Transient Behavior Analysis of Serial Production Lines

Student Thesis

submitted by

Pengfei Zheng

Matriculation Number: 4729268

Institut für Robotik und Prozessinformatik

1. Examiner: Prof. Dr. -Ing. Klaus Dröder
2. Examiner: Prof. Dr. Jochen Steil

Technische Universität Braunschweig

2020.01.01

Eidesstattliche Erklärung

Ich erkläre, dass ich die vorliegende wissenschaftliche Arbeit selbständig, sowie ohne unerlaubte fremde Hilfe verfasst und nicht anderweitig für Prüfungszwecke vorgelegt habe. Alle verwendeten Quellen und Hilfsmittel wurden angegeben, sowie wörtliche und sinngemäße Zitate als solche gekennzeichnet.

Braunschweig, den 01.01.2020

Pengfei Zheng

Acknowledgement

I would like to express my special thanks to Prof. Dr. Jochen Steil as well as my supervisor Tianran Wang for professional guidance and valuable support. They helped me in understanding and going deep into my topic in the early stages of the work, and later also gave me many valuable suggestions and inspirations when I was in difficulty. I am really grateful to them.

Abstract

In the last few years, the research of redundant robots and bi-manual arm manipulator systems has experienced rapid progress among the robotic community.

contents

Keywords: Keywords, Geomatric Machine, Production Lines

Contents

List of Figures

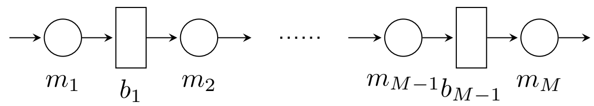
List of Tables

1 Introduction

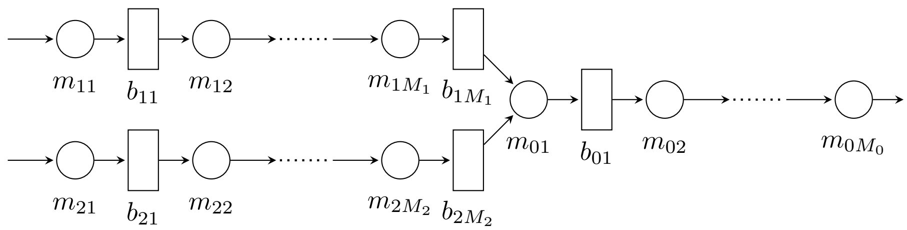
Production system has been studied widely during the last 65 years [1]. A production system is an industrial system that describe a procedure to transform from different resources into useful products. In this process, producing units (human operators, industrial robots, cells, etc.) and resource handling devices (shelves, carts, holders, vehicles, etc.) connected with each others so that desired products can be produced. It is a very important part of manufacturing research and application.

Extensive research has been invested in develpoing for design, modeling, improvement, analysis and control of production systems (for instance, monographs [2–5]). Despite the fact that in practice production systems may take several kinds of physical topologies, serial lines [see Figure 1.1(a)] and assembly systems [see Figure 1.1(b)] are the two most basic structures used in different manufacturing environments. In the literature, meanwhile, while serial production lines have been extensively investigated, assembly systems are been paid much less attention. Early research on assembly systems only considered the cases of multi-sequence-single-server, where different types of parts arrive at a single server in oder to be assembled together [4, 6]. Inspired by these works, few three-server system with limited sequence capacities have been studied [7, 8]. In these papers, dual servers represent component parts production, while the other server describes assembly operations. In addition, assembly systems based on queueing model have been further explored in papers [5, 9, 10] and the references within . The problem of steady-state performance evaluation of assembly system with unreliable machines and limited buffers has been studied in works [11–16] . In particular, the literature [11] developed a deconstruction technique to approximate the steady-state throughput for assembly systems based on machines with geometric models and identical processing times, while the article [12] focuses on assembly systems with geometric machines and a nonidentical processing time through turning the assembly system to a serial line. Furthermore, paper [13] develops the analysis to assembly system by geometric processing times.

The steady state behavior of production systems has been deeply studied in the last few years [15, 16]. Although it is often difficult to declare from the partial view that a production system is in steady state, the steady-state analysis approach is effective and accurate enough for manufacturing systems with large production capacity. The large production capacity allows the system to decay instantaneously to a negligible time compared to the global production run-time, and, therefore, allows it to use steady-state methods. Unfortunately, in addition to large-capacity manufacturing, there are a large number of mid- and small-capacity manufacturing systems in practice, which usually operate in a different method. In some these cases, one production line is usually able to produce several end products but can only produce one type of end product, equipment or special product at a time because of process. This usually lead to small- to medium-size production run-based operation based on the customer's order, where a production run contains only a specific amount of particular type of product. Obviously, when the



(a) Serial line.



(b) Assymbly system.

Figure 1.1: two production system

size of the production run is relatively small, the steady-state approaches cannot provide an ideal and accurate analysis of system. In some industries, a production run somtimes means a batch.

Nevertheless, the transient period of the behavior in production system has received much less research attention because of its complexity and the still large numbers of unsolved problems in steady state production systems research. On the other hand, recent research [17] has proved that transient analysis has become one of the most important fields in production systems research. Indeed, transient behavior of production systems have not been systematically studied and it is considered as one of the most significant directions in production systems research [17]. Specially, transient properties of serial production systems with two machines having the Bernoulli reliability model have been explored in [18–20] based in Markovian analysis. The research was later also extended to the case of multi-machine Bernoulli lines in [21, 22], which set up computationally efficient algorithms with recursive aggregation to approximate the transient performance with high accuracy.

Applications of Bernoulli line transient properties analysis is reported in [22–24]. Specailly, the paper [22] extended the algorithm developed in work [21] to the Bernoulli series production line with time-varying machine parameters. In spite of important results have been obtained regarding the transient behabior of the production system, it should be considered that most of the analysis studies cited above are only applicable to systems with machines based on Bernoulli reliability model, which can only be applied in the situations where the average machine downtime is comparable to its cycle time. Although the paper [25] tried to study the transient properties of serial production lines with machines in the geometric reliability model, the results were only applicable to the case of two-machine lines with an initial buffer occupancy at the beginning. For two-machine production lines with general initial state and longer lines, as far as we know, no analytical methods for have been built up for analysis of their transient performance because of larger dimension of the system state. Thus, the goal of this paper is to set up mathmatical models in python code with the help of the Bernoulli reliabile geomatric machine serial production line. Then, compare the results and consider the causes of the differences. In addition, an analysis of the relationship between the parameter improvement and trasient performance is evaluated.

The remainder of the papaer is organized as follows: Section II introduces the assumptions for the system and presents the performance measures of interest, and theory of Markov chain. Mathmatical modeling and behavior performance evaluation of individual machines, two-machine lines. and multi-machine lines are described in Section III. The conclusions and future work are given in Section VI.

2 Model and Performance Measures

This chapter mainly focus on the mathmatical model, the first part introduces the theoretical basis of the mathmatical mode, and the assumptions to build such mathmatical model, the second part presents the performance measures of transient properties.

2.1 Model

Before the discussion of the mathmatical mode, the classic conception of Markov chain should be first introduced as the theoretical basis of stochastic models to describe a process of real world. This made the foundation of for general stochastic simulation methods known such as Markov chain Mente Carlo, which are suitable for simulating sampling from comlex probability distributions, and also able to be applied in Bayesian statistics and artificial intelligence. Besides, for serial production lines with geometric machines and finite buffers, in particular, there are more assumptions to describe and derive such a system.

2.1.1 Markov chain

A Markov chain is a stochastic model basically describing a sequence of random events in which the probability of each event is related only to the state attained in the previous event [26]. In time-continuous situation, it is known as Markov process. It is named after the Russian mathematician Andrey Markov.

Here is a diagram (see Figure 2.1) representing a two state Markov process, where the states are labelled as A and B respectively. Every number represents the probability of the Markov process transforming from one state to another state with the arrow that indicates the direction. For instance, if the Markov process is now in state B, then the probability it transforms to state A is 0.6, while the probability it remains in state B is 0.4.

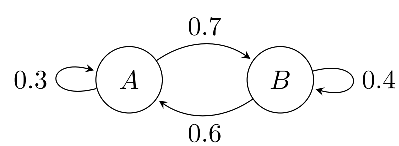


Figure 2.1: A two-state Markov process

The precise definition of the conception "Markov chain" will be given as following. A Markov process is a certain type of stochastic process distiguished by the Markov property [27]. A Markov chain is a Markov process with a calculable (namely, finit or denumerably infinite) number of states. The time parameters can be taken as the set of nonnegative integers or the set of nonnegative real numbers, so we have discrete parameter cases or continuous parameter cases. The adjective "simple" is sometimes used to qualify the Markov chain, but since we really do not discuss the "multiple" chains we should not treat it differently. In addition, we should only discuss the Markov chains which has a " stationary (or temporally homogeneous) transition probabilities" so that the qualifying expression in quotes will be understood. In the end, our discussion does not distinguish between a finite or a calculable infinite number of states and therefore we does not do any special treatment for the former case.

In this part we handle the case of a discrete parameter. Here the essential foundations can be summarized as follows.

Suppose we have an abstract set $\boldsymbol{\Omega}$, called the probability space, having the generic element ω, called elementary event; a Borel field $\mathscr{F}$ of subsets of $\boldsymbol{\Omega}$, called measurable sets or events, where $\boldsymbol{\Omega}$ is considered as a member; and a probability measure $\boldsymbol{P}$ defined on $\mathscr{F}$. The triple $(\boldsymbol{\Omega}, \mathscr{F}, \boldsymbol{P})$ is named as a probability triple. A set in $\mathscr{F}$ of probability zero will be named as a null set; "almost all ω (a.a.ω)" or "almost everywhere (a.e.)" expresses "all ω except a null set". The pair $(\mathscr{F}, \boldsymbol{P})$ will be regarded to be complete in the sense that each subset of a null set is attached to $\mathscr{F}$ and is a null set. If and only if a Borel subfield of $\mathscr{F}$ contains all null sets, it is said to be augmented. If a Borel subfield is given, there exists a unique smallest augmented Borel subfield of $\mathscr{F}$ including the given one. Unless exceptions specified all the following Borel fields will be presumed to be augmented. A (real) random variable is regarded as a single-valued function from a set Δ_0 in $\mathscr{F}$ to the closed real line $X = [-\infty, +\infty]$ thus for every real number c the set of ω in Δ_0 for which $x(\omega) \leq c$ is attached to $\mathscr{F}$. Δ_0 is named as the domain of definition of x and the set Δ of ω in Δ_0 for which $|x(\omega)| < \infty$ is named as the domain of finiteness of x. If neither domain is specific it will be figured out that $\boldsymbol{P}(\Delta) = 1$; otherwise stated, the random variable is finite-valued. It comes from the definition that if A is any Borel set in X, then the set of ω for which $x(\omega) \in A$, to be symbolized as $\omega : x(\omega) \in A$ is attached to $\mathscr{F}$. The probability of the set comes to be denoted by

$$\boldsymbol{P}\{x(\omega) \in A\}. \tag{2.1}$$

Symbols alike to these which will come later should be self-explanatory if we mark that commas or semicolons are served as symbolizing intersection of sets; for instance

$$\begin{aligned} \boldsymbol{P}\{x_\nu(\omega) \leq c_\nu, 1 \leq \nu \leq 2\} &= \boldsymbol{P}\{x_1(\omega) \leq c_1; x_2(\omega) \leq c_2\} \\ &= \boldsymbol{P}\left\{\cap_{n=1}^{2}[x_n(\omega) \leq c_n]\right\}. \end{aligned} \tag{2.2}$$

Suppose we hold a set $\{x_s, s \in S\}$ of random variables, the Borel field created by them is the smallest augmented Borel field regarding which all the random variables in the set are able to be measured. Special cases are $\mathscr{F}\{x_s, s \circ t\}$ where $\circ$ is one of the four symbols $<, \leq, >, \geq$.

The function F described by

$$F(u) = \boldsymbol{P}\{x(\omega) \leq u\} \tag{2.3}$$

for all real u is named as the distribution function of x. We get such that

$$\lim_{u \to +\infty} F(u) - \lim_{u \to -\infty} F(u) = \boldsymbol{P}(\Delta). \tag{2.4}$$

Unless otherwise stated, we will use a distribution function or a probability distribution that of a random variable which is limited with probability one to represent. Thus $\lim_{u\to-\infty} F(u) = 0$, $\lim_{u\to+\infty} F(u) = 1$. A random variable x is discrete if and only if there exists a calculable set A thus $P\{x(\omega) \in A\} = 1$. A possible value c of x is just one like $\boldsymbol{P}\{x(\omega) = c\} > 0$. All random variables engaged in the part are discrete. A function x from Δ_0 in $\mathscr{F}$ to a calculable set A can be taken as a random variable if and only if the set $\{\omega : x(\omega) = c\}$ is attached to $\mathscr{F}$ for every c in A. Actually A could be assigned as any abstract calculable set and we could define an abstract-valued random variable like this.

Suppose Λ_1 and Λ_2 are two sets in $\mathscr{F}$, the conditional probability of Λ_2 in relation to Λ_1 is defined By

$$\boldsymbol{P}(\Lambda_2|\Lambda_1) = \frac{\boldsymbol{P}(\Lambda_1\Lambda_2)}{\boldsymbol{P}(\Lambda_1)} \tag{2.5}$$

given that $\boldsymbol{P}(\Lambda_1) > 0$. When $\boldsymbol{P}(\Lambda_1) = 0$, $\boldsymbol{P}(\Lambda_2|\Lambda_1)$ is undefined. For convenience, undefined conditional probabilities will often be seen in following. If one that comes our from them is multiplied by a variable which is equal to 0, the product is considered to be 0. The conditional probability of the set $\{\omega : x_3(\omega) = c_3\}$ in relation to the set $\cap_{n-1}^{2}\{\omega : x_n(\omega) = c_n\}$ for example is symbolized as

$$\boldsymbol{P}\{x_3(\omega) = c_3|x_1(\omega) = c_1, x_2(\omega) = c_2\}. \tag{2.6}$$

For the random variables $\{x_\nu, 1 \le \nu \le n\}$, it is not necessarily finite-valued, and is said to be independent in case

$$\boldsymbol{P}\left\{\bigcap_{\nu=1}^{n}[x_\nu(\omega) \le c_\nu]\right\} = \prod_{\nu=1}^{n}\boldsymbol{P}\{x_\nu(\omega) \le c_\nu\} \tag{2.7}$$

for any real finite c_ν, $1 \le \nu \le n$. It comes out that the same equation remains true if the sets $\{\omega : x_\nu(\omega) \le c_\nu\}$ are changed to the more general sets $\omega : x_\nu(\omega) \in A_\nu$ where the A_ν are Borel sets in X. Suppose we habe a sequence $\{x_n, n \ge 1\}$, it is a sequence of independent random variables only if any finite number of them are independent. The denumerable sets Λ_ν, $1 \le \nu \le n$ or $1 \le \nu < \infty$ are independent only if their indicators are, the indicator of a set rendered as the function which on the set end is equal to one zero elsewhere.

The mathmatical expectation of a random variable x can be give by the abstract Lebesgue-Stieltjes integral

$$\boldsymbol{E}(x) = \int_{\boldsymbol{\Omega}} x(\omega)\boldsymbol{P}(d\omega). \tag{2.8}$$

In general, we extend this definition to a random variable that supposes that only if the integral existing, one of two values $+\infty$ or $-\infty$ with positive probability is finite or infinite. The conditional expectation of x relative to Λ, provided $\Lambda \in \mathscr{F}$, is defined as

$$\boldsymbol{E}(x|\Lambda) = \int_{\boldsymbol{\Omega}} x(\omega)\boldsymbol{P}(d\omega|\Lambda) = \frac{\int_{\boldsymbol{\Omega}} x(\omega)\boldsymbol{P}(d\omega)}{\boldsymbol{P}(\Lambda)}. \tag{2.9}$$

Specially if x is discrete with all its probable quantities in the measurable set A then

$$\boldsymbol{E}(x|\Lambda) = \frac{1}{\boldsymbol{P}} \sum_{i \in A} i \boldsymbol{P}\{\Lambda; x(\omega) = i\} \tag{2.10}$$

as long as the series is fully converged.

In this part the letters n, m, ν, r, s, t denote non-negative integers unless otherwise stated.

A discrete parameter stochastic process is described as a sequence of random variables $\{x_n, n \geq 0\}$ defined regarding a probability triple $(\boldsymbol{\Omega}, \mathscr{F}, \boldsymbol{P})$. If all random variables are discrete, the union $\boldsymbol{I}$ of all probable quantities of all x_n is a countable set named as the minimum state space of the process and every element of $\boldsymbol{I}$ is a state. Therefore, $i \in \boldsymbol{I}$ if and only if there is an $n \geq O$ so as $\boldsymbol{P}\{x_n(\omega) = i\} > O$. We take from the language of physics where the term "state" indicates that of a material system whose evolution in time is characterized by our stochastic process model.

A discrete parameter Markov chain can be defined by a sequence of discrete random variables $\{x_n, n \geq O\}$ possessing the under property: for arbitrary $n \geq 2, O \geq t_1 < ... < t_n$ and arbitrary $i_1, ..., i_n$ in the state space $\boldsymbol{I}$ we have

$$\begin{cases} \boldsymbol{P}\{x_{t_n}(\omega) = i_n | x_{t_1}(\omega) = i_1, ..., x_{t_{n-1}}(\omega) = i_{n-1}\} \\ \qquad\qquad = \boldsymbol{P}\{x_{t_n}(\omega) = i_n | x_{t_{n-1}}(\omega) = i_{n-1}\}. \end{cases} \tag{2.11}$$

whenever the left member is defined. The condition 2.11 will appear as the Markov property. It is identical to the under apparently weaker condition: for each $n \geq 1$,

$$\begin{aligned} &\boldsymbol{P}\{x_n(\omega) = i_n | x_0(\omega) = i_0, ..., x_{n-1}(\omega) = i_{n-1}\} \\ &\qquad = \boldsymbol{P}\{x_n(\omega) = i_n | x_{n-1}(\omega) = i_{n-1}\}. \end{aligned} \tag{2.12}$$

The proof here is neglected. An important consequence comes from condition 2.11 is that for arbitrary $n \geq 0$ and $0 \leq t_1 < ... < t_n ... < t_{n+m}$; and arbitrary $i_1, ..., i_n, ... m i_{n+m}$ in $\boldsymbol{I}$ we get

$$\begin{cases} \ \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m | x_{t_\nu}(\omega) = i_\nu, 1 \leq \nu \leq n-1\} \\ = \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m | x_{t_{n-1}}(\omega) = i_{n-1}\}. \end{cases} \tag{2.13}$$

The proof of 2.13 is carried out by induction on m. For $m = 0$, it decreases to 2.11. Supposing that 2.13 is true for a certain quantity of m, it comes to use the rules of combining conditional probabilities and 2.11,

$$\begin{aligned} &\boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m+1 | x_{t_\nu}(\omega) = i_\nu, 1 \leq \nu \leq n-1\} \\ &= \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m | x_{t_\nu}(\omega) = i_\nu, 1 \leq \nu \leq n-1\} \times \\ &\times \boldsymbol{P}\{x_{t_{n+m+1}}(\omega) = i_{n+m+1} | x_{t_\nu}(\omega) = i_\nu, 1 \leq \nu \leq n+m\} \\ &= \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m | x_{t_{n-1}}(\omega) = i_{n-1}\} \times \\ &\times \boldsymbol{P}\{x_{t_{n+m+1}}(\omega) = i_{n+m+1} | x_{t_{n+m}}(\omega) = i_{n+m}\} \\ &= \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m | x_{t_{n-1}}(\omega) = i_{n-1}\} \times \\ &\times \boldsymbol{P}\{x_{t_{n+m+1}}(\omega) = i_{n+m+1} | x_{t_\nu}(\omega) = i_\nu, n-1 \leq \nu \leq n+m\} \\ &= \boldsymbol{P}\{x_{t_\nu}(\omega) = i_\nu, n \leq \nu \leq n+m+1 | x_{t_{n-1}}(\omega) = i_{n-1}\}. \end{aligned} \tag{2.14}$$

Therefore 2.13 is true when m is replaced by $m+1$ and the induction is complete.

It is well know that a result comes from measure theory asserts that for all m the validity of 2.13 indicates that of the more general result: for each $\boldsymbol{M} \in \mathscr{F}\{x_t, t \geq t_n\}$ we get

$$\boldsymbol{P}\{\boldsymbol{M}|x_{t_\nu}(\omega) = i_\nu, 1 \leq \nu \leq n\} = \boldsymbol{P}\{\boldsymbol{M}|x_{t_n}(\omega) = i_n\}. \tag{2.15}$$

In this expression Markov property can be expressed verbally as positing that "If we know the state of an event at the previous moments, the probability of it is the same as only the last given state". More vaguely, it can be presented that "the past state is completely independent from the future state". Misleading by these description, it is a normal mistake to believe that Equation 2.15 stays true if the conditional events in it are replaced by a more general event, for instance, replacing i_ν with A_ν, where A_ν is a subset of $\boldsymbol{I}$. No doubt the resulting equation will generally be wrong. Nevertheless, the under extension of 2.15 is correct and trivial: if $\Lambda \in \mathscr{F}\{x_t, t \leq t_n\}$

$$\boldsymbol{P}\{\boldsymbol{M}|\Lambda; x_{t_n}(\omega) = i\} = \boldsymbol{P}\{\boldsymbol{M}|x_{t_n}(\omega) = i\}. \tag{2.16}$$

2.1.2 Descriptive Model

Consider a production line shown in Figure 2.2, in which circles illustrate machines and rectangles illustrate buffers. The system will be defined by the under assumptions.

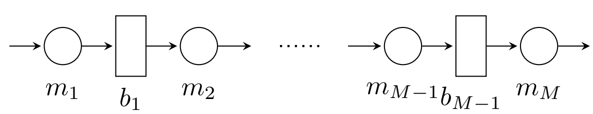


Figure 2.2: Serial production line.

.

1. The system is composed of M machines, arranged in series, and $M-1$ buffers between each consecutive pair of machines.

2. The machines all have the same constant cycle time τ. The time axis is limited in a slot duration τ. Machines start to operate at the beginning of each time slot.

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

$$\begin{aligned} Prob[s_i(n+1) = 0|s_i(n) = 1] =& P_i \\ Prob[s_i(n+1) = 1|s_i(n) = 1] =& 1 - P_i \\ Prob[s_i(n+1) = 1|s_i(n) = 0] =& R_i \\ Prob[s_i(n+1) = 0|s_i(n) = 0] =& 1 - R_i \end{aligned} \tag{2.17}$$

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

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

5. Machine $m_i, i = 2, ..., M$, 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.
6. Machine $m_i, i = 1, ..., M-1$, 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. Machine m_M is never blocked.
7. If a machine is up and neither starved or 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.
8. The system oprerates for a total of T time slots.

Remark 1: Under the assumption 3. the up- and downtime of machine m_i are geometric random variables. The average of its up- and downtime can be calculated as $T_{up,i} = 1/P_i$ and $T_{down,i} = 1/R_i$, respectively. In addition, the machine efficiency, i.e., the probability (proportion of time) that m_i is up in steady state, can be calculated as $e_i = T_{up,i}/(T_{up,i} + T_{down,i}) = R_i/(R_i + P_i)$.

Remark 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 [1,2,28,29] The geometric model has also been successfully applied in industrial case studies (see, for instance, [30, 31]).

Remark 3: 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 [32]). In this paper we consider time-dependent failures.

Remark 4: Assumption 6. 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 [17,33,34]). The analysis of systems defined by both conventions are similar. In fact, a system model under one convention can be converted to the model with the other convention by modifying the buffer capacity by one unit. In this paper we use BBS convention, since it leads to a simpler description [35].

Production systems with machines having other reliability models (e.g., exponential, Weibull, gamma, log-normal, and empirical, etc.) and non-identical cycle times will be studied in future work.

2.2 Performance Measures

To carry out rigorous real-time analysis and control of production systems, transient performance measures must be defined. In the framework of geometric serial lines defined by assumptions 1-8, the performance measures of interest are:

- *Production Rate*, $PR(n)$ = the expected number of finished parts produced by m_M in time slot n;
- *Consumption Rate*, $CR(n)$ = the expected number of raw parts consumed by m_1 in time slot n;
- *Work-in-process*, $WIP_i(n)$ = the expected number of parts in buffer b_i at the beginning of time slot n, $i = 1, ..., M - 1$;
- *Machine Starvation*, $ST_i(n)$ = Prob[m_i is starved by b_{i-1} in time slot n], $i = 2, ..., M$;
- *Machine Blockage*,$BL_i(n)$ = Prob[m_i is blocked by b_i in time slot n], $i = 1, ..., M-1$.

In this paper, we develop analytical methods to calculate these transient performance measures. It should be noted that, in this paper, the term "analytical" refers to formula-based calculation methods (either exact or approximate) as opposed to computer simulation.

3 Transient Performance of Various Geomatric Machines

In this chapter we will use the programming language to implement these mathmatical models from the simple individual machine to multi-machine and attempt to get the transient performance. In the second section, we will try to analysis the difference between this work and another one.

We choose to use python as the implementation language because python is a interpreted, general-purpose and object-oriented programming language. With its extensive mathmatics library and other third-party library, it is very popular to be used as a scientific scripting language to aid in a numerical data processing and manipulation problem like this.

3.1 Individual Geomatric Machine

Although transient analysis of individual geometric machines with constant parameters has been studied in [25], as the basis of all the following models, we still take it as the primary work. Since the performance evaluation method which we try to derive is the foundation of the study in the this paper, we briefly introduce it below.

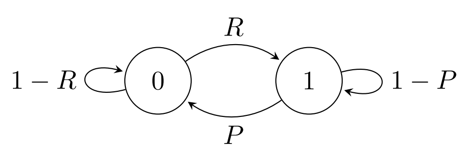


Figure 3.1: State transition diagram of one geometric machine

The state transition schema for an individual geometric machine is illustrated in Figure 3.1. We use $x_i(n), i \in \{0 = down, 1 = up\}$ to indicate the probability that the machine is in state i during time slot n, that is $x_i(n) = Prob[s(n) = i]$. Apparently, the system is described by a two-state ergodic Markov chain and the transformmation of state vector $x(n) = [x_0(n) \quad x_1(n)]^T$ can be described by

$$x(n+1) = A_1 x(n), x_0(n) + x_1(n) = 1 \tag{3.1}$$

where

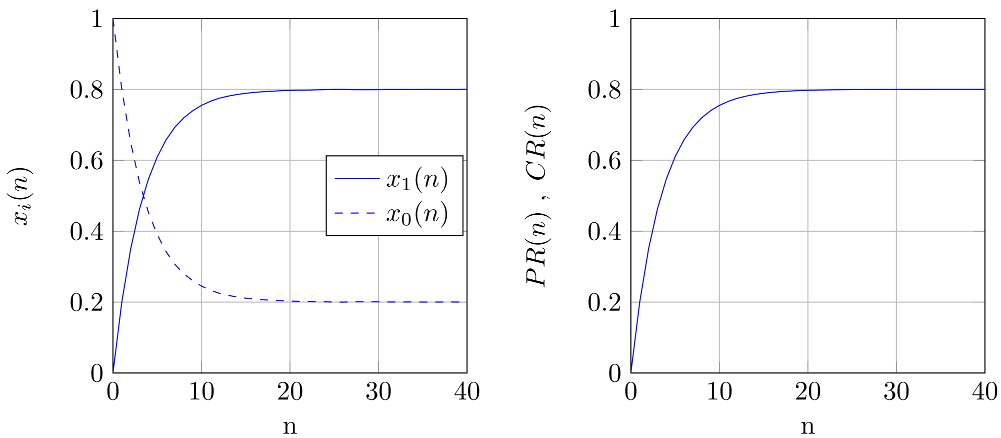
$$A_1 = \begin{bmatrix} 1-R & P \\ R & 1-P \end{bmatrix} \tag{3.2}$$

The production rate and consumption rate of an individual machine with the original state both down(0) and up(1) can be calculated as

$$PR(n) = CR(n) = x_1(n) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(n) = \begin{bmatrix} 0 & 1 \end{bmatrix} A_1^n x(0) \tag{3.3}$$

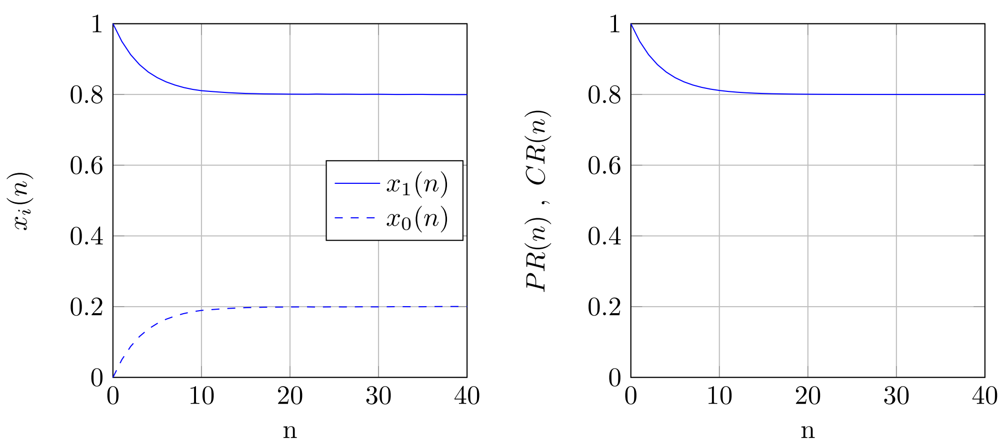
the Figure 3.2 is the contrast of simulation and calculation of one geometric machine with initial state of down with the parameters of breakdown probability $P = 0.05$ and repair probability $R = 0.2$. which are both linear in machine state $x(n)$.

As an illustration, consider a geometric machine with breakdown probability $P = 0.05$ and repair probability $R = 0.2$. The transients of the system state and the performance measures are given in Figure 3.2 and 3.3, assuming the machine is initially down and up, respectively. As one can see, the initial condition of a machine has strong impact on system transients—which may result in production loss (see Fig. 3) or production gain (see Fig. 4).



(a) Result of simulation (b) Result of calculation

Figure 3.2: Transients of an individual geomatric machine when it is initially down



(a) Result of simulation (b) Result of calculation

Figure 3.3: Transients of an individual geomatric machine when it is initially up

In the python code, we use the object-oriented features to help build the model. We seperate the codes into two parts. The first part is a class file called Individual. A `class Individual` represents a geomatirc machine that runs in a two-state Markov chain. It holds the parameters, which are transformed from another file, and calculates once a time slot till the end of the time control parameter `n` changes to zero.

```
import random


class Individual:

    def __init__(self, Pi: float, Ri: float, n: int, X_init=0):
        assert (0 <= Pi and Pi <= 1) is True
        assert (0 <= Ri and Ri <= 1) is True
        assert (0 == X_init or X_init == 1) is True
        self.Pi = Pi  # breakdown probability
        self.Ri = Ri  # repair probability
        self.n = n  # number of slot
        self.x_n = X_init  # state of x

    def run_once(self) -> int:  # simulation of a time slot
        if self.x_n == 0:
            if random.random() <= self.Ri:
                self.x_n = 1
        else:
            if random.random() <= self.Pi:
                self.x_n = 0
        return self.x_n
```

Another file, which is used to call the **class Individual**, are also attached as following. The main purpose of this file is to calculate the the average values of all the evaluation performance in order to get the mathmatical expectation. The final daten are collected in the file called `result.txt`.

```
from individual import Individual
P = 0.05  # break down probability
R = 0.2  # repair probability
n = 40  # numbers of slot
X_init = 1  # initial state of machine
max_range = 1000000  # maxmal quantity of the production lines
sum = []
indi = []

for i in range(max_range):  # initial state of 1
    indi.append(Individual(P, R, n, X_init))

for i in range(n):
    sum.append(0)
    for j in range(max_range):
        sum[i] += indi[j].run_once()
    sum[i] = (sum[i]/max_range)
```

```
str_sum = 'initial 1\n'
i = 1
str_sum += 'x=1\n'
for e in sum:
    str_sum += '(' + str(i) + ',' + str(e) + ')\n'
    i += 1

i = 1
str_sum += 'x=0\n'
for e in sum:
    str_sum += '(' + str(i) + ',' + str(1-e) + ')\n'
    i += 1

# force initial
indi = []
sum = []

for i in range(max_range):  # initial state of 0
    indi.append(Individual(P, R, n))

for i in range(n):
    sum.append(0)
    for j in range(max_range):
        sum[i] += indi[j].run_once()
    sum[i] = (sum[i]/max_range)

str_sum += 'initial 0\n'
i = 1
str_sum += 'x=1\n'
for e in sum:
    str_sum += '(' + str(i) + ',' + str(e) + ')\n'
    i += 1

i = 1
str_sum += 'x=0\n'
for e in sum:
    str_sum += '(' + str(i) + ',' + str(1-e) + ')\n'
    i += 1

file = open("result.txt", 'w')
file.truncate()
file.write(str_sum)
file.close()
```

3.2 Transient Performance of Two-Machine Geometric Lines

Consider a two-machine geometric line defined by assumptions 1-8. illustrated in Figure 3.4. It is easy to show that the system is characterized by an ergodic Markov chain. In addition to machine state $si(n)$, let $h(n)$ denote the number of parts in the buffer at the beginning of time slot n. Then, the state of the Markov chain is defined by a triple $(h(n), s_1(n), s_2(n))$, where $h(n) \in 0, 1, ..., N$ and $s_1(n), s_2(n) \in 0, 1$. Clearly, the system has a total of $4(N+1)$ states. To calculate the transition probabilities among these states, we arrange the states in the following manner: Let $r(h, s1, s2)$ denote the state number of the Markov chain state (h, s_1, s_2), $h \in 0, 1, ..., N, s_1, s_2 \in 0, 1$. defined

$$r(h, s_1, s_2) = 4h + 2s_1 + s_2 + 1. \tag{3.4}$$

Then, the arrangement of the $4(N+1)$ system states can be summarized in Table 3.1. In

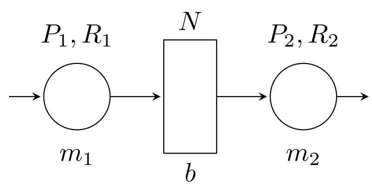


Figure 3.4: State transition diagram of two geometric machine with buffer

other words, each system state is assigned a unique number ranging from 1 to $4(N+1)$. For example, State 1 is when both machines are down and the buffer is empty, while State $4(N+1)$ is when both machines are up and the buffer is full. In addition, given a state number r, its corresponding buffer and machines states can be calculated as follows:

$$\begin{aligned} h^{(r)} &= \left\lfloor \frac{r-1}{4} \right\rfloor, s_1^{(r)} = \left\lfloor \frac{r-1-4h^{(r)}}{2} \right\rfloor \\ s_2^{(r)} &= \left\lfloor \frac{r-1-4h^{(r)-2s_1^{(r)}}}{1} \right\rfloor \end{aligned} \tag{3.5}$$

where $\lfloor a \rfloor$ represents the largest integer not greater than a.

Table 3.1: Arrangement of the System States $k = 0, 1, ..., N$

State number(r)	h	s_1	s_2
$4k+1$	k	0	0
$4k+2$	k	0	1
$4k+3$	k	1	0
$4k+4$	k	1	1

Note that the transition of $h(n)$ is deterministic given $s_1(n)$ and $s_2(n)$. For serial production lines, the expressions that describe the dynamics of $h(n)$ have been derived in [21]

$$h(n+1) = h'(n) + s_1(n)min\{N - h'(n), 1\} \tag{3.6}$$

where

$$h'(n) = h(n) - s_2(n)min\{h(n), 1\} \tag{3.7}$$

In the above expressions, $h'(n)$ represents the occupancy of the buffer as soon as machine m_2 removes a part from the buffer at the beginning of time slot n.

The transitions of $s_i(n)$'s, on the other hand, are probabilistic based on 2.17. Therefore, we can examine each of the $4(N+1)$ states, then, based on 3.6 and 3.7, identify all possible destination states after one time slot by enumerating all four combinations of $s_1(n)$ and $s_2(n)$, and, finally, calculate the corresponding transition probabilities using 2.17. Let A_2 denote the transition probability matrix obtained and let $x_i(n)$, $i \in 1, 2, ..., 4(N+1)$, denote the probability that the system, i.e., the Markov chain, is in state i during time slot n. Then, the evolution of the system state, $x(n) = [x_1(n)x_2(n)...x_4(N+1)^{(n)}]^T$, is given by

$$x(n+1) = A_2 x(n), \sum_{i=1}^{4(N+1)} x_i(n) = 1. \tag{3.8}$$

According the state arrangement 3.4, we have

$$\begin{aligned}
x_{4h+1}(n) =& \text{Prob}[m_1 \text{ down}, m_2 \text{ down, buffer } b \text{ has } h \text{ parts at time } n] \\
x_{4h+2}(n) =& \text{Prob}[m_1 \text{ down}, m_2 \text{ up, buffer } b \text{ has } h \text{ parts at time } n] \\
x_{4h+3}(n) =& \text{Prob}[m_1 \text{ up}, m_2 \text{ down, buffer } b \text{ has } h \text{ parts at time } n] \\
x_{4h+4}(n) =& \text{Prob}[m_1 \text{ up}, m_2 \text{ up, buffer } b \text{ has } h \text{ parts at time } n].
\end{aligned} \tag{3.9}$$

Therefore, based on the definitions given in 2.2,the performance measures of the two-machine geometric line system can be calculated as follows:

$$\begin{aligned}
PR(n) &= \text{Prob}[m_2 \text{ up,buffer } b \text{ not empty during time } n] \\
&= C_1 x(n) = [C_{1,0}\ C_{1,1}\ ...\ C_{1,N}]x(n) \\
CR(n) &= \text{Prob[m1 up and not blocked during time n]} \\
&= C_2 x(n) = [C_{2,0}\ C_{2,1}\ ...\ C_{2,N}]x(n) \\
WIP(n) &= \sum_{i=1}^{N} i \cdot \text{Prob[buffer } b \text{ has } i \text{ parts at time } n] \\
&= C_3 x(n) = [C_{3,0}\ C_{3,1}\ ...\ C_{3,N}]x(n) \\
ST_2 &= \text{Prob}[m_2 \text{ up and buffer } b \text{ empty at time } n] \\
&= C_4 x(n) = [C_{4,0}\ C_{4,1}\ ...\ C_{4,N}]x(n) \\
BL_1 &= \text{Prob}[m_1 \text{ up}, m_2 \text{ down,buffer } b \text{ is full at time } n] \\
&= C_5 x(n) = [C_{5,0}\ C_{5,1}\ ...\ C_{5,N}]x(n)
\end{aligned} \tag{3.10}$$

where

$$\begin{aligned}
C_{1,0} &= [0000], \quad C_{1,i} = [0101], \quad i = 1, ..., N \\
C_{2,N} &= [0001], \quad C_{2,i} = [0011], \quad i = 0, .., N-1 \\
C_{3,i} &= [iiii], \qquad\qquad i = 0, ..., N \\
C_{4,0} &= [0101], \quad C_{4,i} = [0000], \quad i = 1, ..., N \\
C_{5,N} &= [0010], \quad C_{5,i} = [0000], \quad i = 0, ..., N-1
\end{aligned} \tag{3.11}$$

Therefore, all these performance measures are linear in system state $x(n)$.

As an illustration, consider a two-machine geometric line defined by ssumptions 1-8. with machine and buffer parameters

$$P_1 = 0.03, R_1 = 0.18, P_2 = 0.06, R_2 = 0.21, N = 10.$$

Assume that both machines are initially down and the buffer is initially empty. the Figure 3.5 is the the result of simulation of two machine with buffer with the paramters of $P_1 = 0.03, R_1 = 0.18, P_2 = 0.06, R_2 = 0.21$, Buffer $N = 10$.

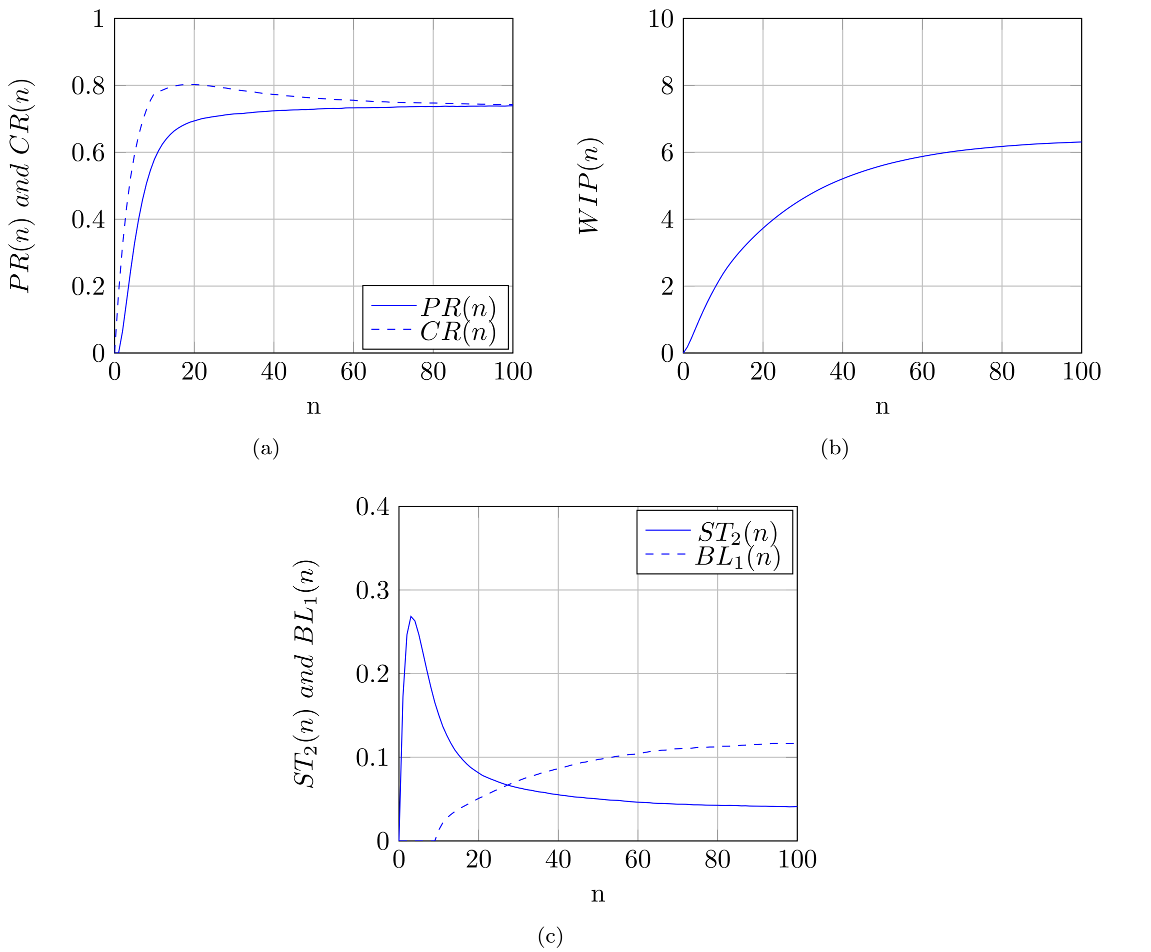


Figure 3.5: Transients of a two-machine geometric line. (a) $PR(n)$ *and* $CR(n)$;(b) $WIP(n)$; (c) $ST_2(n)$ *and* $BL_2(n)$

3.3 Transient Performance of Multi Machine Lines

Consider an M-machine geometric line defined by assumptions 1-8. Due to the memoryless property of geometric distribution, the system is still characterized by a Markov chain. Let $h_i(n)$ denote the number of parts in buffer b_i at the beginning of time slot n. Then, the state of the Markov chain is defined by vector the following is the part about multi geometric machines with four machine and the parameters are $P_i = 0.05, R_i = 0.2, i = 1, ..., 4; N_i = 5, i = 1, ..., 3$.

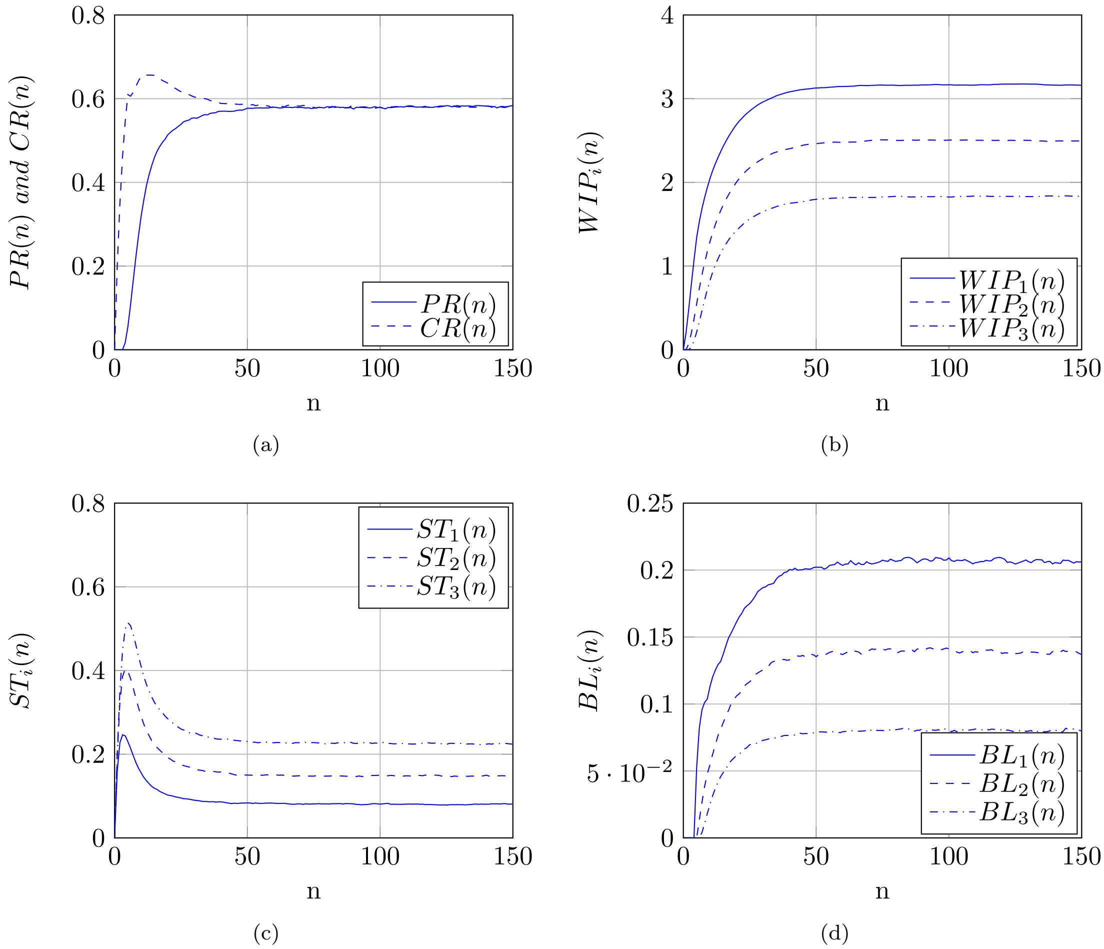


Figure 3.6: Transients of a four-machine geometric line. (a) $PR(n)$ *and* $CR(n)$;(b) $WIP(n)$; (c) $ST_i(n)$;(d) $BL_i(n)$.

3.4 Result Analysis

There's no apparently difference between the work of this paper and [36] in one machine mode, but it shows difference in two-machine lines especially in $ST_2(n)$ and BL_1 these two parameters. From the Figure 3.7 can we tell the differece. And in the four-machine lines model there are more difference shown in Figure 3.8.

For the reasons of these difference, we suppose a few points: First, we doubt that whether the lack of enough quantity of samples cause the problem. But when we enlarege the quantity of simulation time from 10000 to 100000, we found no apparent difference from the result. Second, due to that we don't know what kind of programming language the reseachers coding with, a probable cause may be we take a different platform to

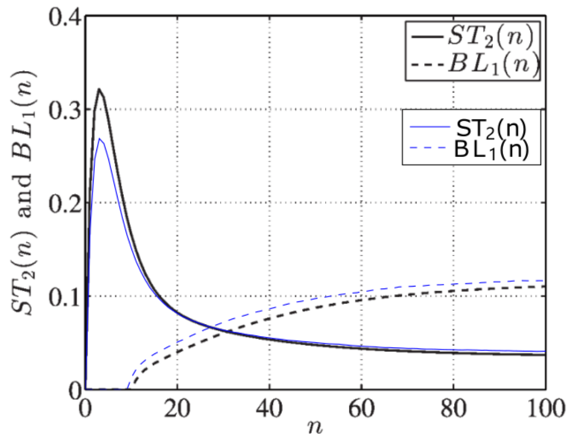


Figure 3.7: Performance contrast of a two-machine geometric line $ST_2(n)$ *and* $BL_2(n)$

programm with, and that may cause some accuracy problem. Third, we also doubt that the researchers may do some kind of curve smoothing in oder to present a perfect result.

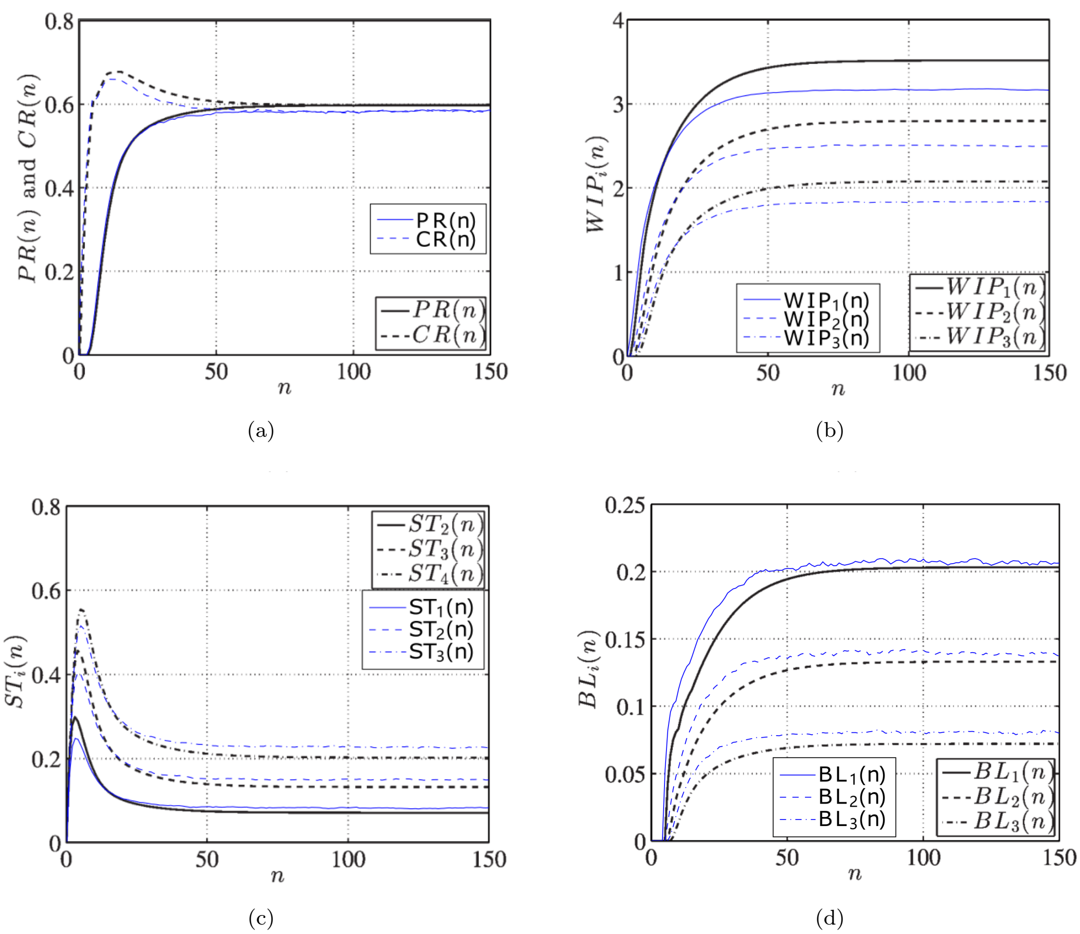


Figure 3.8: Transients of a four-machine geometric line. (a) $PR(n)$ *and* $CR(n)$;(b) $WIP(n)$; (c) $ST_i(n)$;(d) $BL_i(n)$.

4 Impact Analysis in Variation of the Buffer Size

In this chapter we attepmt to explorer the relationship between the paramters and the transient performance of a multi-machine line with five machines. A suitebale buffer size can help to improve the transient performance of the production line.

5 Conclusion

List of References

[1] Papadopolous, HT ; Heavey, Cathal ; Browne, Jimmie: *Queueing theory in manufacturing systems analysis and design*. Springer Science & Business Media, 1993

[2] Buzacott, John A. ; Shanthikumar, J G.: *Stochastic models of manufacturing systems*. Bd. 4. Prentice Hall Englewood Cliffs, NJ, 1993

[3] Askin, Ronald G. ; Standridge, Charles R.: *Modeling and analysis of manufacturing systems*. John Wiley & Sons Inc, 1993

[4] Bonomi, Flavio: An approximate analysis for a class of assembly-like queues. In: *Queueing Systems* 1 (1987), Nr. 3, S. 289–309

[5] RAO, P C. ; Suri, Rajan: Performance analysis of an assembly station with input from multiple fabrication lines. In: *Production and Operations Management* 9 (2000), Nr. 3, S. 283–302

[6] Harrison, J M.: Assembly-like queues. In: *Journal of Applied Probability* 10 (1973), Nr. 2, S. 354–367

[7] Kuo, C-T ; Lim, J-T ; Meerkov, SM ; Park, E: Improvability theory for assembly systems: Two component—One assembly machine case. In: *Mathematical Problems in Engineering* 3 (1996), Nr. 2, S. 95–171

[8] Lipper, EH ; Sengupta, Bhaskar: Assembly-like queues with finite capacity: bounds, asymptotics and approximations. In: *Queueing Systems* 1 (1986), Nr. 1, S. 67–83

[9] Manitz, Michael: Queueing-model based analysis of assembly lines with finite buffers and general service times. In: *Computers & Operations Research* 35 (2008), Nr. 8, S. 2520–2536

[10] Rao, P C. ; Suri, Rajan: Approximate queueing network models for closed fabrication/assembly systems. part I: Single level systems. In: *Production and Operations Management* 3 (1994), Nr. 4, S. 244–275

[11] Gershwin, Stanley B.: Assembly/disassembly systems: An efficient decomposition algorithm for tree-structured networks. In: *IIE TRANSACTIONS* 23 (1991), Nr. 4, S. 302–314

[12] Liu, Xiao-Gao ; Buzacott, John A.: Approximate models of assembly systems with finite inventory banks. In: *European Journal of Operational Research* 45 (1990), Nr. 2-3, S. 143–154

[13] Helber, Stefan: Decomposition of unreliable assembly/disassembly networks with limited buffer capacity and random processing times. In: *European Journal of Operational Research* 109 (1998), Nr. 1, S. 24–42

[14] MASCOLO, MARIA D. ; David, Rene ; Dallery, Yves: Modeling and analysis of assembly systems with unreliable machines and finite buffers. In: *IIE transactions* 23 (1991), Nr. 4, S. 315–330

[15] Chiang, S-Y ; Kuo, C-T ; Lim, J-T ; Meerkov, SM: Improvability of assembly systems I: Problem formulation and performance evaluation. In: *Mathematical Problems in Engineering* 6 (1999), S. 321–357

[16] Chiang, S-Y ; Kuo, C-T ; Lim, J-T ; Meerkov, SM: Improvability of assembly systems II: Improvability indicators and case study. In: *Mathematical Problems in Engineering* 6 (1999), S. 359–393

[17] Li, Jingshan ; E. Blumenfeld, Dennis ; Huang, Ningjian ; M. Alden, Jeffrey: Throughput analysis of production systems: recent advances and future topics. In: *International Journal of Production Research* 47 (2009), Nr. 14, S. 3823–3851

[18] Meerkov, Semyon M. ; Zhang, Liang: Transient behavior of serial production lines with Bernoulli machines. In: *IIE Transactions* 40 (2008), Nr. 3, S. 297–312

[19] Meerkov, SM ; Zhang, L: Transients in production lines with two non-identical Bernoulli machines. In: *Proceedings of 7th International Conference on the Analysis of Manufacturing Systems*, 2009, S. 212–221

[20] Meerkov, Semyon M. ; Zhang, Liang: Unbalanced production systems with floats: Analysis and lean design. In: *International journal of manufacturing technology and management* 23 (2011), Nr. 1-2, S. 4–15

[21] Zhang, Liang ; Wang, Chuanfeng ; Arinez, Jorge ; Biller, Stephan: Transient analysis of Bernoulli serial lines: Performance evaluation and system-theoretic properties. In: *IIE Transactions* 45 (2013), Nr. 5, S. 528–543

[22] Chen, Guorong ; Zhang, Liang ; Arinez, Jorge ; Biller, Stephan: Energy-efficient production systems through schedule-based operations. In: *IEEE Transactions on Automation Science and Engineering* 10 (2012), Nr. 1, S. 27–37

[23] Wang, Junwen ; Hu, Yao ; Li, Jingshan: Transient analysis to design buffer capacity in dairy filling and packing production lines. In: *Journal of Food Engineering* 98 (2010), Nr. 1, S. 1–12

[24] Chen, Guorong ; Zhang, Liang ; Arinez, Jorge ; Biller, Stephan: Feedback control of machine startup for energy-efficient manufacturing in Bernoulli serial lines. In: *2011 IEEE International Conference on Automation Science and Engineering* IEEE, 2011, S. 666–671

[25] Meerkov, Semyon M. ; Shimkin, Nahum ; Zhang, Liang: Transient behavior of two-machine geometric production lines. In: *IEEE Transactions on Automatic Control* 55 (2010), Nr. 2, S. 453–458

[26] Gagniuc, Paul A.: *Markov chains: from theory to implementation and experimentation.* John Wiley & Sons, 2017

[27] Chung, Kai L.: Markov chains. In: *Springer-Verlag, New York* (1967)

[28] Gershwin, Stanley B. ; Gershwin, SB: Manufacturing systems engineering. (1994)

[29] Altiok, Tayfur: *Performance analysis of manufacturing systems.* Springer Science & Business Media, 1997

[30] Li, Jingshan ; Meerkov, Semyon M.: Due-time performance of production systems with Markovian machines. In: *Analysis and modeling of manufacturing systems.* Springer, 2003, S. 221–253

[31] Liberopoulos, George ; Kozanidis, George ; Tsarouhas, Panagiotis: Performance evaluation of an automatic transfer line with WIP scrapping during long failures. In: *Manufacturing & Service Operations Management* 9 (2007), Nr. 1, S. 62–83

[32] Li, Jingshan ; Blumenfeld, Dennis E. ; Alden, Jeffrey M.: Comparisons of two-machine line models in throughput analysis. In: *International journal of production research* 44 (2006), Nr. 7, S. 1375–1398

[33] Dallery, Yves ; Gershwin, Stanley B.: Manufacturing flow line systems: a review of models and analytical results. In: *Queueing systems* 12 (1992), Nr. 1-2, S. 3–94

[34] Papadopoulos, HT ; Heavey, Cathal: Queueing theory in manufacturing systems analysis and design: A classification of models for production and transfer lines. In: *European journal of operational Research* 92 (1996), Nr. 1, S. 1–27

[35] Li, Jingshan ; Meerkov, Semyon M.: *Production systems engineering.* Springer Science & Business Media, 2008

[36] Chen, Guorong ; Wang, Chuanfeng ; Zhang, Liang ; Arinez, Jorge ; Xiao, Guoxian: Transient performance analysis of serial production lines with geometric machines. In: *IEEE Transactions on Automatic Control* 61 (2015), Nr. 4, S. 877–891