1), \\
\widehat{BL}_i(n+1) &= \begin{cases}
p_i \widehat{\mathbf{x}}_{N_i}^i(n)(1 - p_{i+1}^b(n+1)), & i \in \mathbf{VL}_{2M}, \\
p_i \cdot \dfrac{\mathbf{V}_{4,1}^{3,m_k}(p_i^f(n+1), p_{i+1}, p_{i+2}^b(n+1))\widehat{\mathbf{x}}^{(i,i+1)}(n)}{p_i^f(n+1)}, & (i,i+1) \in \mathbf{VL}_{3M}, \\
\mathbf{V}_{4,2}^{3,m_k}(p_{i-1}^f(n+1), p_i, p_{i+1}^b(n+1))\widehat{\mathbf{x}}^{(i-1,i)}(n), & (i-1,i) \in \mathbf{VL}_{3M},
\end{cases} \\
\widehat{ST}_i(n+1) &= \begin{cases}
p_i \widehat{\mathbf{x}}_0^{i-1}(n), & i-1 \in \mathbf{VL}_{2M}, \\
\mathbf{V}_{5,2}^{3,m_k}(p_{i-1}^f(n+1), p_i, p_{i+1}^b(n+1))\widehat{\mathbf{x}}^{(i-1,i)}(n), & (i-1,i) \in \mathbf{VL}_{3M}, \\
p_i \cdot \dfrac{\mathbf{V}_{5,3}^{3,m_k}(p_{i-2}^f(n+1), p_{i-1}, p_i^b(n+1))\widehat{\mathbf{x}}^{(i-2,i-1)}(n)}{p_i^b(n+1)}, & \\
\qquad\qquad (i-2,i-1) \in \mathbf{VL}_{3M},
\end{cases}
\end{aligned}$$

$$
\widehat{WIP}_i(n+1) = \begin{cases} \displaystyle\sum_{k=0}^{N_i} k\widehat{\mathbf{x}}_k^i(n), \quad i \in \mathbf{VL}_{2M}, \\ \mathbf{V}_{3,1}^{3,m_k}(p_i^f(n+1), p_{i+1}, p_{i+2}^b(n+1))\widehat{\mathbf{x}}^{(i,i+1)}(n), \quad (i,i+1) \in \mathbf{VL}_{3M}, \\ \mathbf{V}_{3,2}^{3,m_k}(p_{i-1}^f(n+1), p_i, p_{i+1}^b(n+1))\widehat{\mathbf{x}}^{(i-1,i)}(n), \quad (i-1,i) \in \mathbf{VL}_{3M}, \end{cases}
$$

$$
\widehat{POW}_i(n+1) = \begin{cases} e_{op}p_i(1-\widehat{BL_i}(n+1)-\widehat{ST_i}(n+1)) + e_{bd}(1-p_i) \\ \qquad + e_{id}p_i(\widehat{BL_i}(n+1)+\widehat{ST_i}(n+1)), \quad \text{for } i \notin \mathbf{MC}, \\ \dfrac{p_i\mathbf{V}_{6,1}^{3,m_1}(p_i^f(n+1), p_{i+1}, p_{i+2}^b(n+1))\widehat{\mathbf{x}}^{(i,i+1)}(n)}{p_i^f(n+1)(1-\widehat{ST_i}(n+1))} \\ \quad + (e_{id}p_i + e_{bd}(1-p_i))\widehat{ST_i}(n+1), \\ \qquad \text{for } i \in \mathbf{MC} \text{ and } (i,i+1) \in \mathbf{VL}_{3M}, \\ \mathbf{V}_{6,2}^{3,m_2}(p_{i-1}^f(n+1), p_i, p_{i+1}^b(n+1))\widehat{\mathbf{x}}^{(i-1,i)}(n), \\ \qquad \text{for } i \in \mathbf{MC} \text{ and } (i-1,i) \in \mathbf{VL}_{3M}, \\ \dfrac{p_i\mathbf{V}_{6,3}^{3,m_3}(p_{i-2}^f(n+1), p_{i-1}, p_i^b(n+1))\widehat{\mathbf{x}}^{(i-2,i-1)}(n)}{p_i^b(n+1)(1-\widehat{BL_i}(n+1))} \\ \quad + (e_{id}p_i + e_{bd}(1-p_i))\widehat{ST_i}(n+1), \\ \qquad \text{for } i \in \mathbf{MC} \text{ and } (i-2,i-1) \in \mathbf{VL}_{3M}. \end{cases}
$$

Then, let $n = n + 1$ and return to Step 2.

7.4.2 Accuracy of the approximation method

To investigate the accuracy of the performance approximation method developed above, numerical experiments were carried out. Specifically, we studied a total of $150,000$ Bernoulli lines with the parameters randomly and equiprobably (i.e., uniformly) selected from the

following sets:

$$M \in \{4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}, \quad N_i \in \{4,\ 5,\ 6,\ 7,\ 8,\ 9\}, \quad p_i \in (0.7,\ 1),$$
$$t_{wu} \in \{1,\ 2,\ 3\}, \quad t_{cd} \in \{1,\ 2,\ 3\}. \tag{7.32}$$

For each line thus constructed, we first calculated the maximum number of controlled machines allowed by condition (7.1):

$$\Theta(M) = \left\lfloor \frac{M-1}{2} \right\rfloor. \tag{7.33}$$

Then, we randomly and equiprobably selected the actual number of controlled machines $|\mathbf{MC}|$ from $\{1, 2, \ldots, \Theta(M)\}$. The positions of the controlled machines were selected randomly and equiprobably from all feasible cases under $|\mathbf{MC}|$ and condition (7.1). The parameters of the control policies are also randomly and equiprobably generated based on the buffer capacities.

The accuracy of the performance measure approximations are evaluated based on the average approximation errors of each line studied:

$$\delta_{PR}(n) = \frac{|\widehat{PR}(n) - PR_{sim}(n)|}{PR_{ss}} \cdot 100\%, \qquad \delta_{CR}(n) = \frac{|\widehat{CR}(n) - CR_{sim}(n)|}{CR_{ss}} \cdot 100\%,$$
$$\delta_{WIP}(n) = \frac{\sum_{i=1}^{M-1} \frac{|\widehat{WIP_i}(n) - WIP_i^{sim}(n)|}{N_i}}{M-1} \cdot 100\%, \quad \delta_{BL}(n) = \frac{\sum_{i=1}^{M-1} |\widehat{BL_i}(n) - BL_i^{sim}(n)|}{M-1},$$
$$\delta_{POW}(n) = \frac{\sum_{i=1}^{M} |\widehat{POW_i}(n) - POW_i^{sim}(n)|}{\sum_{i=1}^{M} POW_i^{ss}} \cdot 100\%, \quad \delta_{ST}(n) = \frac{\sum_{i=2}^{M} |\widehat{ST_i}(n) - ST_i^{sim}(n)|}{M-1},$$

where the notations with "$\widehat{\ }$" represent the approximations calculated based on the proposed procedure, the notations with superscript *sim* represent the simulation result, and the notations with superscript or subscript *ss* represent the performance measures' steady state values.

The results are summarized in Figure 7.6. Note that the solid lines represent the median of the errors among 150,000 different lines and the dotted lines are the first and third quartiles.

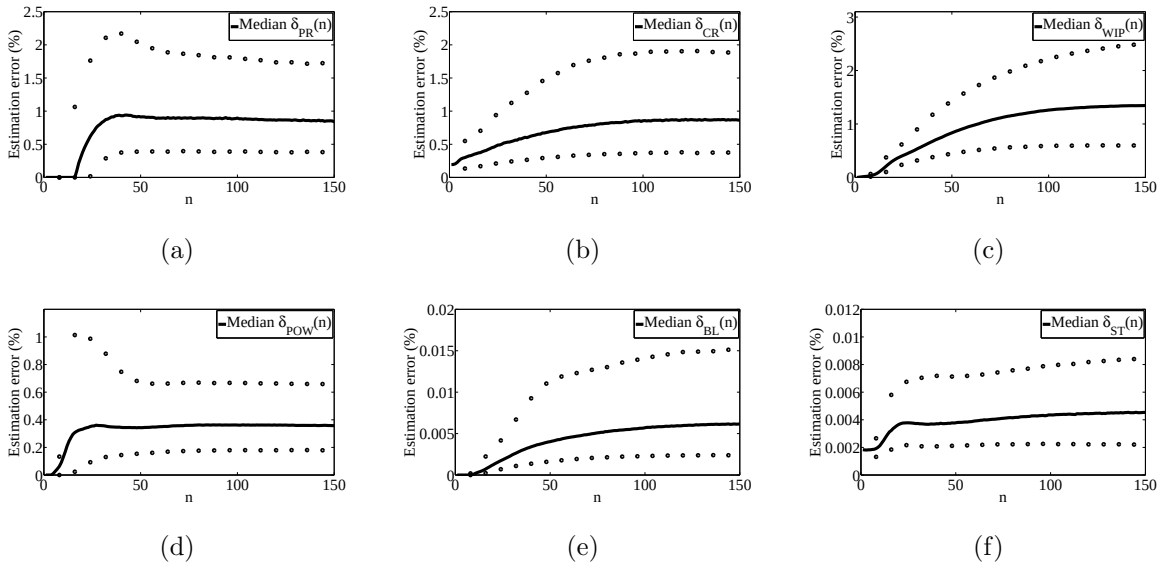


Figure 7.6: Estimation errors of the performance approximations

It shows that, the approximation error is typically within 2% for $PR(n), CR(n), WIP(n)$ and $POW(n)$. In addition, the average approximation error is also very small for $BL(n)$ and $ST(n)$. Taking into account that the parameters of the machines and buffers are rarely known on the factory floor with accuracy better than $5\% - 10\%$, we conclude that the proposed procedure can be used as an effective tool to estimate the transient performance of Bernoulli systems with good accuracy.

As an illustration, consider a ten-machine line (see Figure 7.7) with parameters given as

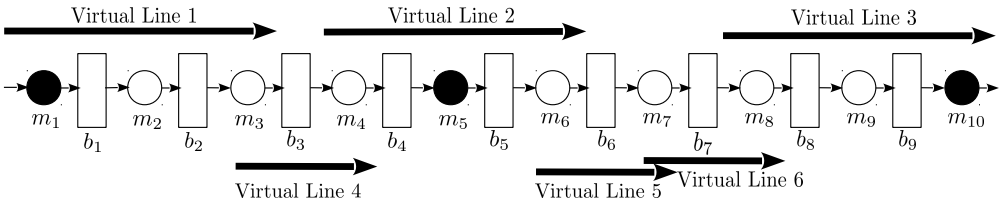


Figure 7.7: Virtual lines construction based on the controlled machines

follows:

$$M = 10,\ \mathbf{MC} = \{1, 5, 10\},\ t_{wu} = 1,\ t_{cd} = 1,\ \mathbf{N} = [4 \quad 4 \quad 5 \quad 6 \quad 5 \quad 6 \quad 4 \quad 6 \quad 4],$$

$$\mathbf{p} = [0.82 \quad 0.79 \quad 0.75 \quad 0.84 \quad 0.87 \quad 0.92 \quad 0.86 \quad 0.95 \quad 0.88 \quad 0.91],$$

$$\mathbf{H}_{\text{on}}^{m_1} = \{\mathbf{h}|h_1 \leqslant 1, h_2 \leqslant 1\}, \quad \mathbf{H}_{\text{off}}^{m_1} = \{\mathbf{h}|h_1 \geqslant 3, h_2 \geqslant 3\},$$

$$\mathbf{H}_{\text{on}}^{m_5} = \{\mathbf{h}|h_4 \geqslant 5, h_5 \leqslant 1\}, \quad \mathbf{H}_{\text{off}}^{m_5} = \{\mathbf{h}|h_4 \leqslant 1\} \cup \{\mathbf{h}|h_5 \geqslant 5\},$$

$$\mathbf{H}_{\text{on}}^{m_{10}} = \{\mathbf{h}|h_8 \geqslant 4, h_9 \geqslant 3\}, \quad \mathbf{H}_{\text{off}}^{m_{10}} = \{\mathbf{h}|h_8 \leqslant 1, h_9 \leqslant 1\}.$$

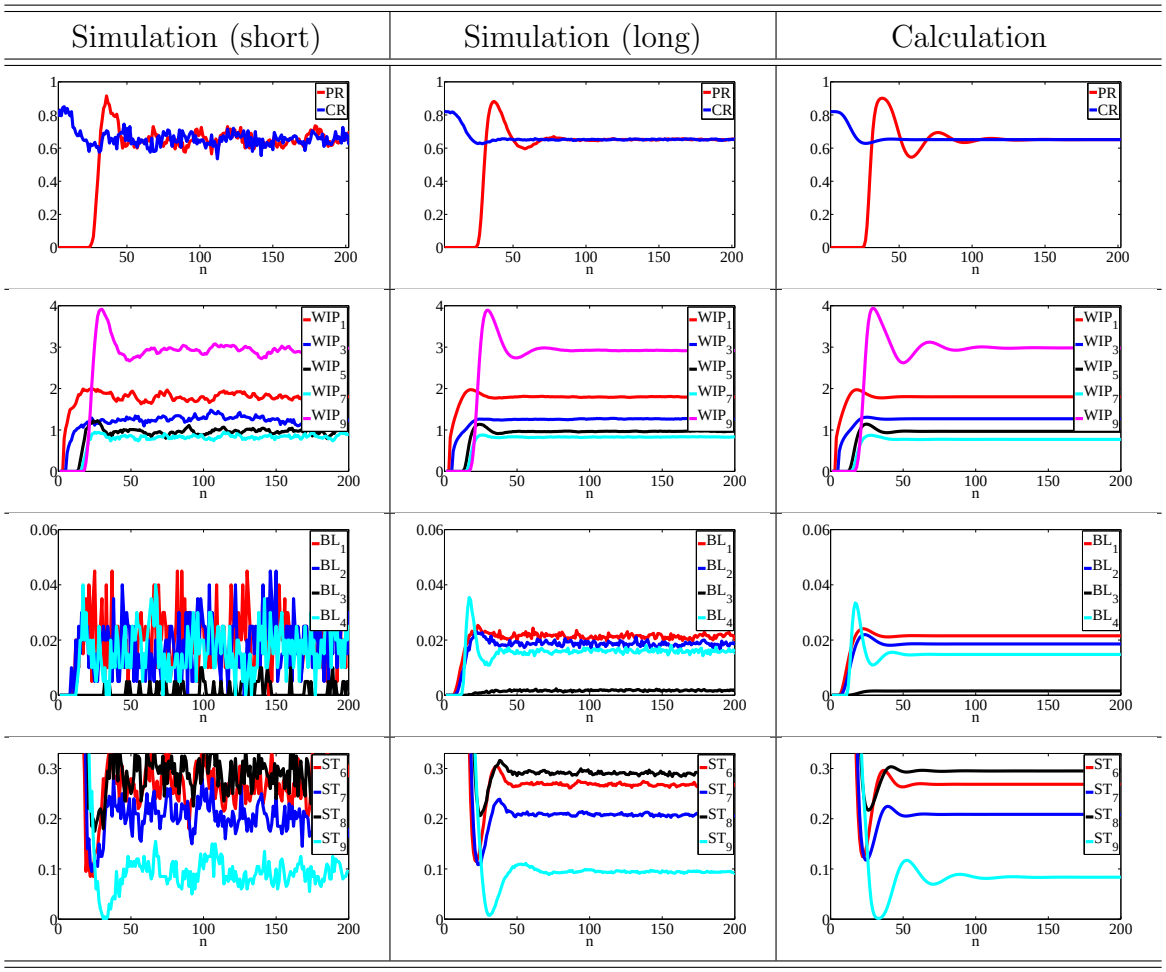


Figure 7.8: Comparison of simulation- and calculation-based methods for transient performance evaluation

For this system, Virtual (three-machine) Lines 1, 2 and 3 are constructed around the three controlled machines m_1, m_5 and m_{10} (indicated by the black circles in the figure). Then, for the rest $M - 1 - 2|\mathbf{MC}| = 10 - 1 - 2 \times 3 = 3$ unused buffers, Virtual (two-machine) Lines 4, 5 and 6 are constructed (see Figure 7.7). Clearly, in this example, $\mathbf{VL}_{2M} = \{3,\ 6,\ 7\}$, $\mathbf{VL}_{3M} = \{(1,2),\ (4,5),\ (8,9)\}$. The performance measures of the system obtained using simulation and the proposed calculation procedure are given in Figure 7.8. Specifically,

the rightmost column in Figure 7.8 are the results obtained by the calculation procedure, while the leftmost column provides the results using simulation with approximately the same computing time (around ten seconds on a PC with 3.4 GHz Intel Core i7 and 8 GB memory). The middle column in the figure gives the simulation results with 100 times the computing time. As one can see, the calculation procedure developed in this Chapter is capable of approximating the system performance measures with high precision and accuracy and relatively small computing effort.

7.5 Summary

In this Chapter, we study serial production lines with Bernoulli machines and finite buffers and assume that some of the machines in the line can be switched on and off during the production process according to a threshold-based feedback control policy. Mathematical models for the system under consideration are derived and analytical methods are developed for calculating the system performance measures of production rate, consumption rate, work-in-process, machine blockage, machine starvation and power, etc., during transients and steady state. Specifically, in two- and three-machine lines, we assume that only one machine is controlled and exact Markovian analysis is used. For longer lines, switch-on/off control of multiple machines is considered. Unfortunately, the exact Markovian analysis approach for two- and three-machine lines cannot be extended to $M>3$-machine lines due to the large number of the system states. Instead, an aggregation-based approximation approach is developed for evaluating the system performance. To investigate the accuracy of this performance approximation method, numerical experiments were carried out and the results show that the proposed method can be used to efficiently calculate the system's performance with high accuracy.

Chapter 8

CONCLUSIONS AND FUTURE WORK

8.1 Conclusions

In this dissertation, we study the problems of performance evaluation, bottleneck analysis, and production control of serial lines, closed lines and assembly systems. Specifically, the following results have been reported:

- The problems of performance evaluation, system-theoretic properties, in the framework of serial production lines with Bernoulli/geometric machines and finite buffers is discussed; Using Markovian analysis, closed-form expressions are provided to calculate the performance measures for one- and two-machine lines; For longer lines, a computationally efficient algorithm is developed to approximate the system performance measures with high accuracy; Properties of system are investigated and a case study is carried out to illustrate the applicability of the methods developed.

- The transient performance of closed serial lines with machines having the Bernoulli reliability model are studied. Specifically, exact mathematical model for the system considered is derived based on Markov analysis. Then, formulas for calculating the system's performance measures during transients are obtained based on the model. An approximation method is proposed to estimate a closed Bernoulli line's performance

with a finite production run.

- The transient performance evaluation problems of assembly systems with Bernoulli machines and finite buffers is studied. A computationally-efficient algorithm is developed to approximate the transient performance measures of Bernoulli assembly system by transforming the system into a pair of interacting serial lines. Extend the algorithm to complex assembly systems with multiple component lines and assembly operations.

- Serial production lines with Bernoulli machines and finite buffers and assume that some of the machines in the line can be switched on and o during the production process according to a threshold-based feedback control policy are studied. Mathematical models for the system under consideration are derived and analytical methods are developed for calculating the system performance measures during transients and steady state. Specifically, in two- and three-machine lines, we assume that only one machine is controlled and exact Markovian analysis is used. For longer lines, switch-on/off control of multiple machines is considered. An aggregation-based approximation approach is developed for evaluating the system performance.

8.2 Future Work

Inspired by the preliminary results obtained, the planned work in the future of this research includes:

- Extension of the results to systems with exponential machines. Since production systems with exponential machines are also Markovian, the aggregation-based approach is still applicable;

- Extension of the results to systems with non-Markovian (Weibull, gamma, log-normal, etc.) and general models of machine reliability;

- For the preliminary work that has been done in the machine operations control, in addition to the limit of Bernoulli reliability model, more general control rules as well as optimal control parameters/policies and adaptive control will be investigated;
- Investigations of bottleneck identification and lean design for production systems with machine switch-on/off control;
- Generalization of the operation control results to production systems with other structures (e.g., assembly, rework, closed lines).

References

[1] N. Viswanadham and Y. Narahari, *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall, Englewood Cliff, NJ, 1992.

[2] J. A. Buzacott and J. G. Shanthikumar, *Stochastic Models of Manufacturing Systems*. Prentice Hall, Englewood Cliff, NJ, 1993.

[3] H. T. Papadopoulos, C. Heavy, and J. Browne, *Queueing Theory in Manufacturing Systems Analysis and Design*. Chapman & Hill, London, UK, 1993.

[4] R. G. Askin and C. R. Standridge, *Modeling and Analysis of Manufacturing Systems*. Wiley, 1993.

[5] S. B. Gershwin, *Manufacturing Systems Engineering*. Prentice Hall, Englewood Cliff, NJ, 1994.

[6] H. G. Perros, *Queueing Networks with Blocking*. Oxford University Press, Oxford, 1994.

[7] T. Altiok, *Performance Analysis of Manufacturing Systems*. Springer-Verlag, New York, NY, 1997.

[8] G. L. Curry and R. M. Feldman, *Manufacturing Systems Modeling and Analysis*. Springer, 2009.

[9] J. Li and S. M. Meerkov, *Production Systems Engineering*. Springer, 2009.

[10] J. Li, D. E. Blumenfeld, N. Huang, and J. M. Alden, "Throughput analysis of production systems: Recent advances and future topics," *International Journal of Production Research*, vol. 47, no. 14, pp. 3823–3851, 2009.

[11] W. K. Grassmann, "Transient solutions in Markovian queueing systems," *Computers & Operations Research*, vol. 4, no. 1, pp. 47–53, 1977.

[12] M. F. Neuts, *Structured Stochastic Matrices of $M/G/1$ Type and Their Applications*. CRC Press, 1989, vol. 5.

[13] D. J. Bertsimas and D. Nakazato, "Transient and busy period analysis of the $G/G/1$ queue: The method of stages," *Queueing Systems*, vol. 10, no. 3, pp. 153–184, 1992.

[14] Y. Narahari and N. Viswanadham, "Transient analysis of manufacturing systems performance," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 230–244, 1994.

[15] J.-M. Garcia, O. Brun, and D. Gauchard, "Transient analytical solution of $M/D/1/N$ queues," *Journal of Applied Probability*, pp. 853–864, 2002.

[16] R. Stolletz and S. Lagershausen, "Time-dependent performance evaluation for loss-waiting queues with arbitrary distributions," *International Journal of Production Research*, vol. 51, no. 5, pp. 1366–1378, 2013.

[17] G.-A. Klutke and L. M. Seiford, "Transient behaviour of finite capacity tandem queues with blocking," *International Journal of Systems Science*, vol. 22, no. 11, pp. 2205–2215, 1991.

[18] M. N. Gopalan and U. D. Kumar, "On the transient behaviour of a merge production system with an end buffer," *International Journal of Production Economics*, vol. 34, no. 2, pp. 157–165, 1994.

[19] S. Mocanu, "Numerical algorithms for transient analysis of fluid queues," in *Proceedings of 5th International Conference on the Analysis of Manufacturing Systems*, Zakymthos, Greece, 2005, pp. 15–20.

[20] L. Lin and J. K. Cochran, "Metamodels of production line transient behaviour for sudden machine breakdowns," *International Journal of Production Research*, vol. 28, no. 10, pp. 1791–1806, 1990.

[21] E. J. Stahlman and J. K. Cochran, "Dynamic metamodelling in capacity planning," *International Journal of Production Research*, vol. 36, no. 1, pp. 197–210, 1998.

[22] H. Missbauer, "Models of the transient behaviour of production units to optimize the aggregate material flow," *International Journal of Production Economics*, vol. 118, no. 2, pp. 387–397, 2009.

[23] S. Shaaban and S. Hudson, "Transient behaviour of unbalanced lines," *Flexible Services and Manufacturing Journal*, vol. 24, no. 4, pp. 575–602, 2012.

[24] F. Yang and J. Liu, "Simulation-based transfer function modeling for transient analysis of general queueing systems," *European Journal of Operational Research*, vol. 223, no. 1, pp. 150–166, 2012.

[25] G. Chen, L. Zhang, J. Arinez, and S. Biller, "Energy-efficient production systems through schedule-based operations," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 27–37, 2013.

[26] S. B. Gershwin, "Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks," *IIE Transactions*, vol. 23, no. 4, pp. 302–314, 1990.

[27] X.-G. Liu and J. A. Buzacott, "Approximate models of assembly systems with finite inventory banks," *European Journal of Operational Research*, vol. 45, no. 2-3, pp. 143–154, 1990.

[28] S. Helber, "Decomposition of unreliable assembly/disassembly networks with limited buffer capacity and random processing times," *European Journal of Operational Research*, vol. 109, no. 1, pp. 24–42, 1998.

[29] M. D. Mascolo, R. David, and Y. Dallery, "Modeling and analysis of assembly systems with unreliable machines and finite buffers," *IIE Transactions*, vol. 23, no. 4, pp. 315–330, 1991.

[30] S. B. Gershwin and M. H. Burman, "A decomposition method for analyzing inhomogeneous assembly/disassembly systems," *Annals of Operations Research*, vol. 93, no. 1-4, pp. 91–115, 2000.

[31] S. Ching, S. M. Meerkov, and L. Zhang, "Assembly systems with non-exponential machines: throughput and bottlenecks," *Nonlinear Analysis*, vol. 69, no. 3, pp. 911–917, 2008.

[32] J.-T. Lim and S. M. Meerkov, "On asymptotically reliable closed serial production lines," *Control Engineering Practice*, vol. 1, no. 1, pp. 147–152, 1993.

[33] Y. Frein, C. Commault, and Y. Dallery, "Odeling and analysis of closed-loop production lines with unreliable machines and finite buffers," *IIE Transactions*, vol. 28, no. 7, pp. 545–554, 1996.

[34] S. B. Gershwin, N. Maggio, A. Matta, T. Tolio, and L. Werner, "Analysis of loop networks by decomposition," in *Third Aegean International Conference on Analysis and Modeling of Manufacturing Systems*, 2001, pp. 239–248.

[35] O. Rose, "Wip evolution of a semiconductor factory after a bottleneck workcenter breakdown," in *Proceedings of the 30th Conference on Winter Simulation*. IEEE Computer Society Press, 1998, pp. 997–1004.

[36] ——, "Estimation of the cycle time distribution of a wafer fab by a simple simulation model," *in Proceedings of the SMOMS*, vol. 99, no. 1999, p. 118, 1999.

[37] ——, "Improving the accuracy of simple simulation models for complex production systems," in *Proceedings of the 2007 INFORMS Simulation Society Research Workshop*, 2007, pp. 5–7.

[38] F. Yang and J. Liu, "Simulation-based transfer function modeling for transient analysis of general queueing systems," *European Journal of Operational Research*, vol. 223, no. 1, pp. 150–166, 2012.

[39] S. M. Meerkov and L. Zhang, "Transient behavior of serial production lines with Bernoulli machines," *IIE Transactions*, vol. 40, no. 3, pp. 297–312, 2008.

[40] ——, "Transients in production lines with two non-identical Bernoulli machines," in *Proceedings of 7th International Conference on the Analysis of Manufacturing Systems*, 2009.

[41] ——, "Unbalanced production systems with floats: Analysis and lean design," *International Journal of Manufacturing Technology & Management*, vol. 23, no. 1-2, pp. 4–15, 2011.

[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, no. 5, pp. 528–543, 2013.

[43] G. Chen, C. Wang, L. Zhang, J. Arinez, and G. Xiao, "Transient performance analysis of serial production lines with geometric machines," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 877–891, 2016.

[44] J. Wang, Y. Hu, and J. Li, "Transient analysis to design buffer capacity in dairy filling and packing production lines," *Journal of Food Engineering*, vol. 98, pp. 1–12, 2010.

[45] D. R. Alexander, I. Premachandra, and T. Kimura, "Transient and asymptotic behavior of synchronization processes in assembly-like queues," *Annals of Operations Research*, vol. 181, no. 1, pp. 641–659, 2010.

[46] W. D. Kelton and A. M. Law, *Simulation Modeling and Analysis*. McGraw Hill Boston, 2000.

[47] A. Berger, L. Bregman, and Y. Kogan, "Bottleneck analysis in multiclass closed queueing networks and its application," *Queueing Systems*, vol. 31, no. 3-4, pp. 217–237, 1999.

[48] J. F. Cox III and M. S. Spencer, *The Constraints Management Handbook*. CRC Press, 1997.

[49] C. Roser, M. Nakano, and M. Tanaka, "A practical bottleneck detection method," in *Proceedings of the 33nd conference on Winter simulation*. IEEE Computer Society, 2001, pp. 949–953.

[50] Z. Kralova and M. Bielak, "Production process synchronization using simulation in witness," in *Proceedings of 7th Conference WITNESS, Kozov, Czech Republic*, 2004, pp. 78–84.

[51] C.-T. Kuo, J.-T. Lim, and S. M. Meerkov, "Bottlenecks in serial production lines: A system-theoretic approach," *Mathematical problems in engineering*, vol. 2, no. 3, pp. 233–276, 1996.

[52] 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, no. 6, pp. 543–578, 2001.

[53] S. Ching, S. M. Meerkov, and L. Zhang, "Assembly systems with non-exponential machines: Throughput and bottlenecks," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 69, no. 3, pp. 911–917, 2008.

[54] S. Biller, J. Li, S. P. Marin, S. M. Meerkov, and L. Zhang, "Bottlenecks in Bernoulli serial lines with rework," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 2, pp. 208–217, 2010.

[55] G. Olsder and R. Sum, "Time-optimal con-trol of flexible manufacturing systems with failure prone machines," in *Proceedings of the 19th IEEE Conference Decision and Control*, 1980.

[56] J. Kimemia and S. B. Gershwin, "An algorithm for the computer control of a flexible manufacturing system," *AIIE Transactions*, vol. 15, no. 4, pp. 353–362, 1983.

[57] T. Bielecki and P. Kumar, "Optimality of zero-inventory policies for unreliable manufacturing systems," *Operations research*, vol. 36, no. 4, pp. 532–541, 1988.

[58] J.-Q. Hu, "Production rate control for failure-prone production systems with no backlog permitted," *IEEE Transactions on Automatic Control*, vol. 40, no. 2, pp. 291–295, 1995.

[59] F. Martinelli and P. Valigi, "Hedging point policies remain optimal under limited backlog and inventory space," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1863–1871, 2004.

[60] J. R. Perkins and R. Srikant, "Failure-prone production systems with uncertain demand," *IEEE Transactions on Automatic Control*, vol. 46, no. 3, pp. 441–449, 2001.

[61] B. Tan, "Production control of a pull system with production and demand uncertainty," *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 779–783, 2002.

[62] A. M. Bonvik, C. Couch, and S. B. Gershwin, "A comparison of production-line control mechanisms," *International Journal of Production Research*, vol. 35, no. 3, pp. 789–804, 1997.

[63] S. B. Gershwin, "Hierarchical flow control: A framework for scheduling and planning discrete events in manufacturing systems," *in Proceedings of the IEEE*, vol. 77, no. 1, pp. 195–209, 1989.

[64] O. Sauer, "Trends in manufacturing execution systems," in *Proceedings of the 6th CIRP-Sponsored International Conference on Digital Enterprise Technology*. Springer, 2010, pp. 685–693.

[65] V. V. Prabhu, H. W. Jeon, and M. Taisch, "Modeling green factory physicsan analytical approach," in *2012 IEEE International Conference on Automation Science and Engineering*. IEEE, 2012, pp. 46–51.

[66] G. Chen, L. Zhang, J. Arinez, and S. Biller, "Energy-efficient production systems through schedule-based operations," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 27–37, 2013.

[67] G. Mouzon, M. B. Yildirim, and J. Twomey, "Operational methods for minimization of energy consumption of manufacturing equipment," *International Journal of Production Research*, vol. 45, no. 18-19, pp. 4247–4271, 2007.

[68] J. Li and S. M. Meerkov, "On the coefficients of variation of uptime and downtime in manufacturing equipment," *Mathematical Problems in Engineering*, vol. 2005, no. 1, pp. 1–6, 2005.

[69] D. Jacobs and S. M. Meerkov, "A system-theoretic property of serial production lines: Improvability," *International Journal of Systems Science*, vol. 26, no. 4, pp. 755–785, 1995.

[70] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*. Springer Science & Business Media, 2012.

[71] H. Emmons and G. Vairaktarakis, *Flow shop scheduling: Theoretical Results, Algorithms, and Applications*. Springer Science & Business Media, 2012, vol. 182.

[72] W. K. Ching, "Machine repairing models for production systems," *International Journal of Production Economics*, vol. 70, no. 3, pp. 257–266, 2001.

[73] L. Haque and M. J. Armstrong, "A survey of the machine interference problem," *European Journal of Operational Research*, vol. 179, no. 2, pp. 469–482, 2007.

[74] M. Delasay, B. Kolfal, and A. Ingolfsson, "Maximizing throughput in finite-source parallel queue systems," *European Journal of Operational Research*, vol. 217, no. 3, pp. 554–559, 2012.

[75] M. Parlar and M. Sharafali, "Dynamic allocation of airline check-in counters: a queueing optimization approach," *Management Science*, vol. 54, no. 8, pp. 1410–1424, 2008.

[76] B. Hu and S. Benjaafar, "Partitioning of servers in queueing systems during rush hour," *Manufacturing & Service Operations Management*, vol. 11, no. 3, pp. 416–428, 2009.

[77] R. Wang, O. Jouini, and S. Benjaafar, "Service systems with finite and heterogeneous customer arrivals," *Manufacturing & Service Operations Management*, vol. 16, no. 3, pp. 365–380, 2014.

[78] J. A. Schwarz, G. Selinka, and R. Stolletz, "Performance analysis of time-dependent queueing systems: Survey and classification," *Omega*, vol. 63, pp. 170–189, 2016.

[79] C. N. Hasan and M. L. Spearman, "Determining job completion time distributions in stochastic production environments," in *Simulation Conference Proceedings,*. IEEE, 1995, pp. 837–845.

[80] B. Tan, "State-space modeling and analysis of pull-controlled production systems," in *Analysis and Modeling of Manufacturing Systems*. Springer, 2003, pp. 363–398.

[81] Z. Jia, L. Zhang, J. Arinez, and G. Xiao, "Finite production run-based serial lines with bernoulli machines: Performance analysis, bottleneck, and case study," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 1, pp. 134–148, 2016.

[82] A. Angius, A. Horváth, and M. Colledani, "Moments of accumulated reward and completion time in Markovian models with application to unreliable manufacturing systems," *Performance Evaluation*, vol. 75, pp. 69–88, 2014.

[83] J. Arinez, S. Biller, S. Marin, S. M. Meerkov, and L. Zhang, "Quantity/Quality improvement in an automotive paint shop: A case study," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 755–761, 2010.

[84] S.-Y. Chiang, C.-T. Kuo, J.-T. Lim, and S. M. Meerkov, "Improvability of assembly systems II: Improvability indicators and case study," *Mathematical Problems in Engineering*, vol. 6, no. 4, pp. 359–393, 2000.

[85] J. Li, D. E. Blumenfeld, and J. M. Alden, "Comparisons of two-machine line models in throughput analysis," *International Journal of Production Research*, vol. 44, no. 7, pp. 1375–1398, 2006.

[86] Y. Dallery and S. B. Gershwin, "Manufacturing flow line systems: a review of models and analytical results," *Queueing Systems: Theory and Applications*, vol. 12, no. 1–2, pp. 3–94, 1992.

[87] H. T. Papadopoulos and C. Heavy, "Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines," *European Journal of Operational Research*, vol. 92, no. 1, pp. 1–27, 1996.

[88] J. G. Kemeny and J. L. Snell, *Finite Markov chains*. van Nostrand Princeton, NJ, 1960, vol. 356.

[89] G. Chen, C. Wang, L. Zhang, J. Arinez, and G. Xiao, "Transient performance analysis of serial production lines with geometric machines," *IEEE Transactions on Automatic Control*, 2014.

[90] J. Li and S. M. Meerkov, "Due-time performance in production systems with Markovian machines," in *Analysis and Modeling of Manufacturing Systems*. Boston, MA: Kluwer Academic, 2003, ch. 10, pp. 221–253.

[91] G. Liberopoulos, G. Kozanidis, and P. Tsarouhas, "Performance evaluation of an automatic transfer line with WIP scrapping during long failures," *Manufacturing & Service Operations Management*, vol. 9, no. 1, pp. 62–83, 2007.

[92] S. M. Meerkov, N. Shimkin, and L. Zhang, "Transient behavior of two-machine geometric production lines," *IEEE Transactions on Automatic Control*, vol. 55, no. 2, pp. 453–458, 2010.

[93] R. O. Onvural, "Survey of closed queueing networks with blocking," *ACM Computing Surveys (CSUR)*, vol. 22, no. 2, pp. 83–121, 1990.

[94] L. M. Werner, "Analysis and design of closed loop manufacturing systems," DTIC Document, Tech. Rep., 2001.

[95] S. B. Gershwin and L. M. Werner, "An approximate analytical method for evaluating the performance of closed-loop flow systems with unreliable machines and finite buffers," *International Journal of Production Research*, vol. 45, no. 14, pp. 3085–3111, 2007.

[96] C. Paik, H. Kim, and H. Cho, "Performance analysis for closed-loop production systems with unreliable machines and random processing times," *Computers & Industrial Engineering*, vol. 42, no. 2, pp. 207–220, 2002.

[97] N. Maggio, A. Matta, S. Gershwin, and T. Tolio, "A decomposition approximation for three-machine closed-loop production systems with unreliable machines, finite buffers and a fixed population," *IIE Transactions*, vol. 41, no. 6, pp. 562–574, 2009.

[98] Y. Feng, X. Zhong, J. Li, and W. Fan, "Analysis of closed loop production lines in automotive body shops," in *IEEE International Conference on Automation Science and Engineering*, 2016, pp. 849–854.

[99] D. S. Kim, D. M. Kulkarni, and F. Lin, "An upper bound for carriers in a three-workstation closed serial production system operating under production blocking," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1134–1138, 2002.

[100] Z. Jia, L. Zhang, J. Arinez, and G. Xiao, "Performance analysis of assembly systems with bernoulli machines and finite buffers during transients," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1018–1032, 2016.

[101] ——, "Performance analysis of assembly systems with Bernoulli machines and finite buffers during transients," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 1018–1032, 2016.

[102] J. M. Harrison, "Assembly-like queues," *Journal of Applied Probability*, vol. 10, no. 2, pp. 354–367, 1973.

[103] U. N. Bhat, "Finite capacity assembly-like queues," *Queueing Systems*, vol. 1, no. 1, pp. 85–101, 1986.

[104] F. Bonomi, "An approximate analysis for a class of assembly-like queues," *Queueing Systems*, vol. 1, no. 3, pp. 289–309, 1987.

[105] E. H. Lipper and E. H. Sengupta, "Assembly-like queues with finite capacity: Bounds, asymptotics and approximations," *Queueing Systems*, vol. 1, no. 1, pp. 67–83, 1986.

[106] W. J. Hopp and J. T. Simon, "Bounds and heuristics for assembly-like queues," *Queueing Systems*, vol. 4, no. 2, pp. 137–155, 1989.

[107] C.-T. Kuo, J.-T. Lim, S. M. Meerkov, and E. Park, "Improvability theory for assembly system: Two component-one assembly machine case," *Mathematical Problems in Engineering*, vol. 3, pp. 95–171, 1997.

[108] P. C. Rao and R. Suri, "Approximate queueing network models for closed fabrication/assembly systems. part i: Single level systems," *Production and Operations Management*, vol. 3, no. 4, pp. 244–275, 1994.

[109] P. C. Rao and R. Suri., "Performance analysis of an assembly station with input from multiple fabrication lines," *Production and Operations Management*, vol. 9, no. 3, pp. 283–302, 2000.

[110] M. Manitz, "Queueing-model based analysis of assembly lines with finite buffers and general service times," *Computers & Operations Research*, vol. 35, no. 8, pp. 2520–2536, 2008.

[111] ——, "Analysis of assembly/disassembly queueing networks with blocking after service and general service times," *Annals of Operations Research*, pp. 1–25, 2014.

[112] J. K.-C. and Y.-D. Kim, "Performance analysis of assembly/disassembly systems with unreliable machines and random processing times," *IIE Transactions*, vol. 30, no. 1, pp. 41–53, 1998.

[113] S.-Y. Chiang, C.-T. Kuo, J.-T. Lim, and S. M. Meerkov, "Improvability of assembly systems I: Problem formulation and performance evaluation," *Mathematical Problems in Engineering*, vol. 6, no. 4, pp. 321–357, 2000.

[114] D. R. Alexander, I. Premachandra, and T. Kimura, "Transient and asymptotic behavior of synchronization processes in assembly-like queues," *Annals of Operations Research*, vol. 181, no. 1, pp. 641–659, 2010.

[115] N. Viswanadham and Y. Narahari, *Performance Modeling of Automated Manufacturing Systems*. Prentice Hall Englewood Cliffs, NJ, 1992.

[116] S. B. Gershwin and S. Gershwin, *Manufacturing Systems Engineering*. PTR Prentice Hall Englewood Cliffs, New Jersey, 1994.

[117] T. Altiok, *Performance Analysis of Manufacturing Systems*. Springer, 1997.

[118] J. Li and S. M. Meerkov, *Production Systems Engineering*. Springer, 2008.

[119] J. A. Buzacott and J. G. Shanthikumar, *Stochastic Models of Manufacturing Systems*. Prentice Hall Englewood Cliffs, NJ, 1993, vol. 4.

[120] K. Kissock, K. Hallinan, and W. Bader, "Energy and waste reduction opportunities in industrial processes," *Strategic planning for energy and the environment*, vol. 21, no. 1, pp. 40–53, 2001.

[121] G. Boyd *et al.*, "Development of a performance-based industrial energy efficiency indicator for cement manufacturing plants." Argonne National Laboratory (ANL), Tech. Rep., 2006.

[122] G. Boyd, E. Dutrow, and W. Tunnessen, "The evolution of the ENERGY STAR® energy performance indicator for benchmarking industrial plant manufacturing energy use," *Journal of Cleaner Production*, vol. 16, no. 6, pp. 709–715, 2008.

[123] Q. Chang, J. Ni, P. Bandyopadhyay, S. Biller, and G. Xiao, "Supervisory factory control based on real-time production feedback," *Journal of Manufacturing Science and Engineering*, vol. 129, no. 3, pp. 653–660, 2007.

[124] D. E. Blumenfeld and J. Li, "An analytical formula for throughput of a production line with identical stations and random failures," *Mathematical Problems in Engineering*, vol. 2005, no. 3, pp. 293–308, 2005.

[125] J. M. Alden and Burns, "General motors increases its production throughput," *Interfaces*, vol. 36, no. 1, pp. 6–25, 2006.

[126] J. Wang, J. Li, and N. Huang, "Optimal scheduling to achieve energy reduction in automotive paint shops," in *Proceedings of International Conference on the Manufacturing Science and Engineering*. American Society of Mechanical Engineers, 2009, pp. 161–167.

[127] C. A. Guerrero, J. Wang, J. Li, J. Arinez, S. Biller, N. Huang, and G. Xiao, "Production system design to achieve energy savings in an automotive paint shop," *International Journal of Production Research*, vol. 49, no. 22, pp. 6769–6785, 2011.

[128] F. K. Pil and S. Rothenberg, "Environmental performance as a driver of superior quality," *Production and Operations Management*, vol. 12, no. 3, pp. 404–415, 2003.

[129] C. J. Corbett and R. D. Klassen, "Extending the horizons: environmental excellence as key to improving operations," *Manufacturing & Service Operations Management*, vol. 8, no. 1, pp. 5–22, 2006.

[130] C. Galitsky, "Energy efficiency improvement and cost saving opportunities for the vehicle assembly industry: An energy star guide for energy and plant managers," *Lawrence Berkeley National Laboratory*, 2008.

[131] T. Gutowski and Murphy, "Environmentally benign manufacturing: Observations from japan, europe and the united states," *Journal of Cleaner Production*, vol. 13, no. 1, pp. 1–17, 2005.

[132] B. Saenz de Ugarte, A. Artiba, and R. Pellerin, "Manufacturing execution system–a literature review," *Production Planning and Control*, vol. 20, no. 6, pp. 525–539, 2009.

[133] Q. Chang, G. Xiao, S. Biller, and L. Li, "Energy saving opportunity analysis of automotive serial production systems," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 334–342, 2013.

[134] M. P. Brundage, Q. Chang, Y. Li, G. Xiao, and J. Arinez, "Energy efficiency management of an integrated serial production line and hvac system," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 789–797, 2014.

[135] M. Mashaei and B. Lennartson, "Energy reduction in a pallet-constrained flow shop through on–off control of idle machines," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 45–56, 2013.

[136] N. Frigerio and A. Matta, "Machine control policies for energy saving in manufacturing," in *Proceedings of IEEE International Conference on the Automation Science and Engineering*, 2013, pp. 651–656.

[137] ——, "Energy-efficient control strategies for machine tools with stochastic arrivals," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 1, pp. 50–61, 2015.

[138] ——, "Energy efficient control strategy for machine tools with stochastic arrivals and time dependent warm-up," *Procedia CIRP*, vol. 15, pp. 56–61, 2014.

[139] N. Frigerio, J. G. Shanthikumar, and A. Matta, "Dynamic programming for energy control of machine tools in manufacturing," in *IEEE International Conference on Automation Science and Engineering*, 2015, pp. 39–44.

[140] M. Yadin and P. Naor, "Queueing systems with a removable service station," *OR*, pp. 393–405, 1963.

[141] K. Baker, "A note on operating policies for the queue M/M/1 with exponential startup," *Infor*, vol. 11, no. 1, pp. 71–72, 1973.

[142] J. Arinez, S. Biller, S. M. Meerkov, and L. Zhang, "Quality/quantity improvement in an automotive paint shop: A case study," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 755–761, 2010.

[143] G. Chen, L. Zhang, J. Arinez, and S. Biller, "Feedback control of machine startup for energy-efficient manufacturing in bernoulli serial lines," in *Proceedings of IEEE Conference on the Automation Science and Engineering*, 2011, pp. 666–671.

[144] S. B. Gershwin, "Design and operation of manufacturing systems: the control-point policy," *IIE transactions*, vol. 32, no. 10, pp. 891–906, 2000.

[145] F. T. Chan, Z. Wang, and J. Zhang, "A two-level hedging point policy for controlling a manufacturing system with time-delay, demand uncertainty and extra capacity," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1528–1558, 2007.

[146] K. Sikdar, "The N threshold policy in the finite buffer GI/MSP/1 queues," *Quality Technology and Quantitative Management*, vol. 9, no. 4, pp. 355–373, 2012.

[147] 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, no. 5, pp. 528–543, 2013.