Factors Affecting the Variance, the Bias and the MSE of Time Averages in Markovian Event Systems

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Master of Science
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Sanjeev Sethi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

In simulation, time averages are important for estimating equilibrium parameters. In particular, we would like to have the variance, bias and mean-square error for time averages. First, we will discuss various factors and their effect on the bias, the variance and the mean-square error. We will use the Markovian Event System to model various systems, including $M/M/1$ queues, $M/E_k/1$ queues, $M/M/c$ queues, sequential queues, inventory systems and queueing networks. We use a numerical method given in [27] for the computation of the variance, the bias and the mean-square error of the time average. The effectiveness of the method is tested by experimenting with models of various stochastic systems. The contribution of this thesis is to use numerical and graphical interpretations to study the general characteristics of the measures. The important characteristics included in our study are decomposability and periodicity.

Acknowledgements

I am deeply grateful to my supervisor Dr. Grassmann for supervising and providing financial support throughout this research study. He has always been very encouraging and patient with me. I am grateful for his guidance and contributions to this thesis. I appreciate the efforts put in me by him at every step.

I extend my gratitude to the committee members, Dr. Grant Cheston and Dr. Nathaniel Osgood for their insightful comments, suggestions and input that contributed to the success of this thesis. I would also like to thank all faculty and staff members for their support and concern.

My thanks are due to my parents, parents in law, siblings and friends for their encouragement and kind concern. Special thanks to my wife Archana Anand for her love, moral support and encouragement at all times.

Above all, I thank God for blessing me with the strength, wisdom and inspiration to work throughout this thesis.

DEDICATION

I dedicate this thesis to my wife Archana Anand, who is the guiding spirit behind all my endeavours and who always supports me.

Contents

List of Tables

List of Figures

List of Abbreviations

CTMC	Continuous Time Markov Chain
DES	Discrete Event System
DTMC	Discrete Time Markov Chain
EAES	Entity Attribute Event Systems
IID	Independent and Identically Distributed
MES	Markovian Event System
MSE	Mean Square Error
SVES	State Variable Event System

Chapter 1

Introduction

The computer-based stochastic simulation of discrete-time stochastic processes is a commonly used method for performance evaluation of various systems. It is predominantly used to gain insight into the steady-state behavior of queueing processes by estimation of steady-state statistical parameters of a system such as the steady-state mean. Other major goals for conducting a simulation experiment [62] are optimizing a system or process under uncertainty, finding the most significant factors affecting system performance, predicting performance of real or proposed systems, comparing several operating strategies and evaluating existing systems. The flexibility and intuitiveness of simulation makes it one of the most widely accepted and used tools for systems analysis and decision making. Simulation can be used to investigate nearly any type of stochastic system by studying an abstract *model* of the relevant process or system. Some examples of the application of simulation include the operation of queueing systems, telecommunication systems, the operation of manufacturing systems, operation of distribution systems, financial risk analysis, healthcare applications, inventory systems and many more (see Hillier and Lieberman [38], Banks, Carson, and Nelson [3], Law and Kelton [43]).

Often decisions made from simulation models require the estimation of average values, probabilities of occurrence of an outcome or measures of variability of random variables. Most common issues that relate to outputs of simulation models are:

1. Inference about the performance of real systems based on results from simulation models.
   We will not consider this issue in our thesis.

2. Underlying variability tied with the simulation model.
   Since the model represents a stochastic system or process with random elements, the outputs produced will be probabilistic. The issue of variability concerns the precision and sensitivity of the model when the simulation is conducted more than once or run for a longer time.

1.1 Motivation and Objective

A simulation study is frequently used to estimate the mean value of a parameter of a process. Typically averages from long simulation runs are used to characterize a system in steady state. In

this thesis, we will use the time average to analyze the steady state behavior of a system. While we are primarily concerned with the long run average number of customers in a system, the time average can represent the long run inventory cost, number of jobs present, etc.

The simulation run of a process beginning in a certain state is initially in a non-stationary phase. According to Oni [52], the simulation output is often contaminated by the presence of transient elements, although these transient elements usually decay exponentially over time. Morse [48] calls the time for the transient to die down the *relaxation time* of the system. This time is related to the *burn-in phase* [29] or *warm-up period* [43, 8] of a system. Ignoring the existence of this period can lead to a significant bias in the final results. The behavior of such systems or processes during the startup or transient period can be analyzed analytically [21] or by using simulation [43]. Subsequently, if the process is stable, it moves asymptotically toward a steady-state (statistical equilibrium), though different parameters converge to their steady state with different rates. For this reason, we will assume that the statistics collected refer to steady state. Morse [49] initiated the analysis of relaxation times by considering the correlation function of the $M/M/1$ queue length. The results show that more heavily loaded systems will move more slowly to their statistical equilibrium. The initialization bias phenomenon caused by the slow convergence to the steady-state results in a bias in the statistics computed from an observed time series. The problem is to reduce the bias or to remove it completely. This problem has been a long outstanding issue in simulation methodology, and has motivated simulation experts to conduct many studies [49, 15, 6, 13, 46, 5, 17, 52, 22, 58, 50, 70, 71, 8, 21, 67, 66, 65].

The estimation of the unknown variance of the time average, which is required for estimation of confidence intervals, is one of the main goals of a simulation study. Unfortunately, an important analytical problem encountered in the analysis of simulation results is that the observations are correlated [49, 57, 59, 10, 36, 40, 72, 4, 44], and thus do not satisfy the precondition of statistical independence. The discussion initiated by Conway in [8] lead to a variety of proposed methods for data collection and statistical analysis from steady state simulation to get around the nonstationarity caused by initial transient period and the autocorrelation of events. These methods either try to take advantage of the correlated nature of the observations, or to weaken/eliminate the autocorrelations among the observations for determining confidence intervals for the parameters estimated. The problem of the autocorrelated nature of the original output data is overcome in the *method of replications* [2] also known as *replication/deletion method*. However, there are different opinions on the efficiency of this method as compared to the other methods of data collection and analysis, all of which are based on a long single run of the simulation experiment. Law and Kelton [43] argue in favor of the method of independent replications, but a number of other authors such as Whitt [69], Conway [8] and Cheng [6], support the long single run approach in steady-state simulation. The method of batch means [2, 60, 7, 18, 73] is another method for obtaining steady-state estimators

and their variances from a single simulation run.

We will use three measures to assess the quality of our estimator: the Bias (Section 2.7.1), the Variance (Section 2.7.2), and the Mean Square Error (Section 2.7.3). As shown in Section 2.7.3, bias and variance are the two aspects of MSE. Both the bias and the variance must vanish for the MSE to vanish. The following question arises from the need to investigate the initialization bias and variance.

- What starting initial condition produces the minimum MSE in a single server and a multi-server system? Is it the empty-and-idle state, the state closest to steady-state mean, the state close to steady state median or mode, or some other state?

The results of simulation studies can provide no or misleading insight if we disregard the random nature and the need for proper statistical analysis of the simulation output data. Another critical issue in the simulation studies of complex systems is the estimation of the length of a simulation run [1, 29, 45, 68] (which determines the effort required) to obtain the desired precision for the contemplated simulation estimators. The required length of a simulation run to obtain a desired statistical precision is estimated [1, 29, 45, 68] by computing the asymptotic variance and the asymptotic bias of the sample means. The ability to estimate the best simulation run length is a valuable information for maintaining a balance between information, cost and acceptable margin of error. Thus, the planning of a simulation experiment requires not only designing statistical methods to analyze the results, but it also requires the estimation of a simulation run length. Lock [45] proposed a run-length determination procedure based on the relative bias, the absolute bias, the variance and the MSE. It shows that the simulation run length is related to a specified precision which is further related to the variance and the bias. Therefore, based on the bias and the variance, a wide variety of stochastic systems are tested to address the following issues.

- Identifying which factors affect the convergence behaviour of the variance and the bias of an estimator, which further affects the simulation run length required to obtain estimators of a given precision.

- What systems are difficult and what systems are easy to simulate?

The length of a simulation run depends on (i) the required precision for the estimators, (ii) the variance of the time average which comprises the marginal variance and the covariance structure, and (iii) the bias induced by the choice of the initial state. The marginal variance is the variance at a given point in time. Both the bias and the variance depend on the covariance structure of the process. The covariance structure depends on the degree of periodicity and degree of decomposability. Here, periodicity indicates that the system will repeat, i.e., the system will visit a certain set of states periodically. Many systems are almost periodic in the sense that some states are visited at regular time intervals. This regular time interval determines the degree of periodicity of a system.

A decomposable system stays in a subset of states. An almost decomposable system tends to stay in a subset of states. The degree of decomposability determines the decomposability of a system. To address the issue of the factors affecting a simulation run length, it is our objective to explore the effect of the degree of decomposability and the degree of periodicity on the variance, the bias and the MSE of a time average. We will use computational methods to accomplish the following tasks, with the help of models of single server queueing systems, multi-server queueing systems and queueing network systems. In particular, we will address the following issues:

- Illustrate the transient behavior of the estimators as they converge toward their respective limiting values under a variety of starting initial conditions. The purpose is to observe different convergence patterns exhibited by a system under different initial conditions
- Analyze the covariance structure of a problem and show its relation with the variance
- Show a relationship between the variance and the bias
- Explore the effect of initial state on bias
- Investigate the impact of variability of the service-time distribution for single server systems

To address these issues, we use the following Markovian systems:

- An $M/M/1$ system
- An $M/E_k/1$ system
- An $M/M/c$ system
- A Sequential Queues system
- A Closed Queueing Network system
- An Inventory system

1.2 Traditional Background of Simulation

Digital simulation of a stochastic process requires a computer to imitate the operation of the process over time to estimate its performance. This section briefly describes various elements of a simulation system. A ***System*** [61, 3] refers to a collection of entities that interact with each other to accomplish one or more goals. In the context of simulation study, each significant *Object* or ***Entity*** [61, 3](e.g. a customer, a server etc.) of a system or process requires an explicit representation. An ***Attribute*** [61, 3] represents a property of an entity. An application of the simulation determines the interactions required among a collection of entities. For example, a branch of a bank with

tellers and customers (*entities*) can make up a system. The customer's account balance represents a customer's attribute. A time interval of a specified length is required to complete an ***Activity*** [61, 3], e.g., service time, interarrival time. Examples of activities are deposits or withdrawals of cash. The *random variables* that describe the state of a system are known as ***State variables*** [61]. The state of the system encompasses the knowledge required to obtain the future distributions of the system. For example, in a queueing system with two servers in sequence with each queue served by a separate server, the variables $X_i(t),\ i = 1, 2$, that represent the number of customers in queue i at time t, are the state variables. The sum of all the $X_i(t)$'s, which gives the total number of customers in the system at time t, is a ***derived variable***, which may represent the ***rewards*** at time t. Other derived variables are ***indicator variables*** [32], which may indicate the existence of a condition. For example, an indicator variable might assume a value 1 when the sum of all the $X_i(t)$'s is zero to indicate that the system is idle and empty, or a value 0 otherwise. The state variables, or variables derived from state variable(s), are called ***System variables***. Therefore, a variable representing the sum of all the $X_i(t)$'s is also a system variable. The state of a system changes instantaneously at the occurrence of an ***Event*** [61, 3], e.g., arrival or departure of a customer.

1.3 Discrete Event Simulation

The dynamic system where the state of a system changes at a particular instants of time and the system evolves over time by the occurence of events are called *discrete event systems* [3]. Some common examples of discrete event systems include flexible manufacturing systems, traffic systems, transportation systems, construction systems, and many more (see [3]). A *Discrete Event System (DES)* is frequently used for modeling, simulating and analyzing queueing systems and inventory systems. In a discrete event simulation, the time advances in discrete steps, of random and variable lengths, to the next state change. The state variable(s) change only at discrete points in time. As nothing happens between two consecutive events, rather than tracking the detailed system dynamics, the discrete event simulation passes over those time intervals [61, 3, 43].

1.4 Representation of Discrete Event Systems

A number of representations have been proposed for carrying out a discrete event simulation. The most prevalent approaches are entity attribute event based [3, 61], and state variable based [61, 54, 47].

1.4.1 Entity Attribute Event Systems

An entity attribute event system (EAES) contains a number of *entities*. Each entity may have certain *attributes* that are set at different points of time during simulation. For example, an attribute *Atime* of a customer entity records the arrival time of the customer at current server, and an attribute *Stime* records the duration of service at the current server. Each entity has a data structure to store its attributes. In addition to elements discussed in Section 1.2, an EAES simulation requires a *Clock*, a variable to represent simulation time. A data structure called the *Event record* is maintained for each event. It contains the information for an event occurring at current time or some future time along with the data needed to execute the event. At the minimum, the event record will contain the event type (arrival, departure etc.) and time of occurrence of the event. A list of current and future events, ordered by time of occurence, is maintained in an *Event List* or *Event Queue*. A *Duration* or *Delay* is an aspect of EAES simulation that is an unspecified length of time which is not known until it ends, e.g., delay of a customer in LIFO queue depends on future arrivals. In EAES, all the entities and their attributes are represented by the state variables of a given system. A system at any point of time is described by its *State*. The state of a system is generally characterized by one or more state variables, e.g., number of customers (entities) waiting in a queue, number of customer (entities) in service, etc. State variables change only at discrete set of points in time. EAES simulation adopts an *event-scheduling approach* to simulation modeling in that a simulation advances in time by executing the events in increasing order of their time of occurrence. An event may schedule other events. It is important to note here that no simulation time passes during the execution of an event. The resource requirements in EAES simulation are very high because of the need to keep the information for each entity, an event record for each event, and an ordered event list.

1.4.2 State Variable Event Systems

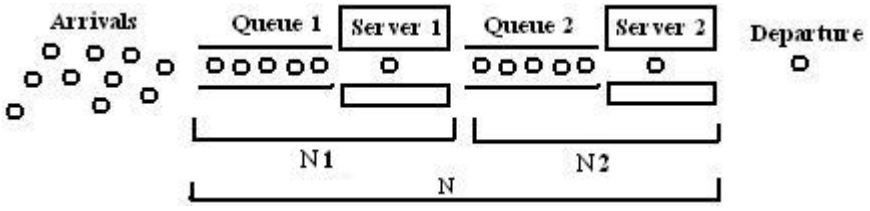


Figure 1.1: Two Servers in Sequence

A State Variable Event System (SVES) [26] requires less resources, and is simpler than an EAES, as an SVES does not contain records of entities. In an SVES, an aspect of the state of a model (e.g. number of customers waiting in queue) is represented by a state variable [61]. The basic model,

called the atomic model, is made up of state variables and actions or activities. The state variables hold the state of a model, while the actions are instrumental in changing the state of a model. However, scheduling is still an integral part of any SVES. Consider the SVES simulation for the system shown in Figure 1.1, i.e. two servers operating in sequence (without blocking) one after the other. Customers arrive at Queue 1. After getting service from Server 1, a customer either joins Queue 2 for receiving service from server 2 or leaves the system if Queue 2 is full. After receiving the service from server 2, the customer leaves the system. Arrivals to the system are Poisson with a mean λ and the service time at each server exponentially distributed with a mean $1/\mu$. Let variable $X1$ represent the number of customers in Queue 1 including the one being serviced by server 1. The number of customers in Queue 2 is represented by variable $X2$. The variable $B1$ is used to indicate the busy or idle status of server 1. $B1 = 0$ indicates that server 1 is idle, alternatively $B1 = 1$ indicates that server 1 is busy. Similarly, the variable $B2$ is used to indicate the busy or idle status of server 2. The variables $X1, X2, B1, B2$ collectively define the *state* of the system. The variables $N1, N2$ and N denote the maximum number of customer(s) permissible in Queue 1, in Queue 2 and in the system. Events in this system include arrival at Queue 1 denoted by **A1**, start service at Server 1 denoted by **S1**, finish service at Server 1 denoted by **F1**, arrival at Queue 2 denoted by **A2**, start service at Server 2 denoted by **S2** and finish service at Server 2 denoted by **F2**. The variable *now* represents current simulation time. Two special events *Start Simulation* denoted by **SS** and *End Simulation* denoted by **ES** are added to the event queue to ensure the begining and end of the simulation run. Event **ES** in line 1 is scheduled to occur at time T. Here T denotes the length of a simulation run, or the simulation time when the simulation terminates. Table 1.1 describes the simulation. At the start of the simulation (i.e. event SS on Line 1) the variables X1,X2,B1 and B2 are set to 0 signifying that both the queues are empty and both the servers are idle. As a result, the simulation of sytem begins in 'empty-and-idle' state. An arrival (event A1 on line 2) is scheduled to occur at current time. Arrivals keep happening even if the system is full. However, as shown on line 3, arrivals to the system are lost when Queue 1 is full (i.e., $X1 = N1$) and there is no change in the system variables. In addition, this arrival event will schedule next occurence of event A1 after a duration of Atime from current simulation time (i.e., now). In contrast, as shown on line 4, an arrival into the system when there is room to accomodate an arriving customer (i.e., $X1 < N1$) increases the value of X1 by 1 and schedules next occurence of event A1 after a duration of Atime from current simulation time. In addition, it schedules event S1 on line 5 at current simulation time because their is a customer in Queue 1 and server 1 is ready to serve i.e., $X1 > 0, B1 = 0$. When the event S1 occurs (line 6), it increases B1 by 1, thus indicating that server 1 becomes busy. It also schedules event F1 after a duration of Stime from current simulation time. The event F1 on line 7 occurs only when server 1 is busy i.e., $B1 > 0$. Event F1 on line 7 decreases X1 and B1 by 1 thus indicating that server 1 becomes

Table 1.1: State Variable Event System Simulation of Two Servers in Sequence

Line	Current Event	Event Condition	Schedule Event	Schedule At Time	Scheduling Condition	Current Event Changes			
						X1	X2	B1	B2
1	SS		ES	T		=0	=0	=0	=0
2			A1	now					
3	A1	$X1 = N1$	A1	now + Atime					
4	A1	$X1 < N1$	A1	now + Atime		+1			
5			S1	now	$X1 > 0, B1 = 0$				
6	S1		F1	now + Stime				+1	
7	F1	$B1 > 0$	A2	now	$X1 > 0, X2 < N2$	-1		-1	
8			S1	now	$X1 > 0, B1 = 0$				
9	A2		S2	now	$X2 > 0, B2 = 0$		+1		
10	S2		F2	now + Stime					+1
11	F2	$B2 > 0$	S2	now	$X2 > 0, B2 = 0$		-1		-1
12	ES	$now \geq T$							

idle and there is room for one more customer in Queue 1. It can also schedule next occurence of event S1 (see line 8) if there is a customer available in Queue 1 to be serviced and server 1 is idle i.e., $X1 > 0, B1 = 0$. On arrival at Queue 2 (event A2 on line 9), the variable X2 is increased by 1 and event S2 is scheduled at current simulation time because there is a customer in Queue 2 and server 2 is ready to serve i.e., $X2 > 0, B2 = 0$. The event S2 on line 10 increase B2 by 1 thus indicating that server 2 becomes busy. It also schedules event F2 after a duration of Stime from current simulation time. *It is important to note here that all the event records in the event list are arranged in the increasing order of their time of occurence.* When event F2 on line 11 occurs under condition $B2 > 0$, it decreases X2 and B2 by 1 thus indicating that there is room for one customer more in Queue 2 and server 2 is ready to serve another customer. It can also schedule event S2 at current simulation time if there is a customer in Queue 2 and server 2 is idle. The simulation proceeds in this fashion to the termination time T of the simulation. Data of interest is saved along the way for statistical analysis after the occurence of event ES.

1.4.3 Markovian Event Systems

Grassmann [23] described a ***Markovian event system*** *(MES)* principally as an event driven system where events happen at certain rates, with the rates depending only on the present state.

A MES can be described by a number of discrete state variables that change only when an event occurs. In this thesis, we are looking at Markovian event system(s) because they are tractable and can imitate Discrete Event System(s) very closely. For exponential distributions in time, schedules in SVES Table 1.1 are replaced with rates in MES. In a MES, events occur at random with rates that depend only on the present state of the system and not on its history, thus making event scheduling redundant in MES. The events that affect the state of a system can be arrivals, departures, reneging, change of phase etc. These events are similar to events in SVES and EAES. Unlike SVES and EAES, which can only be solved by simulation, MES can be solved numerically. Furthermore, prior to a simulation experiment of an MES, some insight into the variation of the results from simulation to simulation can be obtained by numerically solving the MES, and obtaining the estimates of the run length, variance and bias.

A Markovian event system can be viewed as a table of events indicating their effect on the system, their rates of occurrence and the preconditions regulating their occurrence. For example, consider an $M/M/1$ system (Table 1.2) with a single state variable X and waiting room capacity of 4 (including the one being served). In this system, value of state variable X represents the number of entities in the system, i.e., $X = i$ denotes that the system is in state i, or i entities are in the system. An arrival at rate λ will be allowed in system when there are less than 4 entities (i.e., $X < 4$) in the system, and it will increase the state variable X by 1. An entity is blocked from entering the system if there are already 4 entities in the system. A departure at rate μ can only happen if there are 1 or more entities (i.e., $X > 0$) in the system. A departure decreases state variable X by 1, as shown in Table 1.2. The Table 1.3 gives a MES event table similar to the SVES

Table 1.2: Table for Markovian Event System Simulation of a M/M/1 System

Event	**X**	**Rate**	**Condition**
Arrival	+1	λ	$X < 4$
Departure	-1	μ	$X > 0$

simulation Table 1.1 for two servers in sequence as shown in Figure 1.1. In this case, variable $X1$ represent the number of customer(s) in Queue 1, $X2$ denotes number of customer(s) in Queue 2. The variables $N1$ and $N2$ denote the capacity of queue 1 and queue 2 respectively. Arrivals to the system are Poisson with rate λ. Service times at server 1 and server 2 are exponentially distributed with mean $1/\mu_1$ and $1/\mu_2$, respectively. Arrivals to the system increase the value of variable $X1$ by 1. The event consisting of the completion of the service at server 1, departure from server 1 and arrival at server 2 occurs. This event, denoted by *1To2*, occurs at rate of μ_1. The event *1To2* occurs under condition $X1 > 0$ *and* $X2 < N2$ denoting that there is an entity receiving service at server 1, and there is room to accomodate a customer at server 2. Event *1To2* decreases state

Table 1.3: Table for Markovian Event System Simulation of Two Servers in Sequence

Event	**X1**	**X2**	**Rate**	**Condition**
Arrival	+1		λ	$X1 < N1$
Departure1	-1		μ_1	$X1 > 0, X2 = N2$
1To2	-1	+1	μ_1	$X1 > 0, X2 < N2$
Departure2		-1	μ_2	$X2 > 0$

variable $X1$ by 1 and increases state variable $X2$ by 1. If a customer serviced at server 1 cannot find a room in Queue 2, the customer leaves the system at the occurence of event *Departure1*. This event also occurs at rate of μ_1, but under the condition $X1 > 0, X2 = N2$. This event decreases $X1$ by 1. A customer leaving the system after being serviced at server 2 decreases the value of state variable $X2$ by 1, at the rate of μ_2 at the occurence of a *Departure2* event under condition $X2 > 0$, i.e., if there is at least one customer in the second line. It is evident that the timing, scheduling and sorting mechanism used in SVES simulation is not required in MES simulation, thus making MES system simulation efficient as compared to SVES system simulation. Moreover, the efficiency using numerical solutions of MES to estimate the quantities of interest makes MES a useful alternative to SVES simulations. Hence, we will use the MES in our thesis.

The run length for a simulation of a Markovian event system is on the one hand influenced by factors like budget and time constraints, on the other hand it is influenced by the properties of the model. These properties include the degree of decomposability, periodicity, initial state, number of state variables, etc. Markovian event systems allow us to numerically determine the variance, the bias and the MSE that are useful for determining the run length needed for our purpose. This helps us to consider and examine the influence of these factors on the simulation run length when simulating a Markov process.

1.5 Review of Queueing Theory Fundamentals

Waiting lines or *queues*, whether visible or not, are a regular feature of our everyday life. Therefore, many simulation studies involve queues. The mathematical study of waiting lines is commonly known as *queueing theory*. Queueing theory is used to describe these real world queues, and also more abstract queues such as processes waiting in operating systems. An analytical model constructed to study a queueing system may not precisely correspond to the real situation, but the model can provide some insight for understanding the queueing system.

1.5.1 Characteristics of Queueing Systems

Queueing systems arise when there are customers requiring service. The word 'customer' is used in a generic sense and is used interchangeably with the terms like message, request, job, process, packet etc. depending on the application. A queueing model is characterized by the arrival process of customers, service capacity, customer population, waiting room capacity, service times, service discipline and behavior of the customers waiting in queue. The customers arrive from a *calling population* which can have finite or infinite capacity. Queueing systems where the customers arrive from outside the system are generally called *open queueing system*. In contrast to open queueing systems, in *closed queueing systems* there are a fixed number of customers in the system and no customer arrives from outside the system.

Customers exhibit different *queueing behaviors* when waiting in a queue or on arrival at a queue. Like calling population capacity, the waiting room can also have finite or infinite capacity. *Waiting room capacity* or *System capacity* influences the behavior of a customer inside a queue or on arrival at a queue. On arrival a customer may join a queue or *balk* (i.e., leave on arrival) on seeing a long waiting line. A customer waiting in a queue for service may continue to wait or *renege* (i.e., leave after some period of waiting) due to the prolonged wait.

In queueing models, the service time distribution typically characterizes the *service pattern*. The *service rate* concerns the average number of customers completing service per time unit. In a multiserver environment with c servers, the service rate of each busy server (denoted by μ) is assumed to be independent of the number of customers (n) in the system and is constant, provided that at least one customer is in the system. It is true for $c \geq 1$. For c servers, if $1 \leq n < c$ the service rate is $n\mu$, and if $n \geq c$ then service rate will be $c\mu$.

In the following section, we introduce a shorthand notation that is used to characterize a range of these queueing models having a single queue. More than one queue in a system necessitates the arrangement of queues in *sequential* or *parallel* order to one another, or a combination of *sequential* and *parallel* queues thus forming a *queueing network*. We are concerned with systems without a queue, with a single queue, with queues in series and with queueing networks.

1.5.2 Queueing Notation

For classifying queueing systems, one typically uses Kendall's popular notation, given as

$$A/B/C/D/E$$

In this notation, A represents customers interarrival time distribution, B represents service time distribution, C represents the number of servers in a system, D represents the maximum number of customers that a system can accommodate and E represents the size of the calling population. Some other notations and symbols used in our thesis are λ for the arrival rate, μ for the service

rate and ρ for the offered load or traffic intensity (calculated as λ/μ).

Some of the commonly used distributions for A and B include M (symbolizing Markovian or Exponential distribution), D (symbolizing Degenerate or Deterministic distribution), E_k (symbolizing Erlang distribution with parameter k), H_k (symbolizing Hyperexponential distribution with k phases), G (symbolizing General or Arbitrary distribution) and GI (symbolizing General distribution with independent interarrival times). Moreover, the complexity of high speed networks has created significant interest in a traffic arrival process where consecutive arrivals are correlated. The Markov-modulated Poisson process is one such non-GI arrival process. In our thesis, we assume that the calling population is infinite. So to describe the characteristics of a queueing system, we use the notation $A/B/C/D$. In some special cases we digress from this convention.

1.6 Outline of the Thesis

In Chapter 2, we give an elementary review of stochastic processes that are commonly studied using simulation methodology and their classification. Since the transition probabilities associated with the states of a system play an important role in the study of Markov chains, the classification of the states of a system is described. Chapter 2 describes some commonly used models for modeling arrivals to the system, such as the random arrival processes. A brief discussion of the time proportion and the time average as an estimator is given. Three performance measure namely the Bias, the Variance, and the Mean Square Error that assess the goodness of our estimator are described. We also describe the analytical and numerical approach for solving simulation systems in transient and equilibrium conditions. The experimental models selected are explained along with the reasons and possible implications for choosing them. Various factors and behaviors that effect the length of a simulation run are also discussed.

The estimation of the desired characteristics of a system in simulation requires numerical evaluation of the model using the data collected with the help of a computer. In Chapter 3, the computational aspects of the simulation are introduced. It surveys current research studies on the statistical analysis of simulation data. We briefly outline the statistical approach used in analyzing a simulation model. The use of probability and statistics is an integral part of a simulation study. We review some basic probability and statistics particularly relevant to simulation. Typically, the point estimate defined in Section 3.1.1 is used to characterize the system analyzed [16, 14] (or measure the performance of a system). The measure of system performance to be estimated is often the expected number in the system. The confidence interval, as defined in Section 3.1.2, determines the accuracy of the obtained characteristics. We discuss the classical statistical approach to interval estimation of independent and identically distributed observations in Section 3.1.2. The common problems and strategies in the statistical analysis of simulation output data are discussed. A rela-

tionship between variance, covariance and bias is established. We discuss the expected behaviour of the measures of performance for the systems selected for experimentation. Chapter 4 describes underlying algorithms for computing the transient and steady-state measures of interest and verifies the accuracy of the results obtained.

In Chapter 5, empirical results from our experiments are interpreted in detail. Some possible convergence patterns for the measures of performance are discussed. The experimental results are explained. These results give us a better understanding on how to set up simulation experiments in general. We conclude this research study in the final chapter, Chapter 6, by elaborating on the numerical and experimental findings in our experiments. In general, we discuss the possible implications on the experimental designs. Finally, some suggestions about possible future research studies of this topic are given.

Chapter 2

Simulation and Stochastic Processes

Grassmann [33] defined a process as stochastic if it can behave in different ways, and if one can associate a probability with each possible behavior. The stochastic systems considered here are composed of many stochastic processes, with one for each state variable. The state variables jointly define a state. If the number of states is finite, each (global) state can be given a number, and this number is a process by itself. Hence, a set of processes $\{X_i(t), t \in [0, \infty]\}$ can be thought of either as a multi-dimensional process, or, if finite, of a single process represented by the state number. The process being observed can be an evolutionary process or a stationary process progressing in time following certain probabilistic laws. The process is said to be *stationary process* if the distribution of $X_i(t)$ does not change with change with t, otherwise the process is known as *evolving process*. Stationarity is an important property considered in this thesis.

2.1 Classification of Stochastic Processes

A stochastic process is a collection $\{X(t_n)|t_n \in T\}$ of random variables $\{X(t_n)\}$ where T is the index set of the process. A stochastic process with discrete parameter $\{X(t_n),\ n = 0,\ 1,\ 2,\ 3,\ \cdots\}$ or continuous parameter $\{X(t_n),\ t_n \geq 0\}$ is called a *Markov process* if, for any finite subset of time points $t_i \in T,\ (i = 0,\ 1,\ \cdots,\ n)$, where $t_0 < t_1 < \cdots < t_n$, the conditional distribution of $X(t_n)$ given the values of $X(t_{n-1}),\ X(t_{n-2}),\ \cdots,\ X(t_1)$, requires only $X(t_{n-1})$, i.e., the most recent value of the process. In this way, a Markov process possesses a memory-less [30] or *Markovian* property. At any given time t_n, the possible values of $X(t_n)$ are called the *states* of the process at t_n. The states of the system are mutually exclusive. The set of all states (for all t_n) of a stochastic process is called its *state space*. The naming convention states that a process with discrete state space is called a *Markov chain*, otherwise it is known as a *Markov Process*. The process is realized by a function called *sample function*, $x_\omega(t)$, and the values $x_1,\ x_2,\ x_3,\ \cdots$ assumed are called the *realizations* of the process. In a simulation, these realizations are called *replications* or *replicas*.

2.1.1 Discrete Time Markov Chain

Markov processes with a discrete state space are often called the *Markov chains*. Most often a set of integers $\{0,\ 1,\ 2,\ \cdots\}$ is used to represent the state space of a Markov chain. The sequence of random variables $\{X(t_n),\ t \in T\}$ for $t = t_0 < t_1 < \cdots < t_n$, assuming a finite value or countably infinite value, is an example of *Discrete Time Markov Chain* (DTMC), if the conditional probability distribution of $X(t_n)$ depends only on $X(t_{n-1})$. More precisely, given a discrete set of possible states, the process $\{X(t_n),\ t \in T\}$ is called Markovian if

$$\begin{aligned} Pr\{X(t_n) = x_n | X(t_{n-1}) = x_{n-1}, X(t_{n-2}) = x_{n-2}, \cdots, X(t_0) = x_0\} \\ = Pr\{X(t_n) = x_n | X(t_{n-1}) = x_{n-1}\} \end{aligned}$$

2.1.2 Continuous Time Markov Chain

In a queueing system, consider observing the number of persons waiting in the queue for service at any point of time. Such a process, where the time is continuous but the state space is discrete, can be characterized as a *Continuous Time Markov Chain* (CTMC). For example, $X(t)$ could denote the number of persons waiting in queue at time $t \geq 0$. Consequently $\{X(t),\ t \geq 0\}$ has discrete state space, i.e., for each $t \geq 0$, the possible values $X(t)$ can assume are integers $0,\ 1,\ 2,\ 3,\ \cdots$ represented by $x(t)$. In a CTMC, the time a process spends in a given state has an exponential distribution. The exponential distribution is a continuous distribution that has a memoryless property [30]. By observing the process at an equally-spaced discrete set of points in time, the process behaves like a DTMC. For the purpose of thesis, the models examined are assumed to be continuous in time and discrete in state space, i.e., CTMCs.

In the system represented by Table 1.2, on page 9 there is only one state variable. Consequently, value of this state variable will determine the state of the system, i.e., $X = 0,\ 1,\ 2,\ 3$ *or* 4 represents $0,\ 1,\ 2,\ 3$ *or* 4 entities in the system. From the given table (Table 1.2), a transition matrix can be created, showing the transition between states and their respective rates.

$$\mathbf{A} = \begin{bmatrix} -\lambda & \lambda & & & & \\ \mu & -(\lambda+\mu) & \lambda & & & \\ & \mu & -(\lambda+\mu) & \lambda & & \\ & & \mu & -(\lambda+\mu) & \lambda & \\ & & & \mu & -(\lambda+\mu) & \lambda \\ & & & & \mu & -\mu \end{bmatrix}$$

Each entry in the transition matrix is placed according to its effect on the system. In the transition matrix, each row represents an existing state of the system and each column represents the state to which system can potentially travel. The entry at the intersection of a row and column represents the rate at which the system will travel from its existing state, represented by the current row, to

its future state represented by the column. If there is no rate entry at the intersection of a row and column, then no transition is allowed. For a CTMC simulation, the diagonal entry in each row is set to be the negative of sum of all other rates of its row representing the total rate of leaving current state. Once the transition matrix is obtained, it is easy to obtain the equilibrium probabilities associated with various states of the system using

$$\pi A = 0 \tag{2.1}$$

where $[\pi]$ is a vector of equilibrium probabilities, and,

$$\sum_i \pi_i = 1 \tag{2.2}$$

Once the probabilities are obtained, other measures of interest, like expected number in the system, mean of time average, bias, variance of time average, run length, etc., can be computed easily.

Processes with several state variables can also be converted to Markov Chains: in this case, row i represents state i before the state changes, and column j the state after the change. In this way, every finite state MES can be converted to a Markov chain.

2.2 Classification of States of Markov Chains

To estimate the long run behavior of a Markov chain, it is necessary to investigate the classification of states of a Markov chain and their effect on simulation run length. We discuss this here for a DTMC. For a CTMC the results are similar. Consider a Markov chain $\{X_n, n = 0, 1, 2, 3, \cdots\}$. We denote one-step transition probabilities for a stationary Markov chain by $p_{i,j}$ where for each pair of states i and j,

$$p_{i,j} = Pr\{X(t_n) = j | X(t_{n-1}) = i\} \ for\ all\ n = 1,\ 2,\ 3,\ \cdots$$

Define $N_i(m)$ to be the number of visits to state i in the first m transitions. Given the Markov chain was initially in state j, we denote the conditional probability of ever visiting a state k by $f_{j,k}$:

$$f_{j,k} = P\{N_k(\infty) > 0 | X_0 = j\}$$

A state k is accessible from j if $f_{j,k} > 0$. If state j is accessible from state k, and state k is accessible from state j, then state j and k are said to communicate. Communicating states possess following three properties [38]:

(i) If states j and k communicate, and states k and m communicate, then states j and m also communicate.

(ii) A state communicates with itself, i.e., $f_{j,j} > 0$, and

(iii) If state j communicates with state k, then state k communicates with state j.

States that communicate with each other can be assembled together to form an *equivalence class*. A single state might form a class. A Markov chain with a single class having all the states communicating with each other is said to be *irreducible*. The state which upon entering once cannot be left for another state is known as *absorbing state*, e.g., state j is an absorbing state if and only if $p_{j,j} = 1$.

2.3 Recurrent and non-recurrent irreducible Markov Chains

A state is said to be *recurrent* if, after leaving this state, given enough time, the process will always return to this state over and over again. If the process continues endlessly, the recurrent state is visited infinitely often. In contrast to a recurrent states, there are *non-recurrent states or transient states*. Consider a state j which has a non-zero probability that the process will not return to state j. If the process continues unendingly, and if there is a positive probability that the process never returns to state j, then state j is known as *non-recurrent state or transient state*. A class of states can consist of either all recurrent states or all transient states. In addition, not all states in a finite-state Markov chain can be transient. All the states in a finite-state Markov chain are recurrent. As a result, in an irreducible finite-state Markov chain, all the states communicate. So recurrence is also the class property.

The period of a state is defined as a smallest integer n $(n > 1)$ with property $p_{j,j}(t) > 0$ for all $t = n,\ 2n,\ 3n,\ \cdots$. If $n > 1$ the chain is said to be *periodic* and if $n = 1$, it is said to be *aperiodic*. Periodicity can also be shown as a class property just like recurrence. A state is i said to be *ergodic* [38] if it is aperiodic and positive recurrent. Consequently, a Markov chain is said to be ergodic when all the states in it are ergodic.

2.4 Commonly Used Arrival and Service Processes

For simplicity, we concentrate on arrival processes. Many arrival processes have been developed for queueing analysis, including the *Random Arrival Processes*. These processes can also be used for service processes. Two commonly used non-correlated arrival processes are the *Poisson process* and the *Erlang process*.

2.4.1 Poisson Process

The analytical simplicity of a Poisson process makes it the most frequently used arrival stochastic process. The Poisson process has only one parameter, namely the arrival rate λ. In a Poisson process the number of events within a given time interval follows the Poisson distribution and

the time between the events is exponentially distributed. The burstiness of a source traffic, often measured as the ratio of its mean to its variance, is 1 in case of Poisson distribution, thus limiting the burstiness of a Poisson process. Unfortunately, network traffic often has a higher burstiness than the Poisson process.

2.4.2 Erlang Process

In contrast to the Poisson process, where the times between events are exponential, the times between events in an Erlang process follow an Erlang Distribution. The Erlang distribution, found by A.K. Erlang in 1912, was used to approximate the duration of telephone calls. The Erlang distribution has two parameters, namely k and λ. Here k is an integer representing the number of phases and λ is called the rate. The distribution of the sum of k independent and identically distributed random variables each having an exponential distribution is the Erlang distribution. The mean and variance of the Erlang distribution are given as

$$\begin{aligned} E(X) &= k/\lambda \\ Var(X) &= k/\lambda^2 = E^2(X)/k \end{aligned}$$

2.5 Data Collection

To find the expectation of a system's performance measure(s), one must identify, collect and evaluate the relevant sample data from simulation. The measures so obtained can provide information regarding performance, reliability and availability of the system. Alternatively, the purpose of an experiment may require comparison of means and variances of various alternatives, finding the effect of different variables on system performance, or finding optimal levels/values of a set of variables. Measures of system performance typically allow us to measure the effect of different values/levels of one or more changeable (and influential) factors (qualitative or quantitative) on the behavioral response of alternative systems under study. In this section we will discuss the most commonly employed estimators of measure of system performance known as *time proportion* and *time average*. These measures may be used to measure average number of customers in the system, average time spent in the system, average number of customers in queue, average time spent in queue, proportion of time a server is busy (server utilization) etc. Figure 2.1 shows the time average and time proportion (obtained using simulation) for a $M/M/1$ queue with $\rho = 2/3$ and $N = 86$, and their convergence toward the expected values obtained numerically.

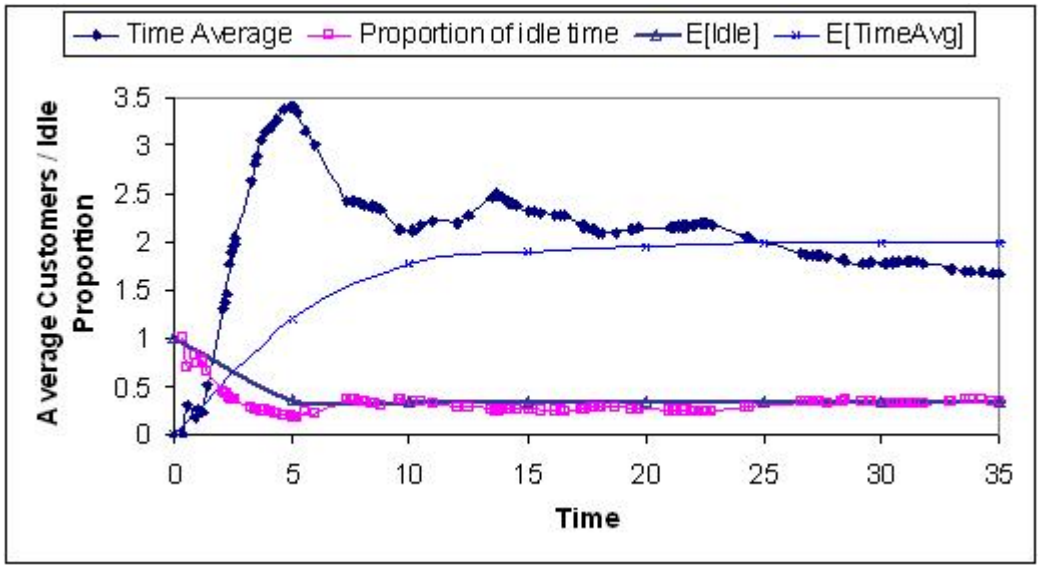


Figure 2.1: Time average and time proportion

2.5.1 Time Proportion

The time proportion estimates the proportion of time a specific state or condition exists in a system, e.g., proportion of time a server is busy or idle, proportion of time the system is empty, proportion of time more than three customers were in the system, etc.

Consider the problem of finding the proportion of a day that the server is in a specific state, e.g., idle. Since this query relates to an observation period and not to a particular point in time, the transient solutions are inadequate to provide the required information for such queries. Suppose in a simulation of a $M/M/1$ queue from time 0 to T, the system contains exactly i customers for a total time of t_i during the period $[0, T]$. The actual proportion of time that the server is idle can be easily calculated by measuring t_0, the length of time units the server is idle, and dividing it by the length of total observation period T. Similarly, t_i/T is the proportion of time that exactly i customers were in the system. In an observation period from 0 to T, if the proportion of time that the system is in state i is denoted by $r_i(T)$, then

$$r_i(T) = \frac{t_i}{T} \quad , \quad r_i(T) \rightarrow \pi_i \; as \; T \rightarrow \infty. \tag{2.3}$$

Consider the problem of finding the proportion of time related to a specific condition. An indicator variable is used to denote the existence of a condition. A value of 1 for indicator variable denotes satisfying the condition and a value of 0 vice versa. In an observation period from 0 to T, the proportion of time a specific condition, say g, is encountered is given as

$$r_g(T) = \frac{1}{T} \int_0^T X_g(t) \; \mathrm{d}t \; for \; 0 \; \leq t \; \leq \; T \tag{2.4}$$

where $X_g(t)$ assumes a value 1 if condition g is satisfied at time t and 0 otherwise. The expectation of time proportions can be calculated as

$$E[r_g(T)] = \frac{1}{T}\int_0^T E[X_g(t)]\ \mathrm{d}t \tag{2.5}$$

and variance can be calculated as

$$Var(r_g(T)) = E[(r_g(t) - E[r_g(T)])^2] = E[(r_g(t))^2] - E[r_g(t)]^2. \tag{2.6}$$

Time proportions can be easily interpreted by making use of procedures available for computing time averages (see Section 2.5.2). Hence, not many research studies particularly deal with time proportions.

2.5.2 Time Average

The time average of a measure of interest (e.g. number of customers in system) is used for estimating its expected value, i.e., $E[X]$. Consider simulation of an $M/M/1$ queue from time 0 to T where the system contains exactly i customer for a total time of t_i during the period $[0, T]$. It follows that $r_i(T) = t_i/T$ is the proportion of time that there are i customers in the system during simulation from time 0 to time T. In this case, the *time average* known as the *time-weighted average* is calculated as

$$\overline{X}(T) = \frac{(\sum_{i=0}^{\infty} it_i)}{T} = \sum_{i=0}^{\infty} i(\frac{t_i}{T}) = \sum_{i=0}^{\infty} i\ r_i(T). \tag{2.7}$$

Alternatively $\overline{X}(T)$ known as *time-integral average* of function $X(t)$ is represented as

$$\overline{X}(T) \quad = \quad \frac{1}{T}\int_0^T X(t)\ \mathrm{d}t. \tag{2.8}$$

The time average calculated by both (2.7) and (2.8) represent the same quantity. The mean and variance of $\overline{X}(T)$ is computed as

$$E[\overline{X}(T)] \quad = \quad \frac{1}{T}\int_0^T E[X(t)]\mathrm{d}t \tag{2.9}$$

and

$$Var(\overline{X}(T)) = E[(\overline{X}(T) - E[\overline{X}(T)])^2] = E[\overline{X}(T)^2] - E[\overline{X}(T)]^2 \tag{2.10}$$

and bias of the estimator $\overline{X}(T)$ is calculated as

$$B(\overline{X}(T)) = E[X] - E[\overline{X}(T)]. \tag{2.11}$$

In ergodic systems, $E[X] = \lim_{T\to\infty} E[\overline{X}(T)]$ and $\lim_{T\to\infty} B(\overline{X}(T)) = 0$. This asymptotic convergence of bias is important for studying simulation output. The calculation of $B(\overline{X}(T))$ requires

calculation of $E[\overline{X}(T)]$. There is extensive literature pertaining to the problem of finding means and variance of time averages, emphasizing their importance. Means and variances of time averages are used to estimate parameters and to construct confidence intervals for estimated parameter(s). Grassmann [27] demonstrated an efficient way to calculate the means and variances of time average in transient Markovian systems, which is used in this thesis.

2.6 Estimators and Estimator Errors

Consider a Markov process $\{X(t), 0 \leq t \leq T\}$. A statistic, such as $\overline{X}(T) = \int_0^T X(t)\mathrm{d}t/T$, computed from the simulation of a system from time 0 to time T is called an *estimator*. The value of the estimator is intended to estimate a parameter, e.g., the expectation of certain variable X (i.e. $E(X)$). The parameter of interest for a system can represent the average number of waiting jobs/customers in a queue, average cost of inventory or some other performance measure of interest. The following issues must be considered before accepting that an estimator truely estimates a parameter.

2.6.1 Unavoidability of Bias

A simulation run begins in a certain given state, e.g., $X(0) = 0$ or any other initial state. This initial state will influence $X(t)$, the value of X at a later time t. Consequently, the dependence of $\overline{X}(T)$ on $X(t),\ 0 \leq t \leq T$, results in the bias of the estimator.

2.6.2 Unavoidability of Variance

In a simulation of a system from time 0 to time T, the value of $X(t)$ varies randomly. Although $X(t)$ changes randomly through time, yet its expectation reaches an equilibrium. The estimator $\overline{X}(T)$ will also vary randomly resulting in an estimation error which can be determined by the variance of an estimator.

2.7 Quality of an Estimator

The quality of estimation is indicated by unbiasedness, minimum variance, and MSE. Considerable attention has been paid to the computation of bias and variance as the measure of quality of a simulation, as they determine the quality of results obtained from the simulation and they determine the amount of data required in order to achieve a certain confidence level.

2.7.1 Bias

The bias can be defined as the expected difference between a parameter and its estimator.

$$B(\overline{X}(T)) \quad = \quad E(X) - E(\overline{X}(T)) \tag{2.12}$$

Here, T is the length of the simulation run. An estimator is unbiased and it truly estimates the parameter if $B(\overline{X}(T)) = 0$. The bias converges to 0 as $T \to \infty$ and $T \times B(X(T))$ converges to a constant as $T \to \infty$.

Importance of Bias

The initialization bias is useful for finding the optimal *initial value* of a system, consequently for finding unbiased estimates. We will look into the effect of various factors on the bias including:

- How does the variability of the service-time distribution influence the initial bias? The service-time distributions selected for the study include the exponential and the Erlang-k distribution.
- How does the degree of decomposability affect the bias?
- How does the degree of periodicity affect the bias?

2.7.2 Variance

The distribution of $\overline{X}(T)$ approaches a normal distribution. To define a normal distribution, we need the mean and the variance. An important result given by Parzen [53] is that the sample mean is approximately equal to its expectation if the variance of the sample mean approaches zero as the length of the sample increases. We use central limit theorem for the statistical analysis of $\overline{X}(T)$ as $T \to \infty$. This theorem indicates that $(\overline{X}(T) - E(X))$ converges to a normal distribution with parameters $E(X) = 0$ and $Var(\overline{X}(T)) = \sigma^2_{\overline{X}}$. Note that $T \times Var(\overline{X}(T))$ typically converges to a constant value V as $T \to \infty$. We call $T \times Var(\overline{X}(T))$ the standardized variance, so the standardized variance converges to the limiting standardized variance, i.e., V. As a result, in a steady-state stochastic simulation of a system, the simulation run length to achieve desired precision is determined to a large extent by V. $T \times Var(\overline{X}(T))$, for a $M/M/1/N$ queue with $\lambda = 7, \mu = 10$ and $N = 6$ and with initial condition of empty-and-idle state ($I = 0$) is plotted in Figure 2.2 with reference to the limiting standardized variance. It shows that $\lim_{T\to\infty} T \times Var(\overline{X}(T))$ converges to a non-zero constant (i.e., the limiting standardized variance) and is important for planning and interpreting experiments. Also note that

$$Var(\overline{X}(T)) \quad = \quad E(\overline{X}(T)^2) - E(\overline{X}(T))^2. \tag{2.13}$$

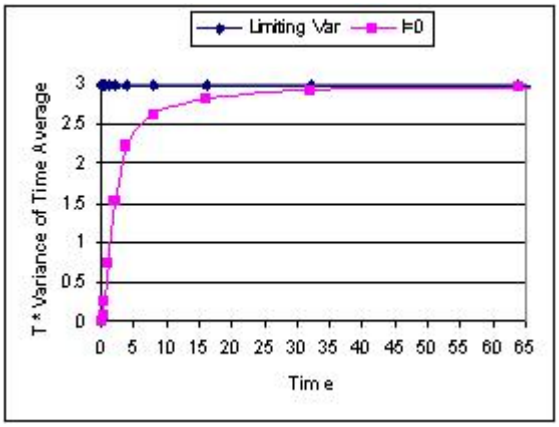


Figure 2.2: $T \times Var(\overline{X}(T))$ for $M/M/1$ queue with $\lambda = 7, \mu = 10$ and $N = 6$

Importance of the Variance

Estimators with small variance are preferred to obtain more precise and consistent results, for attributing more confidence to the conclusions. The asymptotic variance of an estimator measures the variability in the possible outcomes of a long simulation run. The asymptotic standardized variance of an estimator is useful in estimating the sampling period required for the simulation. Hence, we shall consider the variance of an estimator $Var[\overline{X}(T)]$ as a measure of goodness of an estimate.

2.7.3 Mean Square Error

The MSE of an estimator, $\overline{X}(T)$, measures the deviation and dispersion around the true value of the parameter by combining the effect of the bias and the variance as

$$MSE(\overline{X}(T)) = E(\overline{X}(T) - E(X))^2 = Var(\overline{X}(T)) + B^2(\overline{X}(T)). \tag{2.14}$$

It can be shown that as $T \to \infty, \lim_{T\to\infty} B(\overline{X}(T)) \approx 0$, therefore $\lim_{T\to\infty} MSE(\overline{X}(T)) \approx Var(\overline{X}(T))$ [26].

Importance of MSE

Obviously the simulator would like to know how close his estimator is to the true value, which makes the MSE a measure of prime importance. We will consider three cases of an estimator in Table 2.1 having (i) low $Var(\overline{X}(T))$ and high $B(\overline{X}(T))$ (see Line 1) (ii) high $Var(\overline{X}(T))$ and low $B(\overline{X}(T))$ (see Line 2) and (iii) low $Var(\overline{X}(T))$ and low $B(\overline{X}(T))$ (see Line 3). The calculation of $MSE(\overline{X}(T))$ for different values of $Var(\overline{X}(T))$ and $B(\overline{X}(T))$ in the Table 2.1 show that MSE is important for the following reasons:

Table 2.1: Table for $MSE(\overline{X}(T))$ of a System

Line	$Var(\overline{X}(T))$	$B(\overline{X}(T))$	$MSE(\overline{X}(T))$
1	0.1	0.8	0.74
2	0.5	0.1	0.51
3	0.1	0.1	0.11

- For the low values of the bias compared to the variance, the $MSE(\overline{X}(T))$ is also low (see Line 2 and Line 3), thus producing the estimates closer to $E(X)$.

- For comparatively low values of the bias (see Line 2 and Line 3), the $MSE(\overline{X}(T)) \approx Var(\overline{X}(T))$, giving reasonable confidence intervals (see Section 3.1.2). Alternatively, if there is no/negligible bias, then asymptotic MSE (or variance) can be used to estimate required simulation run-length to produce desired confidence interval and desired precision.

- If the bias is high compared to the variance, the bias dominates the MSE.

2.8 Transient and Steady-state Condition

For this research study, it is important to understand the concept of *transient* and *steady-state* conditions. In this section, we will investigate in detail the participation and the importance of probabilities to describe a system, and complement some of the results produced by Lock [45]. We will discuss the manner in which probabilities reflect a potential behavior of a system and their convergence toward equilibrium [20]. We define the conditional probability distribution of a stochastic process $X(t)$ observed in simulation at time $t \in T$, given the initial conditions I at the begining of the simulation at time 0 as

$$F_t(x|I) \quad = \quad P(X(t) \leq x|I). \tag{2.15}$$

The probability that an event $\{X(t) \leq x\}$ occuring given the initial condition I is defined as the conditional probability $P(X(t) \leq x|I)$. In a queueing context, the initial condition I may specify the number of customer(s) in the system and/or whether each teller is busy or not at time 0. Since stochastic systems or processes operate as a function of time, it is important to note that, as the process evolves in time, for each set of initial conditions I and for each value of t, the distribution $F_t(x|I)$ called the *transient distribution* will be different and the *transient conditions* will prevail for some time. After suffciently large time has elapsed, these distributions approach an equilibrium known as *Steady-state condition*. It can be shown for ergodic systems (see Section 2.12.1) that for

all x and for any initial conditions I

$$\lim_{t\to\infty} F_t(x|I) \quad = \quad F(X) \tag{2.16}$$

where $F(X)$ is the *steady-state distribution* of the output process $X(t)$, independent of I. In steady-state as well, the system moves from one state to another, i.e., the convergence of the system to a steady-state only means that the underlying probabilities converge. We will use a $M/M/1$ queueing system with Poisson arrival at a rate of $\lambda = 2$, exponentially distributed service time at a rate of $\mu = 3$, $\rho = 2/3$ and finite waiting space $N = 32$, to visualize the following concepts about transient probabilities and equilibrium probabilities from different perspectives.

1. Show how the probabilities express the potential behavior of a system in transient state and steady state. It is potentially useful for explaining the initialization bias of an estimator.

2. Show the random nature of output data in steady-state, which is the cause of estimation error of an estimator.

3. Show the notion of convergence and faster convergence of some distributions of system variables toward equilibrium than others. To some extent, this is useful for explaining why some systems simulations converge faster toward steady-state or equilibrium.

4. Show the dependence of state variables on initial conditions at different time points and their relationship to equilibrium or steady-state. This can potentially explain the initialization bias of an estimator.

Suppose that the number of customers in a system at any point of time t, including the one in service, is denoted by the state variable $X(t)$. To describe the system in a probabilistic way, first we need to determine the equilibrium distribution of X, the number of customers. Let X denote the number of customers in system when the system is in stochastic equilibrium, which has a truncated geometric distribution. That is to say, if π_i denotes the probability of being in state i when the system is at equilibrium, then for finite population [32]

$$\pi_i = P(X = i) = \frac{\left(\frac{\lambda}{\mu}\right)^i}{\sum_{j=0}^{N}\left(\frac{\lambda}{\mu}\right)^j} \quad i \leq N \tag{2.17}$$

and the probability that the number of customers go beyond a decisive value i in equilibrium is calculated using the formula:

$$P(X > i) = \frac{\sum_{j=i+1}^{N}\left(\frac{\lambda}{\mu}\right)^j}{\sum_{j=0}^{N}\left(\frac{\lambda}{\mu}\right)^j}. \tag{2.18}$$

The probabilities calculated above are equilibrium probabilities. Since a process evolves over time, equilibrium is reached only after the transient period has elapsed which is different for different

systems. We will investigate how fast $X(t)$ approaches this equilibrium. The algorithm given by Grassmann [27] is used to calculate the transient probabilities, $\pi_i(t)$'s, for this particular system of interest. From these $\pi_i(t)$'s, the expected queue length at time t, $Q(t)$, is obtained as

$$Q(t) = E[X(t)] = \sum_{i=0}^{\infty} i\pi_i(t). \tag{2.19}$$

The initial distribution of system is a degenerate distribution with $P\{X(0) = 0\} = 1$ as shown in

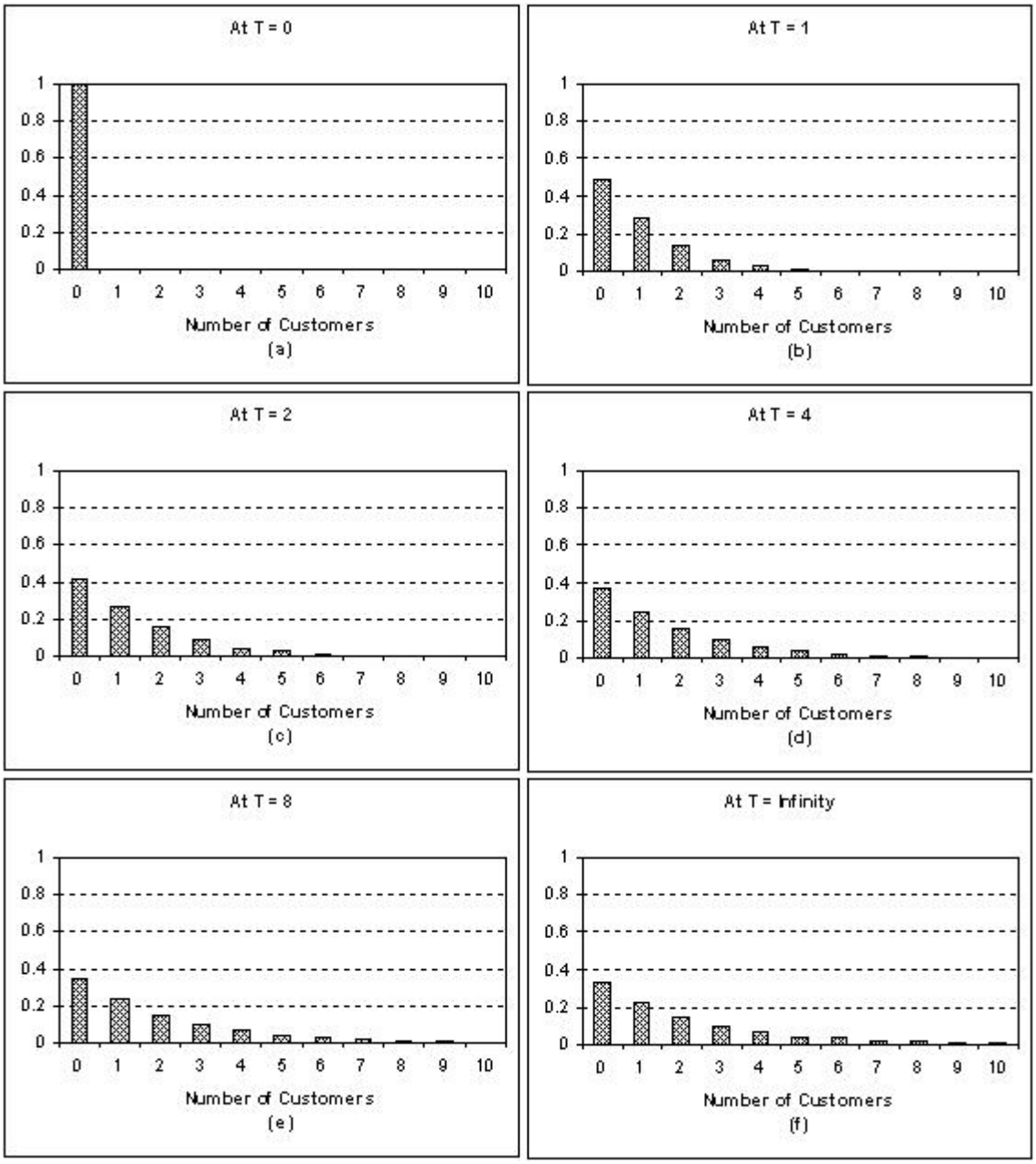


Figure 2.3: Convergence in Distribution for an $M/M/1$ queue

Figure 2.3(a), i.e., the simulation begins at time 0 with no customers in the system. Figure 2.3(a)

to (e) show the transient distributions for $t = 0, 1, 2, 4, 8$. After simulating for 1 time unit, at $t = 1$ (Figure 2.3(b)), the distribution becomes more evenly distributed. Then, slowly the distribution of $X(t)$ moves toward the steady-state distribution which is shown in 2.3(f). Figure 2.3(c) shows the distribution after simulating for 2 time units. Further convergence of system is depicted in Figures 2.3(d) and (e) as the system is simulated for 4 and 8 time units. After simulating long enough the distribution of $X(t)$ becomes invariant (see Figure 2.3(f)) representing the steady-state condition. It is important to note here that the value of $X(t)$ can still vary when the system is simulated after this point of time, as shown in Figure 2.4, causing the estimation error in an estimator. The time period before the steady-state condition has occured is called *transient period*, *burn-in phase* [29] or *warm-up period* [43, 8]. For a prescribed precision $\epsilon > 0$, the steady-state condition occurs at T when

$$E[X(T)] - E[X] < \epsilon. \tag{2.20}$$

We will now see the effect of different initial conditions on the transient behavior of a system.

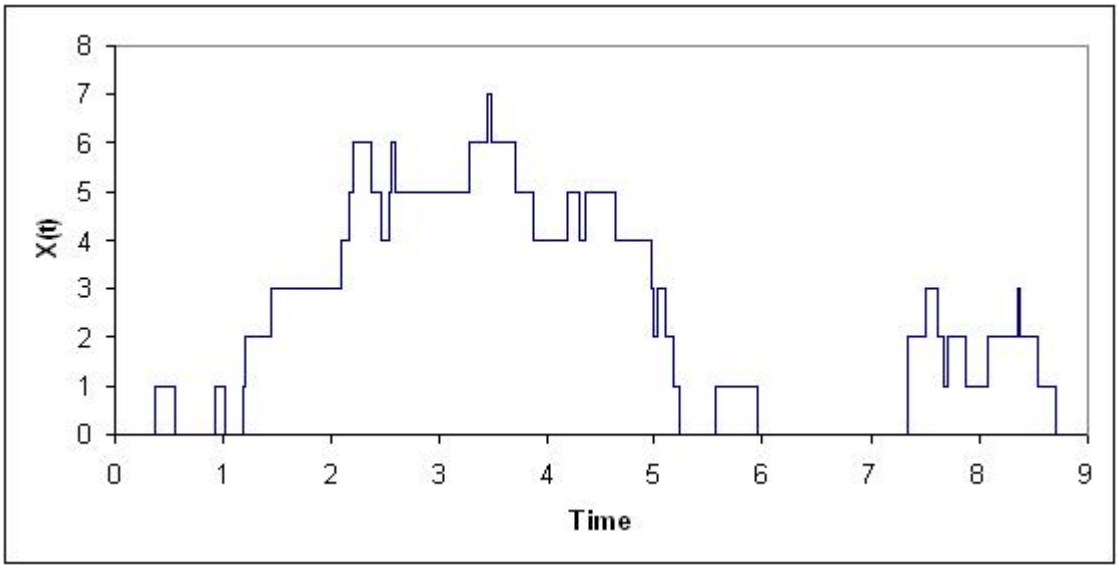


Figure 2.4: Random nature of simulation output

First we note here that, regardless of the initial condition, ergodic systems have an equilibrium distribution. The equilibrium distribution is independent of initial condition. The effect of the initial value $X(0)$ on the transient probabilities, $\pi_j(t)$'s, is investigated by first calculating $\pi_j(t)$'s given that the number of customers in the system at time zero is i. The $\pi_j(t)$'s so obtained represent $p_{ij}(t)$ as

$$\pi_j(t) = p_{ij}(t) = P(X(t) = j | X(0) = i) \tag{2.21}$$

Once $\pi_j(t)$'s are obtained, other measures of interest are easily calculated. The expected number of customers at time t given i customers at time 0 can be calculated as:

$$Q_i(t) = E[X(t)|X(0) = i] = \sum_{j=0}^{\infty} j p_{ij}(t) \tag{2.22}$$

The probability of having more than k customers in the system at time t, given $X(0) = i$ is obtained as:

$$P(X(t) > k|X(0) = i) = \sum_{j=k+1}^{\infty} p_{ij}(t) \tag{2.23}$$

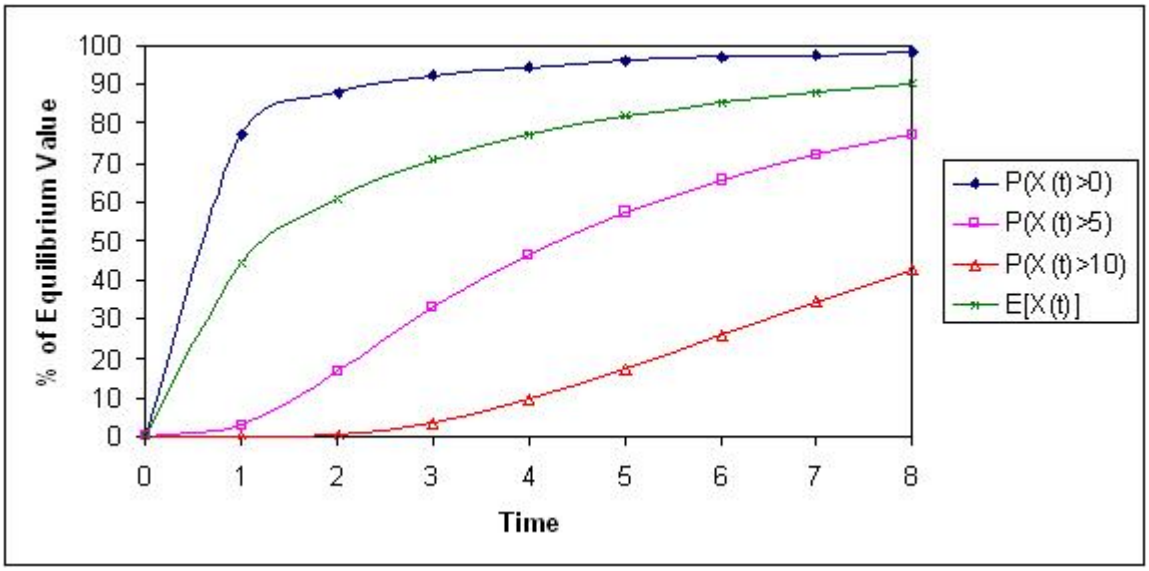


Figure 2.5: Convergence toward respective equilibrium

We would like to emphasize here that even for a simple system there is no general way to answer the question as to when equilibrium is reached. This depends very much on our choice of the estimator used to measure the performace of a system. To visualize this, we calculated $P(X(t) > 0), P(X(t) > 5), P(X(t) > 10)$ and $E[X(t)] = Q_0(t)$ for different values of t assuming no customers are present in system at time $t = 0$. The results are shown in Figure 2.5, where the results are made comparable by representing the measures expressed as a percent of their respective equilibrium values. Two extreme cases observed from Figure 2.5 are of $P(X(t) > 0)$ and of $P(X(t) > 10)$. The convergence of $Q_0(t)$ and $P(X(t) > 5)$ toward equilibrium is between these two extremes. Consequently, the probability $P(X(t) > 0)$ reaches equilibrium rather quickly signifying that the system is close to the equilibrium most of the time. In contrast, the probability of having over ten customers in system, $P(X(t) > 10)$, is below 50% of its equilibrium value, i.e., more time is needed to reach the equilibrium. The dissimilar shape of curves for different measures in Figure 2.5 can also be interpreted as the impact of certain initial conditions on convergence of different measures toward

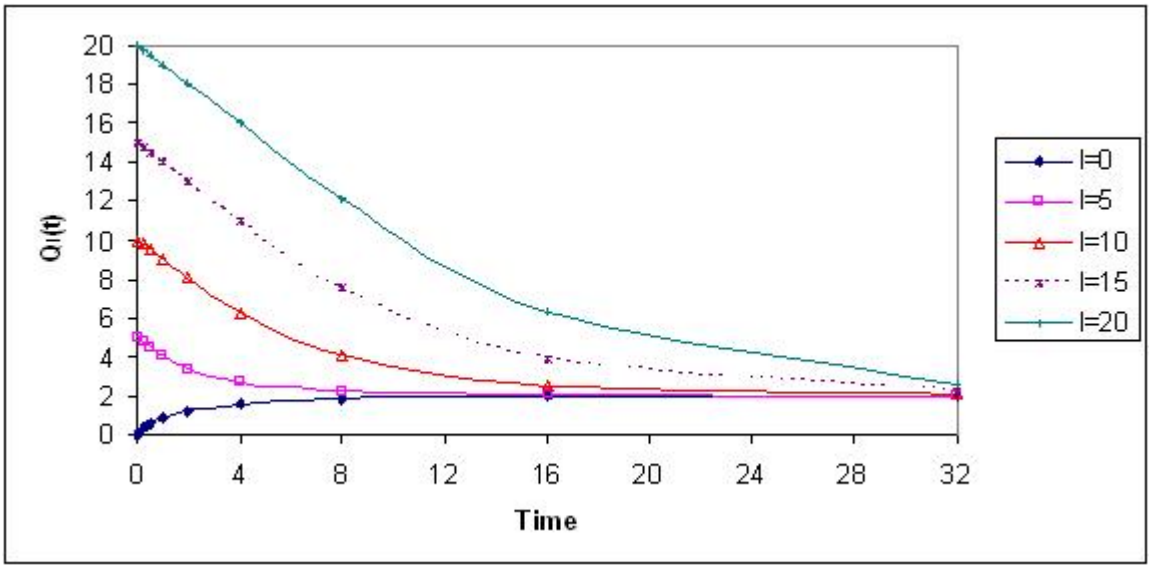


Figure 2.6: $Q_I(t)$ as a function of I for $M/M/1$ queue with $\rho = 2/3$

equilibrium, i.e., each measure has different transient period. Consequently, it shows the impact of the initialization bias introduced by beginning in a certain state on different measures.

To visualize the impact of initialization bias on the estimate obtained from an estimator, we plotted Figure 2.6 for different initial conditions. It shows that rate of the convergence of transient $Q_I(t)$'s for different initial conditions to equilibrium is different. The curve for the initial condition $I = 5$ shows the least bias by converging to equilibrium faster than the curves for other selected initial conditions. Recall that the ρ here is 2/3, therefore $E(X) \approx \frac{2/3}{1/3} = 2$. The curve for initial condition $I = 20$ shows the highest bias and converges slower to steady-state than the curves for all other selected initial conditions. Similarly, Figure 2.7 shows the probability that the system is idle at time t. One has different durations of transient periods introduced by different initial conditions, $X(0) = I$ where $I = 0, 1, 2, 3, 4$. The p_{I0} converges to π_0 faster when $X(0) = 3$ than when $X(0) = 0$, thus showing a shorter transient period. There is even more rapidly reducing effect of initialization bias for initial condition $X(0) = 2$. Furthermore, the effect of a certain initial condition, $X(0) = 4$, on different performance measures is depicted in Figure 2.8. It shows that different measures have different transient periods and converge toward equilibrium at different rates. It also shows that the effect of initialization bias is different for different measures.

To visualize the effect of initial condition from a different perspective, we plotted the Figure 2.9 (see page 32) showing $Q_I(t)$ for $t = 0, 1, 2, 4$ *and* 8 as a function of $X(0) = I$. The x axis of Figure 2.9 represents $X(0) = I$ and the Y axis represents the expected number of customers. The plotted $Q_I(t)$ values for $t = 0$ result in a straight line going through origin for the reason

$$E[X(0)|X(0) = I] = I = Q_I(0) \tag{2.24}$$

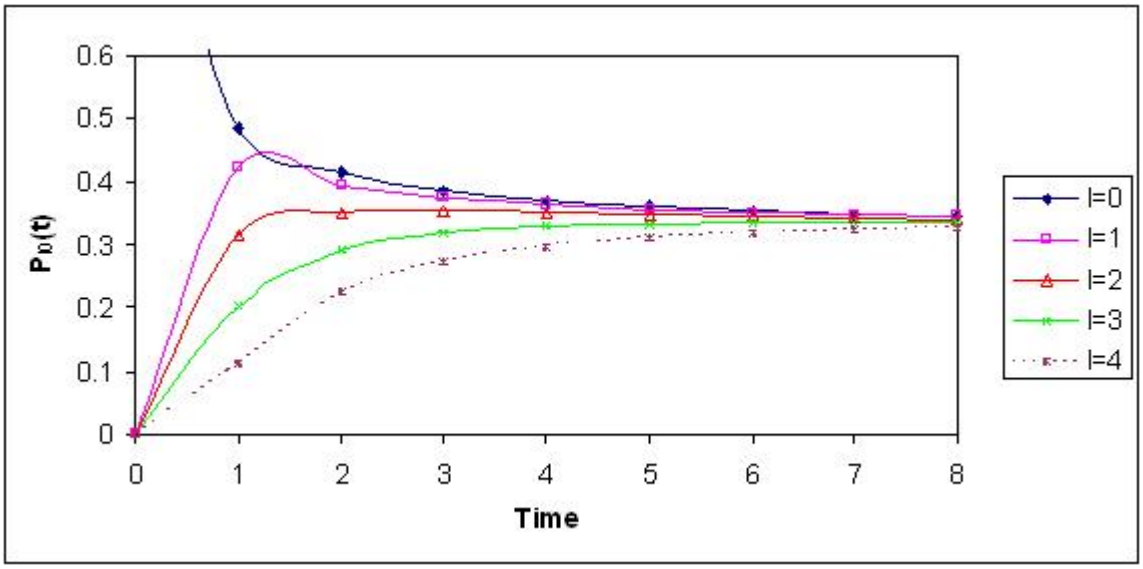


Figure 2.7: Effect of the initial conditions on the probability of being idle

The $Q_I(t)$ values plotted for $t = 8$ result in almost a horizontal line indicating that the effect of initial value I is negligible and the queue length is independent of I in the long run. The obvious behavior for other curves for t between 0 and 8 is an upward bend.

2.9 Behaviour of the Systems

In this section, we will describe different behaviours of a system that affect the length of a simulation run for obtaining the estimates with a prescribed accuracy in Markovian event systems.

2.9.1 Almost Decomposable Systems

A system, in principle, is an almost decomposable system when its components can be grouped in a manner such that the interactions between the subcomponents is much weaker than the interactions within each of the subcomponents. That is to say, by suitable reordering of the states, a stochastic matrix P is of the form $P = P^* + \epsilon C$ (see Table 2.2 (A)), where $\epsilon > 0$ is a small number defining the maximum degree of coupling between the subsystems in P. Also, the number of steps required and the likelihood of the steps to reach the states with high rewards can help in determining the decomposability of a system (see Table 2.2 (B)). Since a stochastic queue with ρ close to 1 tends to stay in higher states, it shows a high degree of decomposability in a straightforward manner and it will be used in this thesis. Similar problems are also found when studying a stochastic queueing network model. Complex systems consisting of subsystems (components and sub-components) are generally described by matrices that are likely to be large and sparse matrices. Courtois [9] extensively studied and reported applications of such Markov chains to computer systems and queueing

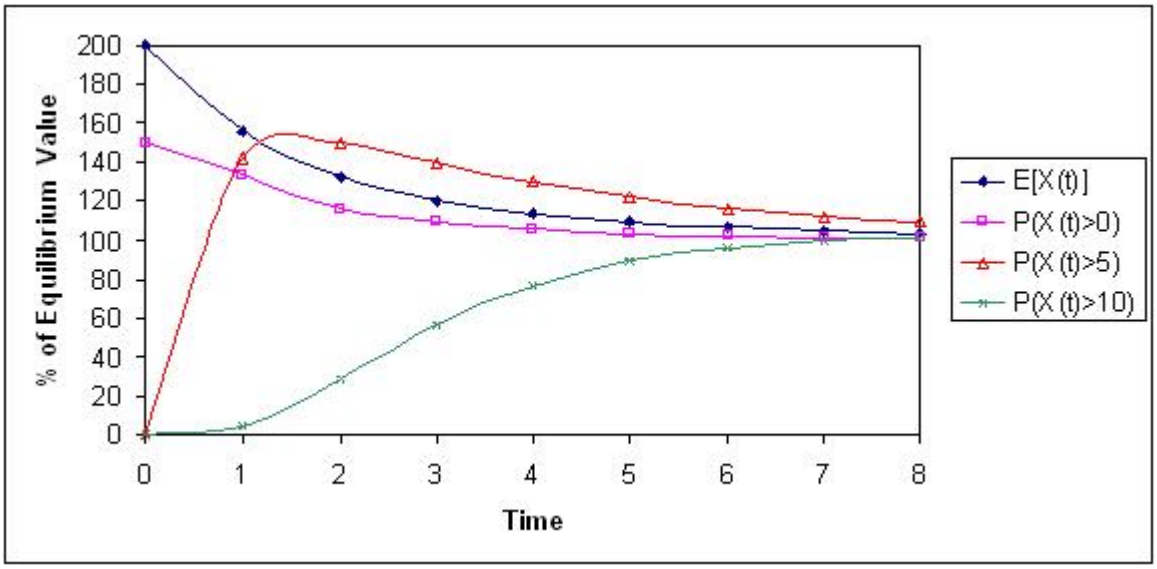


Figure 2.8: Speed of convergence of other measures for $X(0) = 4$

networks by regarding large systems as *Nearly Completely Decomposable (NCD) Markov chains*. A set of submatrices (representing a subsytem) along the main diagonal when superimposed, represents such a system [64]. The elements in the matrices of nearly completely decomposable systems, except the matrices along the main diagonal, converge to zero in limit. Such large scale compound systems show weak and slow interactions between the classes, and strong and fast interactions within a class. The subsystems of such a system can be studied separately to analyze the performance of a particular aspect of a system. However studying a subsystem in isolation does not give the information about its influence on the whole system or about its cooperation and interaction with other subsystems of a system. An example of such a system is memory hierarchies [9]. The intensity of the interactions between the classes will impact the length of a simulation run to obtain results with desired precision. We expect that the weaker the interactions between classes the longer a simulation must run.

2.9.2 Periodic Systems

Periodicity is another property of Markov chains that can effect simulation run length. A finite-state Markov chain with recurrent and aperiodic states is said to be *ergodic*. Periodic systems never reach a steady state as the periodicity in periodic systems never wears off. Periodic systems are not asymptotically independent as the state variable in the long run is not independent of present state. The present state affects a state in a far away future. In continuous time Markov chains, there are no periodic systems. However, there are systems that are almost periodic, for example inventory systems.

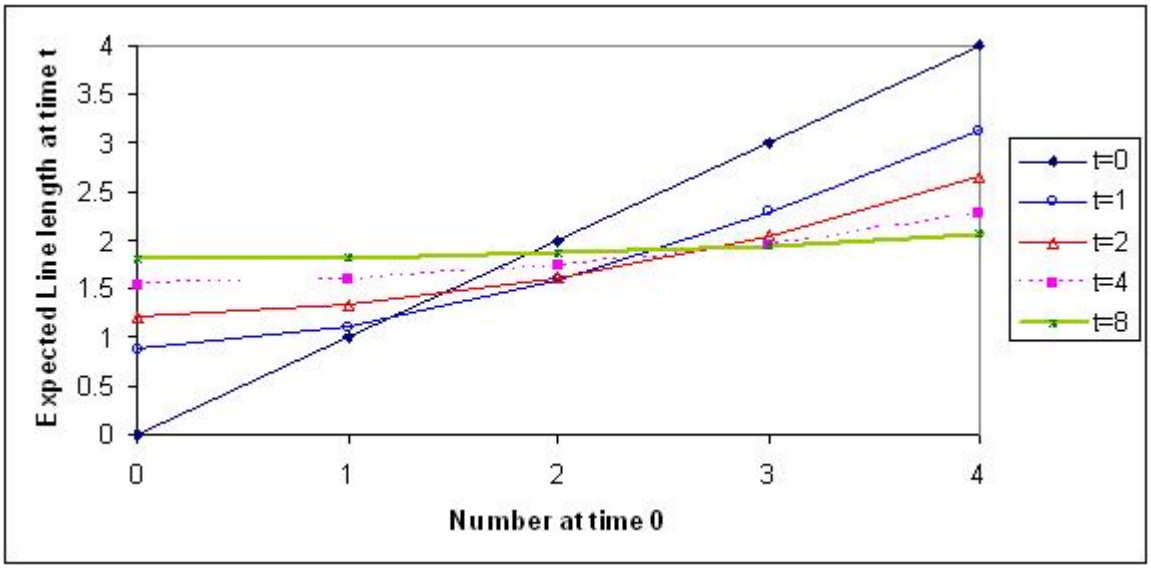


Figure 2.9: Dependence on the initial conditions

2.10 The Expected Rewards

The objective of a simulation is, usually, to find the expected rewards per time unit in equilibrium, given the reward in state i is r_i per time unit. A reward function $f_x(i)$ is used to define the reward in state i. Consider, for example, a transient or equilibrium Markov process $\{Y(t), t \geq 0\}$ with state space Ω. Here, $Y(t)$ represents the state of the process at time $t \geq 0$. Let r_i be the reward rate for state i. If $f_x(.)$ is a function of $Y(t)$ such that $f_x(Y(t)) = r_{Y(t)}$, then the reward rate at time $t \geq 0$ is $X(t) = f_x(Y(t)) = r_{Y(t)}$. Consequently, the reward accumulated from time 0 to T is $X(T) = \int_0^T X(t)\ \mathrm{d}t$, and the average reward per time unit is $\overline{X}(T) = \frac{1}{T}\int_0^T X(t)\ \mathrm{d}t$.

An important issue is the choice of the reward type. There are two types of reward structures. The first reward structure is single state and the second type is averaging. For single state reward structures, the reward for being in a given state is set to one, and for all other states were set to zero. On the other hand, for averaging of rewards, a particular system variable becomes the reward for a state. For example, in an $M/M/1$ queue the reward for being in state i is given as

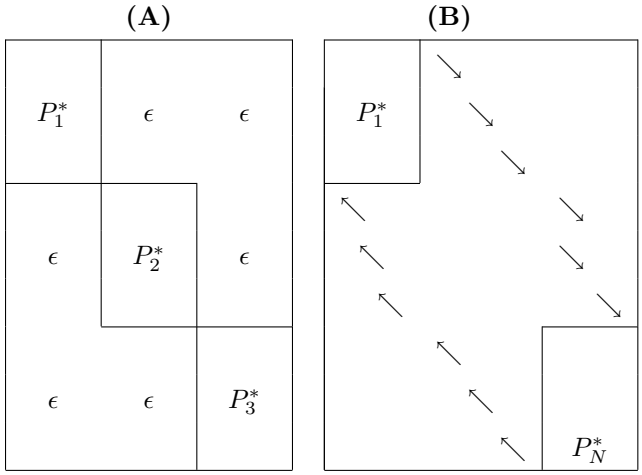
$$f_x(i) = r_i = i \qquad i \in S, \tag{2.25}$$

here i represents the number of customers in system and S is the set of all possible states. However, the reward for being in state i in a sequential queueing system with two queues is given as

$$f_x(i) = r_i = X1_i + X2_i \qquad i \in S, \tag{2.26}$$

where $X1_i$ and $X2_i$ denote the number of customers in queue 1 and queue 2 respectively in state i, and S is the set of all possible states. Similarly, in a closed queueing network with three queues where we are concerned only with the number of customers in queue 1 and queue 2 combined, the

Table 2.2: Almost decomposable Systems



reward for being in a state i is given as $f_x(i) = r_i = X1_i + X2_i$ for $i \in S$. We have looked into averaging rewards structure only. The expected rewards can be the expected number of customers in the system. A general formula for computing the expected reward per time unit in equilibrium for all queueing systems is

$$E(X) \quad = \quad \sum_{i \in S} \pi_i \; f_x(i) \tag{2.27}$$

This can be easily worked out once the steady-state probabilities π_i, $i \in S$ are obtained from equilibrium solutions.

2.11 Relation Between Bias and Variance

In this section, we will establish a close relation between the bias and the variance of a time average. This relation can potentially explain in general, the increase or decrease in the variance with the increase or decrease in the bias. Of course, the bias depends on the initial state i, which we indicate by using the symbol B_i. In fact, the variance can be obtained as the sum of products of the B_i with certain factors c_i. Details about the c_i are given in a paper of W. Grassmann [19]. A similar relation exists between the bias and the MSE. Therefore, in general we expect the variance to increase with the increase in bias and vice versa.

According to Grassmann [29] the variance can be factored into the variance of marginal distribution and the integral of the correlation coefficients, if the simulation time is long. If the sample mean of I random variables X_i in discrete case is

$$\overline{X} \quad = \quad \sum_{i=1}^{I} \frac{X_i}{I},$$

$$\text{then} \quad Var(\overline{X}) = \frac{1}{I^2}\left(\sum_{i=1}^{I} Var(X_i) + \sum_{i=1}^{I}\sum_{j=1_{j\neq i}}^{I} Cov(X_i, X_j)\right).$$

Since the central limit theorem holds for ergodic Markov chains, the $\overline{X}$ is asymptotically normal. For the covariance we have

$$\begin{aligned} Cov(X_n, X_{n+m}) &= \sum_i \sum_j (r_i - \mu_n) P(X(n) = i) P(X(n+m) = j | X(n) = i)(r_j - \mu_{n+m}) \\ &= \sum_i \sum_j (r_i - \mu_n)\pi_i P_{ij}^m (r_j - \mu_{n+m}) \end{aligned} \tag{2.28}$$

Note that for $m = 0$ this formula yields variance.

Covariance is the measure of the dependency of observations. The planning of experiments and statistical analysis of data from queueing systems requires consideration of the autocorrelation of the data [72, 57]. In this section, we will discuss the relation between variance and covariance of a process, and the stochastic convergence of a discrete process in time toward stationarity. For a sample of size I, we have the sample mean of I random variables X_i as

$$\overline{X} = \sum_{i=1}^{I} \frac{X_i}{I}$$

$$\text{therefore,} \quad Var(\overline{X}) = \frac{1}{I^2}\sum_{i=1}^{I}\sum_{j=1}^{I} Cov(X_i, X_j) \tag{2.29}$$

$$= \frac{1}{I^2}\left(\sum_{i=1}^{I} Var(X_i) + \sum_{i=1}^{I}\sum_{j=1_{j\neq i}}^{I} Cov(X_i, X_j)\right) \tag{2.30}$$

$$\begin{aligned} where,\ Cov(X_i, X_j) &= E[(X_i - E(X_i))(X_j - E(X_j))],\ (see\ (2.28))\ and \\ E(X_i) &= E(X_j) = \mu,\ \ therefore \\ Cov(X_i, X_j) &= E[(X_i - \mu)(X_j - \mu)]\ and \\ Cov(X_i, X_i) &= Var(X_i) = \sigma^2 \end{aligned}$$

also for a random variable X we have

$$Var\left(\frac{X}{I}\right) = \frac{1}{I^2} Var(X)$$

Therefore, from equation (2.29) we get

$$\begin{aligned} Var(X_1 + X_2 + \ldots + X_I) &= Var(X_1) + 2Cov(X_1, X_2) + 2Cov(X_1, X_3) + \ldots + 2Cov(X_1, X_I) \\ &+ Var(X_2) + 2Cov(X_2, X_3) + 2Cov(X_2, X_4) + \ldots + 2Cov(X_2, X_I) \\ &+ \ldots + Var(X_{I-1}) + 2Cov(X_{I-1}, X_I) + Var(X_I) \end{aligned}$$

Let ρ_{ij} be the correlation coefficient between X_i and X_j, that is

$$\rho_{ij} = \frac{Cov(X_i,X_j)}{\sqrt{(Var(X_i)Var(X_j))}} = \frac{Cov(X_i,X_j)}{\sigma^2}$$

or,

$$\rho_{ij}\sigma^2 = Cov(X_i, X_j)$$

we therefore get,

$$\begin{aligned} Var(X_1 + X_2 + \ldots + X_I) = \\ \sigma^2(1 \quad &+2\rho_{12} \quad +2\rho_{13} \quad +\ldots \quad +2\rho_{1(I-1)} \quad 2\rho_{1I} \\ &+1 \quad\quad +2\rho_{23} \quad +\ldots \quad +2\rho_{2(I-1)} \quad +2\rho_{2I} \\ &\qquad\qquad\qquad\quad \vdots \qquad\qquad \vdots \\ &\qquad\qquad\qquad\quad +1 \qquad\quad +2\rho_{(I-1)I} \\ &\qquad\qquad\qquad\qquad\quad +1). \end{aligned} \tag{2.31}$$

Table 2.3: Variance of Sum

	1	**2**	**3**	...	**I-1**	**I**
1	1	$2\rho_{12}$	$2\rho_{13}$	...	$2\rho_{1(I-1)}$	$2\rho_{1I}$
2		1	$2\rho_{23}$	...	$2\rho_{2(I-1)}$	$2\rho_{2I}$
⋮				⋮		⋮
I-1				...	1	$2\rho_{(I-1)I}$
I				...		1
Column Sum	S_1	S_2	S_3	...	S_{I-1}	S_I
Cumulative Sum	A_1	A_2	A_3	...	A_{I-1}	A_I

Equation 2.31 can be conveniently manipulated by arranging it in a tabular form as shown in Table 2.3. For instance, from Table 2.3, we can see that the column sums are $S_1 = 1, S_2 = 2\rho_{12} + 1, \ldots, S_I = 2\rho_{1I} + 2\rho_{2I} + \ldots + 2\rho_{(I-1)I} + 1$ and the cumulative sums are $A_1 = S_1, A_2 = S_1 + S_2, \ldots, A_I = S_1 + S_2 + \ldots + S_I$. In equilibrium, $Var(X_i) = Var(X_j) = \sigma^2$. Therefore in equilibrium $Var(X_1 + X_2 + \ldots + X_I)$ can be expressed as $A_I\sigma^2$, i.e., $A_I\sigma^2$ gives the value of $Var(X_1 + X_2 + \ldots + X_I)$, and hence

$$\begin{aligned} Var(\overline{X}_I) &= Var\left(\frac{X_1 + X_2 + \ldots + X_I}{I}\right) = \frac{1}{I^2}Var(X_1 + X_2 + \ldots + X_I), \text{ or} \\ Var(\overline{X}_I) &= \frac{A_I\sigma^2}{I^2} \end{aligned} \tag{2.32}$$

From equation (2.32), the convergence of $Var(\overline{X})$ or $Var(\overline{X}_I)$ toward zero can be decided on

the basis of convergence of $\frac{A_I}{I^2}$ toward zero, thus showing the dependence of variance on covariance/correlation. In non-decomposable Markov chains in equilibrium the correlation coefficient between state variables measured at $t = m$ and the same variables at $t = n$ for $n > m$ is a function of $n - m = h$ (i.e. depends only on the difference between n and m) and is denoted by $\rho(h)$. Thus, the correlation coefficient between state variables at mh and nh for state variable observed at $t = h, 2h, 3h, \ldots$ is denoted as

$$\rho_{mn} \quad = \quad \rho(nh - mh)$$

For working with Table 2.3, we need ρ_{mn} for $n > m$ where

$$\begin{aligned} \rho_{12} = \rho_{23} = \rho_{34} = \ldots = \rho(h) \\ \rho_{13} = \rho_{24} = \rho_{35} = \ldots = \rho(2h) \\ \vdots \\ \rho_{1I} = \rho((I-1)h) \end{aligned}$$

By using these values of ρ_{mn}, the S_I's and A_I's can be easily calculated. In general case

$$\begin{aligned} S_I &= 1 + 2[\rho_{(I-1)I} + \rho_{(I-2)I} + \ldots + \rho_{1I}] \\ &= 1 + 2[\rho(h) + \rho(2h) + \ldots + \rho((I-2)h) + \rho((I-1)h)]. \end{aligned}$$

If for $\epsilon > 0$, for non-decomposable Markov chains $\rho(h)$ converges to zero and S_I converges to a certain value r. Hence, there is an m such that for $I > m$ we have $S_I = r \pm \epsilon$,

$$Var(\overline{X}_I) = \frac{\sigma^2[A_m + (I-m)(r \pm \epsilon)]}{I^2} = \sigma^2 \left[\frac{A_m - m(r \pm \epsilon)}{I^2} + \frac{I(r \pm \epsilon)}{I^2} \right] \cong \frac{\sigma^2 r}{I} \qquad (2.33)$$

Consider now general processes, possible non-markovian ones. Mathematically, the *short range dependence* of a process is expressed by the exponential decrease of its autocorrelation function [12] ,i.e., $\rho(nh) \sim a^{|nh|}$, as $|nh| \to \infty$, $0 < a < 1$. Here, $\sim$ denotes that, in the long run the expressions on the two sides are proportional to each other. In contrast, the autocorrelation function of *long-range dependent* processes decay hyperbolically as compared to the exponential decay of the traditional queueing models [12] ,i.e., $\rho(nh) \sim (nh)^{-\alpha}$, as $|nh| \to \infty$ where $0 < \alpha < 1$. For discrete time non-decomposable Markov processes with discrete state space, if $\rho(nh)$ converges exponentially fast toward zero, S_I converges toward r and $Var(\overline{X}_I)$ converges toward $\frac{\sigma^2 r}{I}$. On the other hand, in case of decomposable systems, such as the one given in Table 2.4 if $\rho(nh)$ converges toward some value $g \neq 0$ as $n \to \infty$, then for large I, S_I increases at the rate of g as, $S_{I+1} = S_I + 2\rho(Ih) = S_I + 2(g)$, suggesting the correlation between the initial distribution and the equilibrium distribution. Hence for non-ergodic Markov chains (e.g., decomposable systems) $\rho(nh)$ converges to a constant because the system will remain in a set of states which is determined by the starting initial state. This

Table 2.4: Queueing network system with $N = 3$

$X1, X2, X3$	3,0,0	2,1,0	1,2,0	0,3,0	2,0,1	1,1,1	0,2,1	1,0,2	0,1,2	0,0,3
3,0,0						ϵ			ϵ	ϵ
2,1,0						ϵ			ϵ	ϵ
1,2,0						ϵ			ϵ	ϵ
0,3,0						ϵ			ϵ	ϵ
2,0,1			ϵ						ϵ	ϵ
1,1,1			ϵ						ϵ	ϵ
0,2,1			ϵ						ϵ	ϵ
1,0,2			ϵ			ϵ				ϵ
0,1,2			ϵ			ϵ				ϵ
0,0,3			ϵ			ϵ			ϵ	

can be visualized by considering a system represented by Table 2.4 and setting $\epsilon = 0$. As a result $S_I \to \infty$. Generally we have systems where S_I converges toward r and in these cases $Var(\overline{X})$ converges toward $\frac{\sigma^2 r}{I}$. In discrete Markov processes even in the presence of periodicity, the $\overline{X}$ converges toward μ, which is one of the conditions for ergodicity.

So far we have seen that for I observations of a short range dependent process, $Var(\overline{X}_I)$ converges toward $\frac{\sigma^2 r}{I}$, whereas for ν independent observations the $Var(\overline{X}_\nu)$ is $\frac{\sigma^2}{\nu}$. The sample size required for achieving the same variance for ν independent observations will be $I = r\nu$, i.e., one independent observation is equivalent to r dependent observations, and ν independent observations will give the same variance as $r\nu$ dependent observations.

From (2.32) and (2.33), we see that $Var(\overline{X}_I) \to 0$ as $I \to \infty$. Hence, sample mean converges toward $E[\overline{X}(T)]$.

2.12 Insights

In this section, we will discuss the insights obtained from the study of stochastic processes that provide recommendations for the convergence behavior of performance measures. We discuss this here for a DTMC. The results are similar for a CTMC.

2.12.1 Ergodicity

Stochastic processes are typically analyzed to measure steady-state means and other performance measures for different input parameters. Generally averages are used to measure the performance of a system. Therefore, to describe the physical systems in a useful way using the theory of stochastic

processes, the key requirement is to be able to measure the time average and other probabilistic quantities from observations of a stochastic process $\{X(t),\ t \geq 0\}$ such as

the mean	$m(t) = E[X(t)]$,
the covariance core	$K(s,t) = Cov[X(s), X(t)]$ and
the one-dimensional distribution function	$F_{X(t)}(x) = P[X(t) \leq x]$

If we consider a single finite record $\{X(t), t = 1, 2, 3, \ldots, T\}$ of a discrete parameter stochastic process, or a finite record $\{X(t), 0 \leq t \leq T\}$ of a continuous parameter stochastic process, it is of great consequence to know under what circumstances (if any) is it possible to use a single finite record, to estimate the quantities mentioned above. Physical systems having such properties, where the estimates obtained become more and more accurate as the length T of the record obtainable becomes larger, are called *ergodic*. In discrete-time stationary processes $\{X_t,\ t \geq 1\}$, the time average from 1 to T of an individual replication is defined as $\frac{1}{T}\sum_{t=1}^{T} x_t$. If the process is ergodic, then the time average converges to the expectation of stationary distribution $E[X]$ with probability one as $T \to \infty$, i.e., $\frac{1}{T}\sum_{t=1}^{T} x_t = E[X_t] = E[X]$ *as* $T \to \infty$. As a result $E[X]$ is viewed as the long run average of a single replication.

Consider again from Section 2.8, the example of finding proportion of time that more than five customers are in a queue. We can define, for a discrete time stationary process, an indicator function $G = I(X_t > 5)$ which will assume value of 1 if there are more than five customers (i.e. condition $X_t > 5$ is satisfied or $G \in X_t$) in the queue, otherwise it will assume value 0. The indicator function will make use of the sample function $\{x_t,\ t = 1,\ 2,\ \cdots,\ T\}$ to count the number of times t_g for which X_t satisfies G. In this case, t_g/T gives the proportion of time more than five persons were waiting in queue ($G \in X_t$). The expectation in this case is given as

$$E[I(G \in X_t)] = 0P\{I(G \in X_t) = 0\} + 1P\{I(G \in X_t) = 1\} = P\{I(G \in X_t)\}$$

As discussed in Section 2.8, it shows that time proportions calculated in this way converge toward their probabilities. Therefore, instead of computing the time average from a sample function, we can use an indicator function to find the proportion of the time that an event has occurred, and verify that ergodicity can also be based on the time proportions. It also holds equally well for processes converging to stationary processes. A given discrete parameter stochastic process $\{X(t), t = 1, 2, \cdots\}$ with a sequence of sample means $\{\overline{X}(T)\}$ where $\overline{X}(T) = \frac{1}{T}\int_0^T X(t)dt$ is formed from increasingly larger samples and

$$\lim_{T \to \infty} Var[\overline{X}(T)] \quad = \quad 0 \tag{2.34}$$

is said to be ergodic [53]. Using equation (2.34), for large enough sample sizes T and for almost all possible sample functions that could have been observed, $\overline{X}(T) \approx E[\overline{X}(T)] = \frac{1}{T}\int_0^T x(t)dt$. As a result, ergodic stochastic processes have the property that estimating the (population) averages

can be formed from the corresponding sample (or time) averages from an observed record of the process. Therefore, in an *ergodic* process, the properties of the random variables in the process can be estimated from a single time series. In general, a process is called *ergodic* if the average derived from a single replication or sample function converges to the corresponding average of several replications.

2.12.2 Correlation and Stationarity of Processes

A covariance stationary process with covariance function $K(\tau)$ is ergodic if the following two conditions hold [55]

$$\begin{aligned} \lim_{T\to\infty} \frac{1}{T} \sum_{0}^{T-1} K(\tau) &= 0 \\ \textit{and } \lim_{T\to\infty} \frac{1}{T} \sum_{0}^{T-1} K(\tau)^2 &= 0. \end{aligned}$$

In this thesis, we will consider the processes that become stationary after enough time has passed. The discussion of correlation is important in order to describe the stationarity of a process. Consider a stochastic process $\{X(t), t \geq 0\}$. Each $X(t)$ may have a different distribution, $F_{X(t)}$, called *marginal distribution*. A process is called *stationary process* if the marginal distributions are independent of t. If the distribution of a process $X(t)$ does not change as t changes, the process is said to have become *stochastically (or probabilistically) stationary*. For a random variable X, we will denote the marginal distribution of a stationary process by $F_X(.)$. The mean and standard deviation of X are denoted by μ and σ respectively. Also a process is stationary if the underlying joint distribution of random variables in the process remains consistent as the time advances, i.e., the random mechanisms generating the process are time invariant.

As t changes, it becomes interesting to explore the joint distributions of $X(t)$. Joint distributions are difficult to work with. Thus, a simpler but very important concept of (weak) stationarity in a time series analysis called *autocovariance function* was introduced. It characterizes the first two moments of the process. The covariance $Cov(X(t), X(t+\tau))$ or $K(t,\tau)$ between two random variables is defined as

$$Cov(X(t), X(t+\tau)) = E[(X(t) - \mu)(X(t+\tau) - \mu)].$$

The autocovariance function measures the dependence between different elements of the process $X(t)$. One observes the process at time t and later at time $t+\tau$ to know the strength of relation between $X(t)$ and $X(t+\tau)$, which can be measured by covariance between $X(t)$ and $X(t+\tau)$. The process is said to be *covariance stationary* if the covariance between $X(t)$ and $X(t+\tau)$ depends only on τ and not on t, i.e., $K(t,\tau) = K(\tau)$ is independent of t. The auto-correlation function of a

covariance stationary process is given as

$$\rho(\tau) = K(\tau)/\sigma^2, \quad -1 \leq \rho(\tau) \leq 1. \tag{2.35}$$

The degree of correlation is commonly measured by the auto-correlation function $\rho(\tau)$. High value of $\rho(\tau)$ represent high correlation and vice versa. Dependent random variables are typically correlated [32]. The correlation can either be positive ($0 < \rho(\tau) \leq 1$) or negative ($-1 \leq \rho(\tau) < 0$). Two random variables moving in same direction are positively correlated, and vice versa.

For the purpose of this thesis, we will be investigating processes that eventually become stationary or covariance stationary. For large enough τ when $K(\tau)$ is close to zero, $X(t)$ and $X(t+\tau)$ are practically independent. If $X(t)$ and $X(t+\tau)$ are independent and the process is stationary, the process beginning at $t+\tau$ is statistically interchangeable with a new replication or realization of process. Thus, one replication can be sufficient for estimating means and other measures of interest. Technically in ergodic processes, $\frac{1}{T}\int_0^T K(t)\mathrm{d}t$ converges to zero. The main idea is that, if a process is ergodic, a single replication is adequate for estimating means and other performance measures.

In a covariance stationary process, characteristically, the covariance $K(\tau)$ will decrease with τ. If $K(\tau)$ draws closer to zero exponentially with τ, the process is showing evidence of *short range dependence*. If the $K(\tau)$ moves towards zero hyperbolically ($K(\tau) = O(1/\tau^{\alpha})\ for\ 0 < \alpha < 1$) the process is said to exhibit *long range dependence*. For the purpose of thesis, we will be investigating the processes with short range dependence. It is worthwhile to mention here that the integral $\int_0^T K(t)\mathrm{d}t$ accurately distinguishes between short range dependence and long range dependence. The process is short range dependent if the integral converges as $T \to \infty$. The process is long range dependent if the integral does not converge, however $K(t) \to 0\ as\ t \to \infty$.

2.13 Central Limit Theorem

The central limit theorem is the basis for many statistical procedures. The central limit theorem states that under very general conditions the sum of a large number of independent random variables, each having a finite variance, is normally distributed. The distribution of a phenomenon under study may not be normal, however its average will be. The central limit theorem also holds for ergodic processes. As a result of this theorem, the normal distribution is very important.

2.14 Criteria for Models Selection

In our modeling and analysis of stochastic processes, we select the systems that are ergodic. Ergodicity in stochastic system implies that the time average converges toward the expected number in the system over a long run i.e. as $T \to \infty$, $\overline{X}(T) \to E[X]$ and the mean of time average

$E[\overline{X}(T)] = E[X],\ or\ E[\overline{X}(T)] = \frac{1}{T}\int_0^T E[X(t)]\ dt$. Ergodicity also implies that the time average converges toward the sample average or they are equivalent. In a restricted way, a system is ergodic, if the long run proportions of being in a state i converges toward equilibrium probability of state i. It is important because the long run proportion of time a process is in a certain state represents that state's equilibrium probability, which is not true for *non-ergodic* processes.

Stochastic processes may be *divergent* or *convergent*. Convergent processes converge to a steady state, whereas divergent processes will have transient states only. It is important to note here that an ergodic system will have one steady state behavior, whereas a finite state non-ergodic system has several equilibrium behaviors. A fully decomposable Markov chain is non-ergodic because it has several long run behaviours. The systems in which the state variables in long run are independent of present state are asymptotically independent. Systems with several long run probability distributions and the periodic systems are not asymptotically indpendent, as the present state in these system effect the most remote future. However, a CTMC can not be periodic, but it can be almost periodic. Hence, almost periodic systems along with all convergent systems are ergodic. Even though in almost periodic systems the sample average does not converge to expectation, the time average converges to expectation. Hence, almost periodic systems can be simulated even in the presence of periodicity.

Simulation systems can be classfied in different dimensions such as dynamic versus static, ergodic versus non-ergodic, stochastic versus deterministic, periodic versus non-periodic, convergent versus divergent, time homogeneous versus time heterogeneous. In this thesis, we will study continuous stochastic ergodic systems having time homogeneous transition rates. We will find bias, variance and MSE of time average in a number of ergodic systems that are non-decomposable and non-periodic including an almost decomposable system and an almost periodic system.

2.15 Experimental Models

The experimental models that are used in our study are briefly discussed below. These models range from queueing models to non-queueing models, and simple models to more challenging models. For the complex queueing models, different setups (i.e., series and network) are covered. These experimental models may characterize the behavior of real-world systems, as these models may constitute a component of real-world systems. The fundamental nature of our study of queueing models and non-queueing models is to compare the tendency of convergence of performance measures and simulation run length of complex queueing models with simple ones. We will use the *MES* approach to describe the models using the event table to highlight the state variable(s) that represent the state of a system, e.g., number of entities in the system.

2.15.1 The $M/M/1$ Model

Our first experimental model is the simplest and the most popular single server model, *M/M/1/N* queue. A single server queueing system containing a single queue with exponentially distributed inter-arrival and service time as shown in Figure 2.10. Including the customer in service, the *M/M/1/N* queue can accommodate at the most N customers. The *M/M/1/N* queueing model is useful to approximate a system whose service times have standard deviation approximately equal to its mean, as the mean and standard deviation of exponential distribution are equal. We define a state variable X to represent the number of customer in the system i.e. $X = i$.

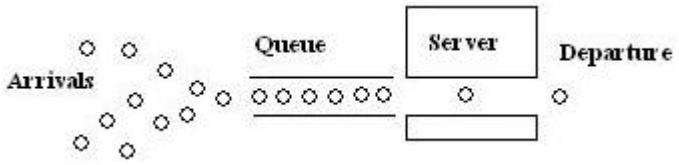


Figure 2.10: $M/M/1$ Queueing Model.

Table 2.5: Event Table for a $M/M/1$ System

Event	**X**	**Rate**	**Condition**
Arrival	+1	λ	$X < N$
Departure	-1	μ	$X > 0$

The transition matrix generated from Table 2.5 is given as

$$\mathbf{A} = \begin{bmatrix} -\lambda & \lambda & & & & \\ \mu & -(\lambda+\mu) & \lambda & & & \\ & \mu & -(\lambda+\mu) & \lambda & & \\ & & \mu & -(\lambda+\mu) & \lambda & \\ & & & \mu & -\mu & \end{bmatrix}$$

The expected number of customers in the system can be explicitly calculated with the formula given in [34]. Clearly the computational feasibility and simplicity of the *M/M/1* queue makes it an elementary investigational model of prime interest.

2.15.2 The $M/E_k/1$ Model

To vary and examine the effect of different service-time distributions, we move away from exponential service distribution observed in the $M/M/1$ model. The other single server model selected

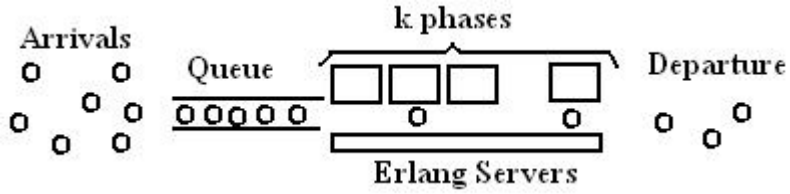


Figure 2.11: $M/E_k/1$ Queueing Model.

to examine the effect of the service time distribution on our performance measures is the $M/E_k/1$ model. The $M/E_k/1$ model is used to describe a system where an entity entering service is expected to traverse a set of k exponential phases of service, each with mean $1/k\mu$. The service, accessible to a single entity, begins in phase k and finishes after phase 1. In contrast to Markovian service pattern in the $M/M/1$ model, the main reason for choosing this model is its non-Markovian service pattern. The state of a $M/E_k/1$ system will be represented by two distinct state variables. One state variable corresponds to the number of entities in the queue and the other state variable signifies the active phase in service. We define $X1$ to represent the number of entities i in the queue, and $X2$ to signify the service phase j (where $1 \leq j \leq k$) occupied by the entity in service. The event table for this model and the corresponding transition matrix are described by the Table 2.6 and Table 2.7 respectively.

Table 2.6: Event Table for a $M/E_k/1$ System

Event	**Effect**		**Rate**	**Condition**
	X1	**X2**		
Arrival into an empty system		$+k$	λ	$X1 = 0, X2 = 0$
Arrival into a busy system	+1		λ	$X1 < N, X2 \neq 0$
Next phase		-1	$k\mu$	$X2 \geq 1$
Departure when queue is occupied	-1	$+k$	$k\mu$	$X1 > 0, X2 = 1$
Departure when queue is empty		-1	$k\mu$	$X1 = 0, X2 = 1$

2.15.3 The $M/M/c$ Model

Our next experimental model is a $M/M/c/N$ queue. The model contains a single queue and $c \geq 1$ servers operating in parallel as shown in Figure 2.12. Arrivals to the system are Poisson with rate λ and the service time distribution at each server is exponential with mean $1/\mu$. If the number of customers in system (n) is less than number of servers (c), i.e., $n < c$, the arriving customer will directly go to an available server and will leave the system after service at the rate of $n\mu$. If all the

Table 2.7: Generator matrix for a $M/E_k/1$ System

i,j	$0,0$	$0,1$	$\dots$	$0,k$	$\dots$	$N,1$	$\dots$	N,k
$0,0$	$-\lambda$			λ				
$0,1$	$k\mu$	$-(\lambda+k\mu)$			λ			
$\vdots$		$\ddots$	$\ddots$			$\ddots$		
$\vdots$			$\ddots$	$\ddots$			$\ddots$	
$N-1,k$				$k\mu$	$-(\lambda+k\mu)$			λ
$N,1$					$k\mu$	$-k\mu$		
$\vdots$					$\ddots$	$\ddots$		
N,k							$k\mu$	$-k\mu$

servers are busy, the rate of leaving system of this system is at its maximum $c\mu$. A $M/M/c$ queue can be modeled to represent a single queue of customers at a bank, being served by more that one teller operating in parallel. If the bank can accommodate N customers, then it can be modeled as a $M/M/c/N$ queue system with parameters $\lambda,\ \mu\ c$ and N. We define the state variable X to represent the number of entities i in the system, i.e., $X = i$. Table 2.8 describes the event table for $M/M/c$ model.

Table 2.8: Event Table for a $M/M/c$ System

Event	**X**	**Rate**	**Condition**
Arrival	+1	λ	$X < N$
Departure when not all servers are occupied	-1	$X\mu$	$0 < X < c$
Departure when all servers are occupied	-1	$c\mu$	$X \geq c$

The transition matrix generated from Table 2.8 is given as

$$\mathbf{A} = \begin{bmatrix} -\lambda & \lambda & & & & & & \\ \mu & -(\lambda+\mu) & \lambda & & & & & \\ & 2\mu & -(\lambda+2\mu) & \lambda & & & & \\ & & i\mu & -(\lambda+i\mu) & \lambda & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & c\mu & -(\lambda+c\mu) & \lambda & \\ & & & & & \ddots & \ddots & \ddots \\ & & & & & & c\mu & -c\mu \end{bmatrix}$$

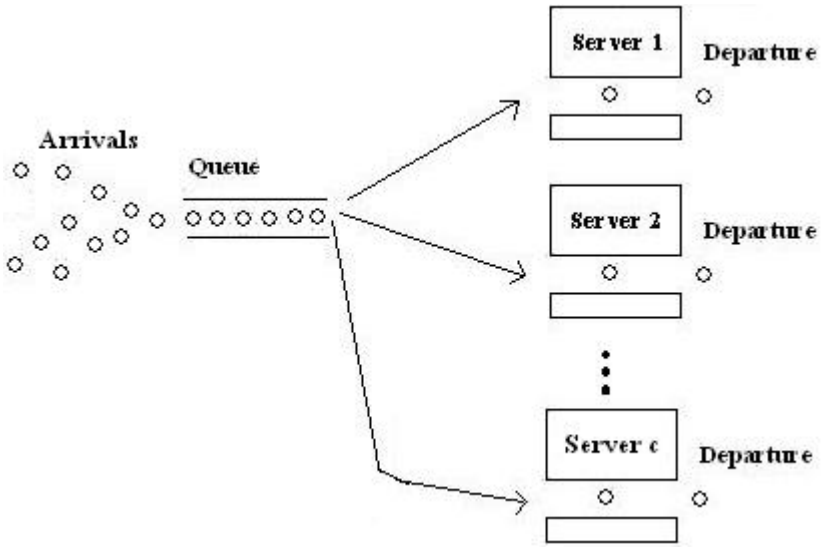


Figure 2.12: $M/M/c$ Queueing Model.

2.15.4 The Sequential Queues Model

The advantage of selecting a sequential queues (also referred to as tandem queues) model is that different queues can be examined separately for the rate of convergence of performance measures to find the tendency for the length of a simulation run by changing the properties of a Markov Chain. In this thesis, we are looking at 2 and 3 queues with $\mu_1 = \mu_2$ and $\mu_1 = \mu_2 = \mu_3$ respectively. The service time distributions are exponential with mean $1/\mu_1, 1/\mu_2$ and $1/\mu_1, 1/\mu_2, 1/\mu_3$ for 2 and 3 queues, respectively. The arrivals to the system are Poisson with rate λ. The checkout operation

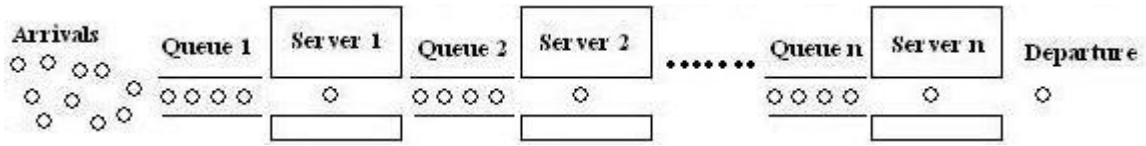


Figure 2.13: Model of Queues in Series.

in a cafeteria is a common example of multiple *M/M/1* queues in series as shown in Figure 2.13. Each queue in the setup is assumed to have a maximum capacity. Suppose the maximum number of customers a queue can accommodate is: N_1 *for queue* 1 *and* N_2 *for queue* 2 and so on. To start with, the arriving customers always join *queue* 1 first. After getting service from *server* 1, a customer moves on to join *queue* 2 for receiving service from *server* 2. After receiving the service on the last server the customer leaves the system. If an arriving customer finds *queue* 1 fully occupied, the customer balks. There are two possibilities when a customer having just finished the service

at *server* 1 cannot find waiting room in *queue* 2, either the customer will balk without blocking *queue* 1 or the customer will wait endlessly at *server* 1 thus blocking *queue* 1. Table 2.9 shows the event table for two sequential queues without blocking, whereas Table 2.10 shows the event table for two sequential queues with blocking. Similarly, one can make an event table for three or more sequential queues.

Table 2.9: Event Table for a Two Sequential Queues System without Blocking

Event	**X1**	**X2**	**Rate**	**Condition**
Arrival into Queue 1	+1		λ	$X1 < N1$
Departure from Queue 1 and System	-1		μ_1	$X1 > 0, X2 = N2$
Departure from Queue 1 and Arrival into Queue 2	-1	+1	μ_1	$X1 > 0, X2 < N2$
Departure from Queue 2 and System		-1	μ_2	$X2 > 0$

Table 2.10: Event Table for a Two Sequential Queues System with Blocking

Event	**X1**	**X2**	**Rate**	**Condition**
Arrival into Queue 1	+1		λ	$X1 < N1$
Departure from Queue 1 and Arrival into Queue 2	-1	+1	μ_1	$X1 > 0, X2 < N2$
Departure from Queue 2 and System		-1	μ_2	$X2 > 0$

2.15.5 The Closed Queueing Network Model

We examine a closed queueing network system with three exponential queues connected in a triangular form to study effect of decomposability on the performance measures and the length of a simulation run. Consider a number of delivery vehicles travelling between two queues regularly for delivery. However, sometimes vehicles may travel for maintenance to and from a third queue from either of the other two queues. The rates at which the vehicles leave queue 1, queue 2 and queue 3 are μ_1, μ_2 and μ_3 respectively. Departures from a queue i are split in such a way that they will arrive in queue j with a probability p_{ij} and arrive in the other queue with a probability $1 - p_{ij}$. As a result, the rate of going from queue i to queue j denoted by λ_{ij} is $\mu_i \times p_{ij}$. The vehicles are restricted to move inside the queueing network keeping the number of vehicles in the network fixed and unchanged. Each queue is assumed to be able to accomodate all the vehicles in the network.

The service times of all queues follow the exponential distribution. The event table for a closed queueing network system in given in Table 2.11.

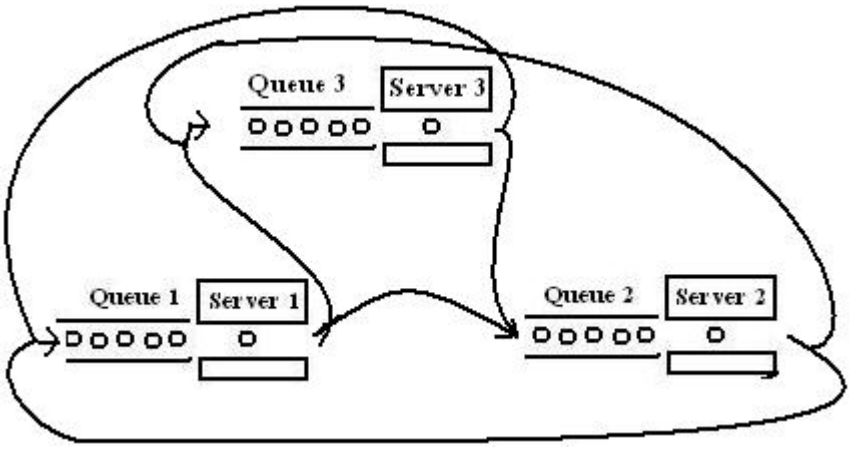


Figure 2.14: Closed Queueing Network Model.

Table 2.11: Event Table for a Queueing Network System

Event	**X1**	**X2**	**X3**	**Rate**	**Condition**
Arrival to Queue 1 from Queue 3	+1		-1	λ_{31}	$X3 > 0$
Arrival to Queue 3 from Queue 1	-1		+1	λ_{13}	$X1 > 0$
Arrival to Queue 2 from Queue 3		+1	-1	λ_{32}	$X3 > 0$
Arrival to Queue 3 from Queue 2		-1	+1	λ_{23}	$X2 > 0$
Arrival to Queue 1 from Queue 2	+1	-1		λ_{21}	$X2 > 0$
Arrival to Queue 2 from Queue 1	-1	+1		λ_{12}	$X1 > 0$

2.15.6 The Inventory Model

Our experimental model of an inventory system is an unusual candidate for our study, as it does not contain any queues. The event table of inventory system is given in Table 2.12. We consider an inventory control policy where a merchant orders N units of the product when the number in inventory drops down to zero. The arrivals of demands are Poisson with rate λ, and each demand is for exactly 1 unit. We assume that replenishment of inventory to its maximum level occurs instantaneously, i.e., the time between placing and receiving an order is zero. The merchant keeps maximum of N units of the product in his stock. The inventory models can be studied for characterizing types of policies to follow and to find the run length in systems with periodic behavior.

Table 2.12: Event Table for an Inventory System

Event	**X**	**Rate**	**Condition**
Sale and Replenishment	N	λ	$X = 1$
Sale	-1	λ	$1 < X \leq N$

Chapter 3

Simulation and Estimation of Parameters

This chapter deals with the analysis of the simulation output data of a single system. The measures of performance of a system typically contain one or more numerical parameters of the system, and are generally represented by one or more characteristics of either the state variable or by quantities derived from state variable(s). For example, in simulation of a queueing system, the principal objective is to obtain a good estimation of a useful measure of performance, such as, average waiting time in queue, average queue length, average time spent in service, average number in system, server utilization, etc. The classical statistical methods of estimation require the observations to be independent. Unfortunately, the observations obtained from simulation are dependent. However, the classical estimation procedures are still applied for obtaining the point estimate(s) (Section 3.1.1) and interval estimate(s) (Section 3.1.2).

3.1 Basic Probability and Statistics

This section shows that estimates of the unknown true value of a parameter along with a determination of its accuracy can be obtained by using point estimates (Section 3.1.1) and interval estimates (Section 3.1.2), if the random variable of interest follows a particular distribution or is represented by an empirical distribution.

3.1.1 Point Estimation

A *point estimate* of a particular parameter is a numerical value of a *statistic* or *estimator* computed from a set of sample data to reflect the true value of parameter as closely as possible. Inputs to a stochastic simulation model are random variables producing random outputs. The output data produced by a simulation experiment being random in nature is nothing more than a statistical sample, and must be treated statistically in order to estimate the true characteristics of the model examined. Hence the output data is subject to the same statistical analysis methods that are used elsewhere in statistics. Let $X_1, X_2, X_3, \cdots, X_I$ be I simulated random variables (independent or correlated) of a process with mean $E[X] = \mu$ and variance $Var(X) = \sigma^2$. The dependent observations behave similar to the independent ones. Formally, a function $f(X_1, X_2, X_3, \cdots, X_I)$ of

random variables $X_1, X_2, X_3, \cdots, X_I$ is called a *statistic*. A parameter θ of the distribution of the process $X(t)$ is estimated by a statistic $\overline{\theta}(X_1, X_2, X_3, \cdots, X_I)$ known as the *estimator* of θ. The most common goal of simulation studies is the estimation of the mean, μ, of the analyzed process. The expectation of the sample mean is assumed to estimate the value of parameter of the process, i.e., $E(\overline{X}) = \mu$. The estimate for μ is typically

$$\overline{X} = \sum_{i=1}^{I} \frac{X_i}{I} \tag{3.1}$$

An estimator, being a function of random variables, is itself a random variable. Hence, $\overline{X}$ is a random variable. The distribution of $\overline{X}$ depends on the I and the distribution of X_i.

3.1.2 Interval Estimation

An interval estimate is used as a measure of the error in the point estimate. In this section, we will first describe the method for obtaining interval estimates by considering the observations obtained from the simulation of a stochastic process to be IID. Dependent observations behave similar to independent observations.

In simulation literature, one frequently divides a long run into several sub intervals, and calculates the means of the sub intervals. These means are called *batch means*. A random variable X_i is associated with an interval or batch i. The definition of term *batch* depends on the technique applied in the simulation for calculation of the sample mean. The *batch* can be a single observation, a complete replication, or a collection of observations in a subinterval during a run. Some possible definitions for X_i are

$$X_n = \{_0^1 \tag{3.2}$$

$$\text{As time} - \text{average value for batch n} : X_n = \frac{1}{T_n - T_{n-1}} \int_{T_{n-1}}^{T_n} X_n(t)\mathrm{d}t \tag{3.3}$$

$$\text{As observation} - \text{average value for batch n} : X_n = \frac{1}{N_n} \sum_{m=b+1}^{b+N_n} X_n(m) \tag{3.4}$$

$$\text{where } b = \sum_{k=1}^{n-1} N_k \tag{3.5}$$

Consider I sampled observations (regarded as IID realizations) of random variables $X_1, X_2, X_3, \cdots, X_I$ (each of length n) having some probability distribution with realizations given as

$$x_{11}, x_{12}, \cdots, x_{1n},$$

$$x_{21}, x_{22}, \cdots, x_{2n},$$

$$\cdots, \cdots, \cdots, \cdots$$

$$x_{I1}, x_{I2}, \cdots, x_{In}$$

The *sample average* (i.e., average across different rows) of the I random variables X_i is given as

$$\overline{X} = \overline{X}_I = \frac{\sum_{i=1}^{I} X_i}{I}. \tag{3.6}$$

In ergodic systems, the time average is equal to the sample average, i.e., they are equivalent. It is desirable to collect the sample data, and then use it to construct a *confidence interval* of values that will, with high probability, contain true value of the parameter. So before sampling we insist that the proposed interval contain the true value with the specified high probability $1-\alpha$, $0 < \alpha < 1$. Here, α is called the *level of confidence*. For example, the sampling distribution of $\overline{X}$ will be used to choose lower and upper confidence limits $\overline{X}_L$ and $\overline{X}_H$ such that for a specified probability $1-\alpha$ where $0 < \alpha < 1$

$$P\{\overline{X}_L < E(X) < \overline{X}_H\} = 1 - \alpha. \tag{3.7}$$

The interval $(\overline{X}_L, \overline{X}_H)$ is called $(1-\alpha)100\%$ confidence interval. The variable $\overline{X}$ is a random variable with variance $Var(\overline{X})$. The first step to construct a confidence interval to assess the precision of $\overline{X}$ as an estimator of μ is to estimate $Var(\overline{X})$ which is given as (see [55] for details)

$$\begin{aligned} Var(\overline{X}) &= E(\overline{X}^2) - (E(\overline{X}))^2 \\ &= \frac{1}{I^2}\left(\sum_{i=1}^{I} Var(X_i) + \sum_{i=1}^{I}\sum_{j=1_{j\neq i}}^{I} Cov(X_i, X_j)\right) \end{aligned} \tag{3.8}$$

where $Cov(X_i, X_i) = Var(X_i)$, $Var(X_i) = \sigma^2, E[X_i] = \mu$. If the central limit theorem holds, $\overline{X}$ is normal as is well known, and if $\sigma_{\overline{X}} = \sqrt{Var(\overline{X})}$, then

$$\mu_{\overline{X}} - z_{\frac{\alpha}{2}}\sigma_{\overline{X}} \leq E(X) \leq \mu_{\overline{X}} + z_{\frac{\alpha}{2}}\sigma_{\overline{X}} \tag{3.9}$$

where $z_{\frac{\alpha}{2}}$ is the $\frac{\alpha}{2}$ quantile of the standardized normal distribution. Here

$$\epsilon = z_{\frac{\alpha}{2}}\sigma_{\overline{X}} \tag{3.10}$$

defines the accuracy of simulated results. As discussed in Section **??**, if correlated

$$\begin{aligned} Var(\overline{X}) &= \sigma^2_{\overline{X}} = \frac{\sigma^2 r}{I}, \\ \text{therefore,} \qquad \sqrt{Var(\overline{X})} &= \sigma_{\overline{X}} = \sqrt{\frac{\sigma^2 r}{I}}. \end{aligned}$$

Hence, to get an additional decimal digit of accuracy, the sample size must be increased by a factor of 100. Results obtained from equation (3.1), known as the *point estimate*, are used to characterize the system analyzed, whereas results obtained from equation (3.9), known as the *interval estimate*, state the accuracy of the obtained characteristics. On constructing a very large number of $(1-\alpha)100\%$ intervals, the proportion of confidence intervals containing $E(X)$, called *coverage for confidence interval*, should be $1-\alpha$. However, the accuracy of the estimator of $\overline{X}$ and $\sigma_{\overline{X}}$, and the assumption of normality may change the actual coverage probability.

3.2 Finding Variance and Confidence Interval Statistically

To construct a valid confidence interval for the parameter of interest θ, which is an important goal in simulation, the first requirement is an estimation of the variance, if it is not known. The sections 3.1.1 and 3.1.2 discussed the procedures for point estimation and interval estimation respectively, under the assumption of normality. In practise in simulation, one would first obtain the estimator $\overline{\theta}$ and then the estimator $\sigma^2_{\overline{\theta}}$. Assuming $\overline{\theta}$ to be normally distributed, equation (3.9) is used to construct $(1-\alpha)100\%$ confidence interval for θ. To obtain approximately unbiased estimates of the variance of the point estimate, $\sigma^2_{\overline{\theta}}$, is one of the major problems in simulation output analysis. There are two cases we have to deal with in simulation (*Case I*) when $\{X_1, X_2, \cdots, X_I\}$ are statistically independent and, (*Case II*) when $\{X_1, X_2, \cdots, X_I\}$ are not statistically independent. In *Case II*, when the observation are not independent, two sources of error have been observed in estimation of variance (i) the bias in $\sigma^2_{\overline{\theta}}$ as an estimator of σ^2, and (ii) the omission of covariance. Several statistical procedures developed for constructing point and interval estimates can be classified as being *fixed-sample-size procedures* [43], and *sequential sampling procedures* [43, 42]. Fixed sampling procedure fixes the sample size (number of replications and length of each run), whereas in sequential sampling more and more data are collected until an acceptable confidence interval can be constructed.

3.3 Challenges in Steady State Simulation

As noted above, one of the major problems in simulation output analysis is obtaining approximately unbiased estimates of $\sigma^2_{\overline{\theta}}$, the variance of the point estimator which is typically required for the estimation of a valid confidence interval. To estimate the long run (steady-state) average of samples from a single simulation run, one has to address many issues like run length, startup conditions, initialization bias, batch size, dependence between observations, etc. Specifically, in this section we will discuss:

- What are the statistical errors that the simulator makes?

    1. Is the variance underestimated or overestimated?
    2. By how much is the simulator off by assuming independence and using classical statistical analysis methods?
    3. Is the run length underestimated?

3.3.1 Initialization Bias and Startup Conditions

While using any method (such as Independent Replications, Batch Means, Regenerative Method etc.) to estimate the long-term performance measure (or steady-state parameter) of the system, it is important to ensure that the bias due to the initial conditions is removed to achieve at least a covariance stationary process and the behavior of simulated system will be close to that of a steady-state system. If the run of a simulation is very long, the estimators do not depend on the initial conditions; however, their rate of convergence does (see Section 2.8). Theoretically, the initial conditions don't matter much in the long run. However, the run length of an experimental run is always finite, and so introduces a bias causing the estimated steady state (or asymptotic) distribution parameter to be essentially a parameter of transient distribution. So the analysis methods experience one or both of the following problems:

1. $\overline{X}$ is not an unbiased estimator of μ, i.e., $E[\overline{X}] \neq \mu$

2. $\overline{Var}(\overline{X})$ is not an unbiased estimator of $Var(\overline{X})$. Here, $\overline{Var}(\overline{X})$ is the estimated variance of $\overline{X}$

Another reason for the initialization bias is that one cannot start simulation with a steady state distribution. No simulation would be required if one could start simulation with a steady state distribution. According to Conway [8], the problem of Initialization Bias can be resolved by the following choices:

(a) Discard data from the burn-in phase from consideration.
Even though ignoring some initial observations tend to decrease the bias, it can increase the variance.

(b) Select starting conditions to minimize the burn-in phase.
This requires starting the simulation in a state that is representative of steady-state, thus reducing the burn-in phase [71].

Madansky [46] used the MSE rather than variance as a yardstick, and showed that in the case of the M/M/1 queue, and a very long simulation run length using state 0 (idle) as initial condition rather than the steady state mean (λ/μ) minimizes the mean-square error of the estimate of mean. Madansky [46] also developed an approximate tradeoff between the number of replications and the run length of a single replication when the system begins in an empty and idle state. In more complex systems, beginning a simulation run in empty-and-idle initial condition is not easily justifiable.

3.3.2 Valid Estimates and Run Length

To discuss the validity of estimates, one needs to know what information the simulator has. From Equation (3.9) we get

$$\mu_{\overline{X}} - z_{\frac{\alpha}{2}}\sigma_{\overline{X}} \leq E(X) \leq \mu_{\overline{X}} + z_{\frac{\alpha}{2}}\sigma_{\overline{X}}.$$

There are issues with using this formula for estimation of confidence interval. The first problem is the bias of an estimator (however small) as discussed in the Section 3.3.1. If we consider bias as well, then the interval estimate becomes

$$\mu_{\overline{X}} + B(\overline{X}) - z_{\frac{\alpha}{2}}\sigma_{\overline{X}} \leq E(X) \leq \mu_{\overline{X}} + B(\overline{X}) + z_{\frac{\alpha}{2}}\sigma_{\overline{X}}$$

Here $B(\overline{X})$ gives the effect of the bias. The variance, $\sigma^2_{\overline{X}}$, is typically underestimated, and hence the $\sigma_{\overline{X}}$. However, $Var(\overline{X}(T))$ is not useful for estimating how close we are to $E(X)$. For that, we will use $MSE(\overline{X}(T))$ which gives a higher number. However, in the long run when bias is negligible, the mean square error is close to the variance. The $MSE(\overline{X}(T))$ is not available to the simulator, but it is obtainable by the approach we will use later. Depending upon the formula used for estimating the confidence interval, the run length of simulation may vary.

To illustrate the fact that the estimation of a probability by sample proportions or estimation of expectation by sample average improves with the larger sample size, we plotted in Figure 3.1 the simulation results of an $M/M/1$ queue with $\lambda = 2$ and $\mu = 3$. To create Figure 3.1, first we calculated numerically the expected values for $E[X(t)], P(X(t) > 0)$ and $P(X(t) > 5)$ and plotted the values from time $t = 0$ to time $t = 8$ at an interval of 0.25. Next we found by simulation the sample average, proportion of time the server is busy and the proportion of time more than 5 customers were present in the system for $t = 0.25, 0.5, 0.75, \ldots, 8$ for sample of size of 50 replications and we plotted the results. Finally we did the same calculations and plotted the results for a sample of size of 500 replications. Clearly the sample proportions are closer to their corresponding probabilities for larger sample size. In transient state or in equilibrium state, the values obtained from sample(s) deviate unsystematically from the actual values. However, the expectation and the probabilities in Figure 3.1 converge toward their equilibrium. Even if the simulation runs are very long, it has to stop at some point of time. One needs to make sure that one is running it long enough [1, 29, 45, 68] to obtain simulation estimators at the required precision for obtaining statistically significant results.

3.3.3 Correlation

Autocorrelation measures lack of statistical independence. The estimation of correlation is difficult. A reason for this is that the output process of virtually all simulations is non stationary (the distribution of successive observations changes over time) and autocorrelated [49, 10, 11, 39]. Figure

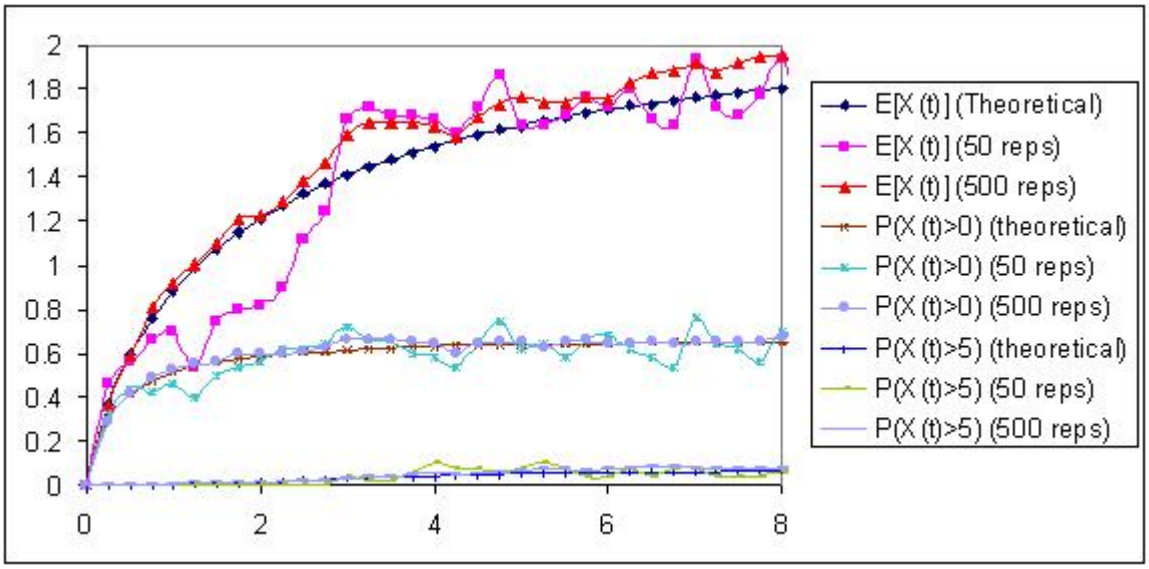


Figure 3.1: Actual and Theoretical Values of $E[X(t)], P[X(t) > 0]$ and $P[X(t) > 5]$ for an $M/M/1$ Queue

3.2 shows the correlation between the observations (at different time points) of a process for different initial conditions. It shows that the observations of the process are correlated to each other and the corrrelation decreases over time. Some conditions need to be established before performing

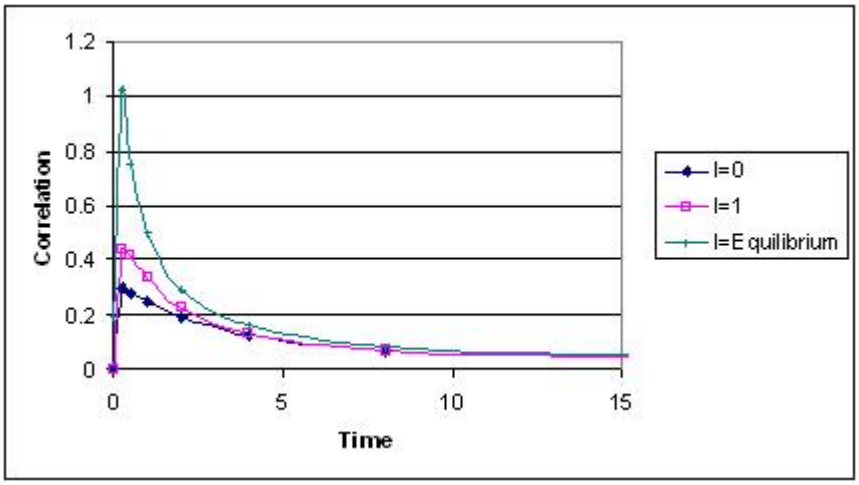


Figure 3.2: Effect of initial conditions on Correlation in $M/M/1$ Queue, $\rho = 0.4$, buffer $= 5$

the statistical analysis of the simulation output. We will be satisfied if the output is covariance stationary during the sampling period, i.e., the variance of queue length is finite and the covariance function of queue length is time-invariant.

3.3.4 Batch Size

To remove the effect of the initial bias, no attempt is made to record the output of the simulation during the transient period, which is treated as warm-up period. At the end of this warm-up period, the observations are collected for analysis. Choosing a large batch interval size would effectively lead to independent batches and hence, independent runs of the simulation. However, since the number of batches would be few, one cannot invoke the central limit theorem to construct the needed confidence interval. On the other hand, choosing a small batch interval size would effectively lead to significant correlation between successive batches. Therefore, one cannot apply the results for constructing an accurate confidence interval [2]. Unfortunately, there is no widely accepted and relatively simple method for choosing an acceptable batch size m or equivalently choosing a number of batches k. But there are some guidelines that can be picked from the research literature [63, 7, 60].

3.4 Theoretical Behavior of the Convergence of Performance Measures

In this section, we discuss our conjecture regarding the behavior of different performance measures for various systems depending on our discussion in Sections 2.2, 2.3, 2.9, 2.12. The purpose is primarily to indicate when to expect long simulation runs depending on the structural properties of the systems. The simulation run, one would conjecture, must be longer when the expected number of steps to reach the important states is large. The important states are the states with high probability or high rewards. The length of the simulation also depends on the values of the rates. If the rates are low, the system change is slow, and the simulation takes longer.

We consider two states i and j to be far apart from each other if either or both of the following conditions are true.

1. The transition rates from one state to another state are small or many steps are needed to move from state i to state j.

2. If r_i and r_j are the rewards for being in state i and j respectively, and $|r_i - r_j|$ is large.

These conditions are also applicable when one of the states is $E(X)$. Therefore, one expects a higher bias when $E(X)$ is far from an initial state than when $E(X)$ is closer to the initial state. For example, consider simulation of an $M/M/1$ queue starting in empty-and-idle condition. The reward of being in state i is i i.e. $r(i) = i$. If we keep everything else the same and increase only the buffer size, the $E(X)$ of the system increases. As the $E(X)$ moves away from initial condition, the bias is increased. The number of steps to reach $E(X)$ also increase. Consequently, the required length of

Table 3.1: Comparative rates for $M/M/1/N$, $M/M/2/N$ and $M/M/4/N$ systems with $\lambda = 9$ *and* $\rho = 0.9$

M/M/1/N					
	9				
10		9			
	10		9		
		10		$\ddots$	
			$\ddots$		9
				10	

M/M/2/N					
	9				
5		9			
	10		9		
		10		$\ddots$	
			$\ddots$		9
				10	

M/M/4/N					
	9				
2.5		9			
	5		9		
		7.5		$\ddots$	
			$\ddots$		9
				10	

the simulation run increases. On the other hand, if it is difficult to move far away from the state close to the expectation, then the variance and possibly the bias will decrease. For example, we compare the transition rates of $M/M/1/N$, $M/M/2/N$ and $M/M/4/N$ systems for the parameters $\lambda = 9$ *and* $\rho = \frac{\lambda}{c\mu} = 0.9$ (see Table 3.1). If we represent the matrics for $M/M/1/N$, $M/M/2/N$ and $M/M/4/N$ systems by $M1$, $M2$ and $M4$ respectively, then clearly we have rates for going to lower states in $M4 < M2 < M1$ i.e. the rates for $M/M/4/N$ system are lowest. Therefore, once the expectation is reached, it becomes harder to move far away from the $E(X)$. So, we expect the bias to converge faster for the $M/M/4/N$ system. As discussed in Section 2.11, the variance is related to the bias. As a result, the variance might also converge faster for the $M/M/4/N$ system. Consequently, simulation run length for $M/M/4/N$ system will be shorter.

Queueing networks can be made almost decomposable. Table 2.4 on page 37 shows an almost decomposable queueing network system with $N = 3$. To do that, one has to select a center such that the arrivals to the center and departure from the center are rare. We conjecture that the bias will increase as the rates to and from the selected center decrease. We need to investigate the nearly closed queueing network system for the behaviour of the bias and the variance. In this case, the classes are formed by the number in the other centers. Similarly, open queueing networks are almost decomposable if arrivals to and departures from the network are rare. However, we are not examining open queueing networks in this thesis.

The inventory system selected represents an almost periodic system. In the discrete case, $\overline{X}(T)$ is in steady state for $T = N, 2N, 3N, \ldots$ etc. and there will be no bias. Each cycle will be exactly of length N. After N steps the system will be in steady-state and there will be no bias. In the

discrete case, the matrix formulated for the inventory system described by Table 2.12 will be

$$\mathbf{A} = \begin{bmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ 1 & & & & \end{bmatrix}$$

The matrix for the continuous case for inventory system described by Table 2.12 will be

$$\mathbf{A} = \begin{bmatrix} & \lambda & & & \\ & & \lambda & & \\ & & & \ddots & \\ & & & & \lambda \\ \lambda & & & & \end{bmatrix}$$

This system is an almost periodic system because the expected number of steps to reach the starting state next time is close to integer N. Therefore, the bias will be small after N steps. We need to investigate the behaviour of variance and bias.

In summary, we conjecture that the quality of results obtained by simulation is affected by particular types of measures used. The convergence of our performance measures will take longer when the transition rates are low, as it will take longer to reach important states. This can be examined for different values of ρ for the selected systems. The convergence will also be slower when the number of steps to reach the important states are large. This can be experimented with different initial conditions, system capacities, number of queues and number of servers for the selected systems.

Chapter 4

Analytical and Numerical Methods

The versatility of simulation makes it very popular. However, despite substantial improvements in computing power and simulation software, simulation is still a slow and expensive way to study complex stochastic systems that perform continuously. When applicable, analytical methods can serve a complementary role for studying stochastic systems to significant advantage. An analytical method is favorably suited to the preliminary analysis of a system for studying causal relationships. In absence of such a less expensive procedure or with a very complex problem, simulation routinely provides the only practical approach to a problem. A preferred mathematical model will reasonably abstract the essence of a problem, will reveal the essential structure of the problem and will supply the necessary information for satisfactory results. In this chapter, we briefly describe the analytical procedure used to obtain information regarding performance measures in a MES simulation. Section 4.1 describes the method used for numbering of states in a MES. In Section 4.2 we discuss some of the methods for finding transient probabilities. Algorithms for finding transient solutions such as expectation of a time average, bias of a time average and variance and MSE of a time average are discussed in Section 4.3. In Section 4.4 we discuss the state reduction method which is useful for obtaining equilibrium solutions such as expected rewards. Finally, in Section 4.5 we verify the accuracy of our results, obtained for various models, using a MES.

4.1 State Numbering

The transient and steady-state solutions of many stochastic processes with a finite number of states can be found by converting them into a CTMC. The foremost step in analytical study of a MES is to find the number of states in a system. In one-state variable models, the expression $N + 1$ determines the number of states in a system. However, for two-state variable models, such as $M/E_k/1$ and a sequential system with two queues, the allowable state combinations of the variables $X1$ and $X2$ up to the maximum attainable respective capacities $N1$ and $N2$ (see Table 4.1) determine the number of states in the system. It can be further extended to three or more-state variable systems. Here, the variables $N1$ and $N2$ represent the maximum number of customers allowed in queue 1 and queue 2, respectively. The states are numbered as shown in Table 4.1.

Table 4.1: State Description in a Sequential Queueing System with Two Queues

X1	**X2**	**State Number**
0	0	0
0	1	1
0	2	2
⋮	⋮	⋮
0	N2	N2
1	0	N2+1
⋮	⋮	⋮
N1	N2	(N1+1)(N2+1)-1

After determining the maximum number of states in a MES, transition matrices for experimental models are developed by using the transition rates. The systems that we are investigating require only rates for building a transition matrix. These rates are associated with the events, such as arrivals, departures, changes of phase, and so on. Several cases of systems are examined by altering the buffer lengths, arrival rates and/or service rates. The rates are chosen in a manner to study different alternatives of a MES like underutilized systems, balanced systems, overloaded systems. The rates also influence the degree of decomposability of a system. Each rate a_{ij} represents the rate of transition from state i to state j. As shown in (4.1), the transition rates describe a CTMC by formulating a transition matrix $Q = [a_{ij}]$.

$$\mathbf{Q} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \tag{4.1}$$

There is no rate of staying in a state. The convention is to use the diagonal entry in a row to express the sum of all the off diagonal entries in that row multiplied by -1. For a CTMC, the diagonal entry in each row is set to be the negative of sum of all other rates of its row (see (4.2)) representing the total rate of leaving current state.

$$a_{ii} \quad = \quad - \sum_{j=1_{j \neq i}}^{N} a_{ij} \tag{4.2}$$

The states of finite-state queueing system are defined by d non-negative variables $X_1, X_2, \ldots, X_d$. The X_n may be queue lengths, phase types etc. Suppose each of the X_n's can assume N_n possible values. The number of states in this case are $N_1 \times N_2 \times \ldots \times N_d$ and the matrix will contain $(N_1 \times N_2 \times \ldots \times N_d)^2$ elements. Even though for the purpose of this thesis we use all the entries of

transition matrix for finding equilibrium solutions, in principle, the transition matrices are usually sparse and one needs only non-zero elements for numerical methods. Since numerical methods manipulate the transition matrix for finding the solution, the numerical methods are faster than simulation [29] for small values of d. In such cases we prefer numerical methods to simulation methods. However, the effort of numerical methods increase exponentially with d [29], and this eventually makes numerical methods much slower than simulation. The following sections discuss different analytical approaches for finding transient and equilibrium solutions.

4.2 Transient Solutions

Transient solutions are essential to our analysis in order to study transient behavior of systems that converge slowly toward their steady state. Transient solutions describe the behavior of a system before it approaches the steady state. The transient probabilities $\pi_i(t)$, and the probability of being in state i at time $t > 0$ are essential for the transient solutions. The initial probabilities are given as $\pi_j(0)$, for $j = 1, 2, \ldots$. Once $\pi_j(t)$'s are obtained, other measures of interest, like expected number in the system, mean of time average, bias, variance of time average, run length etc. can be computed easily. In literature, many methods are suggested to find transient solutions of Markov processes, such as, Euler's method, Runge-Kutta method and similar numerical integration methods [41]. One of the widely used methods [51] is the method of Runge-Kutta for solving differential equations. The probability of being in state j at time $t > 0$, i.e., $\pi_j(t)$, can be found by solving following differential equations (4.3):

$$\frac{d\pi_j(t)}{dt} = \sum_{i=1}^{N} a_{ij}\ \pi_i(t) \tag{4.3}$$

For large values of N, the computation using the Runge-Kutta method becomes very expensive. The randomization method [23, 27, 25], however, has been found to be numerically stable, and hence, it is chosen for the computation of transient solutions in this thesis. The transient solution technique that we applied allows us to examine the behaviour of the system from start until it has reached equilibrium. In addition, it also allows us to experiment with different initial conditions.

4.2.1 The Randomization Method

The randomization method embeds a discrete Markov process in a Poisson process. Consider a CTMC $X(t)$ with N number of states, generator matrix Q and a finite state space Ω. The initial probabilities are given by $\pi(0) = [\pi_1(0), \pi_2(0), \cdots, \pi_N(0)]$. In matrix notation, the solution for equations in (4.3) can be written as

$$\pi(t) = \pi(0)\ exp\ (Qt) \tag{4.4}$$

The $\pi(t)$ can be found by expanding the following power series.

$$\begin{aligned}\pi(t) &= \sum_{i=0}^{\infty} \pi(0)\ \frac{(Qt)^i}{i!} \\ &= \sum_{i=0}^{m-1} \pi(0)\ \frac{(Qt)^i}{i!} + \mathbf{R}_m \end{aligned} \tag{4.5}$$

where $\mathbf{R}_m$ is a row vector containing the truncation error of using first m terms of the power series to estimate $\pi(t)$. Since Q contains negative diagonal elements, the round-off errors in the computation of $\pi(t)$ are high, especially for high values of t. Therefore, instead of using (4.5), a preferred method for the calculation of $\pi(t)$ will require a series expansion of a matrix $P = [p_{ij}]$ which contains no negative elements. We define a Poisson process with a rate $F \geq |diag\{Q\}|$ and a uniformized Markov chain having state transition matrix given as

$$\mathbf{P} = \frac{\mathbf{Q}}{F} + \mathbf{I} \tag{4.6}$$

The matrix P generated this way will have non-negative diagonal elements. The off-diagonal elements remain non-negative. The row sums of P are 1 and hence, P is stochastic by choice of F. From (4.4), we get

$$\begin{aligned}\pi(t) &= \pi(0)\ exp\ (Qt) \\ &= \pi(0)\ exp\left[\left(\frac{\mathbf{Q}}{F} + \mathbf{I}\right) Ft - \mathbf{I}Ft\right] \\ &= \pi(0)\ exp(\mathbf{P}Ft)\ exp(-Ft) \\ &= \sum_{i=0}^{\infty} \pi(0)\ \mathbf{P}^i\ \left[(Ft)^i\ \frac{exp(-Ft)}{i!}\right] \\ &= \sum_{i=0}^{m-1} \pi(0)\ \mathbf{P}^i\ \left[exp(-Ft)\ \frac{(Ft)^i}{i!}\right] + \mathbf{R}_m \end{aligned} \tag{4.7}$$

The Poisson density $p(i; Ft)$ can be represented as

$$p(i; Ft) = exp(-Ft)\ \frac{(Ft)^i}{i!}.$$

Consequently, Equation (4.7) can be simplified as

$$\pi(t) = \sum_{i=0}^{m-1} \pi(0)\mathbf{P}^i\ p(i; Ft) + \mathbf{R}_m \tag{4.8}$$

In (4.8), for large value of i, the computation of P^i is resource intensive. This is because of the number of operations involved. In particular, each iteration will require complexity $O(N^3)$. This is facilitated by introducing a recursive computing method as

$$\begin{aligned}\Pi^{(0)} &= \pi(0) \\ \Pi^{(i)} &= \pi^{(i-1)}\ P \end{aligned} \tag{4.9}$$

this results in $\Pi^{(i)} = \pi^{(0)}\ P^{(i-1)}$. Therefore, (4.8) is further simplified to

$$\pi(t) \quad = \quad \sum_{i=0}^{m-1} \Pi^{(i)}\ p(i; Ft) + \mathbf{R}_m \tag{4.10}$$

where $\pi(t) = [\pi_i(t)]_{i=1,2,...}$ and $\Pi^i = [\Pi_j^{(i)}]_{j=1,2,...}$.

Grassmann [25] described how $\mathbf{R}_m$ can be estimated with reasonable accuracy by the cumulative Poisson distribution. To reiterate, the randomization method is numerically robust because the computations deal only with positive elements. In addition, the structure of the matrix P is preserved by recursively computing $\Pi^{(i)}$, which is important because P is sparse. This also holds when dealing with large systems. The method of randomization was demonstrated superior to the Runge-Kutta method in most cases. We applied the algorithm given in [21] for finding the transient solutions for an $M/M/1$ queue. The means and variances of time averages in transient Markovian systems of significant size are computed using an efficient algorithm given in [27] which uses the method of randomization. For the exact computer algorithm formulation, we refer the reader to [45].

4.3 Algorithm for Computing Performance Measures

The measures of interest such as the mean $E[\overline{X}(T)]$, the variance $Var[\overline{X}(T)]$, the bias $Bias[\overline{X}(T)]$ and the Mean Square Error $MSE[\overline{X}(T)]$ can be easily computed after obtaining the transient probability $\pi(t)$ and steady-state probability, from (4.10). Grassmann [27] adopted the randomization method for finding transient solutions in a CTMC. Earlier, similarity transforms [37, 56] were used extensively for finding the mean and variance. With this method, the calculation of eigenvalues for large systems seems infeasible. However, it worked well with small systems. In contrast, Grassmann's method [27] is suitable for both small and large systems, and is a fast-converging method to compute the mean and the variance of time averages in Markovian systems. In view of these facts, Grassmann's method [27] will be used in this thesis. Moreover, the scope of this method is extended to compute the measures for all the models examined. The following sections discuss the essential algorithms for computing the measures of interest.

4.3.1 The Expectation of a Time Average

In Section 2.8, we have seen how the probabilities reflect the potential behaviour of a continuous time process $X(t)$ as they converge toward their respective equilibrium. If $\pi_i(t)$ denote the transient probabilities at time t, then the expectation at time t is typically given by

$$E[X(t)] \quad = \quad \sum_{i=1}^{N} \pi_i(t)\ f_x(i) \tag{4.11}$$

Applying randomization from (4.10), we get

$$E[X(t)] = \sum_{n=0}^{m-1} \sum_{i} \Pi_i^{(n)} \; p(n; Ft) \; f_x(i) + \mathbf{R}_m \tag{4.12}$$

Consider a DTMC having a probability distribution $\Pi_i^{(n)}$ after n steps. If X_n denotes the number in the system after n steps, then the expectation is given as

$$E(X_n) = \sum_{i} \Pi_i^{(n)} \; f_x(i) \tag{4.13}$$

From (4.12) and (4.13), we get

$$E[X(t)] = \sum_{n=0}^{m-1} p(n; Ft) \; E(X_n) + \mathbf{R}_m \tag{4.14}$$

Considering $E(X_n)$ to be bounded for $0 \leq n \leq m$, one can apply Fubini's theorem and find from (2.9) and (4.14)

$$E[\overline{X}(T)] = \sum_{n=0}^{m-1} \frac{1}{T} \int_0^T p(n; Ft)\mathrm{d}t \; E(X_n) + \mathbf{R}_m \tag{4.15}$$

If $q(n; FT)$ denotes the time average of the Poisson random variable from 0 to T then (see e.g. [35])

$$\begin{aligned} q(n; FT) &= \frac{1}{T} \int_0^T p(n; Ft)\mathrm{d}t \\ &= \frac{1}{FT} \sum_{j=n+1}^{\infty} p(j; FT)\mathrm{d}t \end{aligned} \tag{4.16}$$

Therefore, the expectation of a time average can be computed from (4.15) as

$$E[\overline{X}(T)] = \sum_{n=0}^{m-1} q(n; FT) \; E(X_n) + \mathbf{R}_m \tag{4.17}$$

4.3.2 The Bias of a Time Average

In this section, we describe a method to determine the bias of a time average. By definition (see Section 2.7.1), in absolute terms, the bias of a time average can be easily computed as

$$B[\overline{X}(T)] = E(X) - E[\overline{X}(T)].$$

In terms of state probabilities, from (2.27) and (4.11), this can be expressed as

$$\begin{aligned} B[\overline{X}(T)] &= \sum_{i} \pi_i \; f_x(i) - \sum_{i} \pi_i(t) \; f_x(i) \\ &= \sum_{i} f_x(i)(\pi_i - \pi_i(t)), \end{aligned} \tag{4.18}$$

where $\pi_i - \pi_i(t)$ measures the deviation between the equilibrium probability and the transient probability.

4.3.3 The Variance and MSE of a Time Average

In this section, we describe a method to determine the variance of a time average. By definition (see Section 2.7.2) the variance of a time average can be computed as

$$Var[\overline{X}(T)] = E[\overline{X}(T)^2] - E^2[\overline{X}(T)]. \tag{4.19}$$

Since $E[\overline{X}(T)]$ can be obtained from (4.17), we need to determine $E[\overline{X}(T)^2]$ and with it $Var[\overline{X}(T)]$. From (2.8) we have $\overline{X}(T) = \frac{1}{T}\int_0^T X(t)\ \mathrm{d}t$. It follows that

$$\begin{aligned}
\overline{X}(T)^2 &= \left[\frac{1}{T}\int_0^T X(t)\ \mathrm{d}t\right]^2 \\
&= \frac{1}{T^2}\int_0^T X(s)\mathrm{d}s \int_0^T X(t)\mathrm{d}t \\
&= \frac{1}{T^2}\int_0^T \int_0^T X(s)X(t)\mathrm{d}s\mathrm{d}t \\
&= \frac{2}{T^2}\int\int_{0<s<t<T} X(s)X(t)\mathrm{d}s\mathrm{d}t
\end{aligned}$$

According to Grassmann [27], using Fubini's theorem one obtains

$$E[\overline{X}(T)^2] = \frac{2}{T^2}\int\int_{0<s<t<T} E[X(s)X(t)]\mathrm{d}s\mathrm{d}t. \tag{4.20}$$

Here, the probability distribution of $X(s)$ and $X(t)$ is by definition $\pi_i(s)$ and $p_{ij}(t-s)$. The random variable $X(s)$ represents the state of the system for continuous time process at time s, while $X(t)$ represents the state of the system in the discrete- time process in time $t-s$. The randomization method can be applied to determine the values of $\pi_i(s)$ and $p_{ij}(t-s)$, and consequently $E[X(s)X(t)]$. Therefore,

$$\begin{aligned}
E[X(s)X(t)] &= \sum_i f_x(i)\pi_i(s)\sum_j f_x(j)p_{ij}(t-s) \\
&= \sum_i\sum_j\sum_m\sum_n f_x(i)f_x(j)\Pi_i^{(m)}\Pi_i^{(n-m)} \\
&\quad \times p(m;Fs)p(n-m;F(t-s)).
\end{aligned} \tag{4.21}$$

Note that, for two random variables X_m and X_n representing the number in the system after m and n jumps respectively, the expectation of (X_mX_n) is given as

$$E(X_mX_n) = \sum_i\sum_j f_x(i)f_x(j)\Pi_i^{(m)}\Pi_i^{(n-m)}. \tag{4.22}$$

Substituting (4.22) into (4.21), we get

$$E[X(s)X(t)] = \sum_{0\le m}\sum_{\le n} E[X_mX_n]p(m;Fs)p(n-m;F(t-s)). \tag{4.23}$$

and (4.20) becomes

$$\begin{aligned} E[\overline{X}(T)^2] &= \frac{2}{T^2}\sum_{0\le m}\sum_{\le n} E[X_m X_n] \\ &\quad \times \int\int_{0<s<t<T} p(m;Fs)p(n-m;F(t-s))\mathrm{d}s\mathrm{d}t \\ &= \sum_{0\le m}\sum_{\le n} E[X_m X_n]\ I_n. \end{aligned} \tag{4.24}$$

According to Grassmann [27], by applying Beta and Gamma functions I_n is determined as

$$\begin{aligned} I_n &= \frac{2}{T^2}\int_0^T p(m;Fs)\int_0^{T-s} p(n-m;Ft)\mathrm{d}t\mathrm{d}s \\ &= \frac{2}{FT}q(n+1;FT). \end{aligned} \tag{4.25}$$

If we define

$$S_n = \sum_{m=0}^{n} E(X_m X_n), \tag{4.26}$$

then from (4.25) and (4.26) we get

$$E[\overline{X}(T)^2] = \frac{2}{FT}\sum_{n=0}^{m} q(n+1;FT)S_n + r_m, \tag{4.27}$$

where r_m is the truncation error. The joint distribution of X_m and X_n can, in principle, be used to calculate $E(X_m X_n)$. One could, for instance, form the matrix $\boldsymbol{P}$, and find $\boldsymbol{P}^{n-l}$, $l, n = 0, 1, \ldots, m-1$. Unfortunately, for a matrix of dimension $N \times N$, the determination of $(m-1)^{th}$ power requires $O((m-1)N^3)$ operations, which is computationally impractical. For example, a large system with $N = 1000$ and $m = 1001$ will require a trillion operations. The computation of $E(X_m X_n)$ is avoided by calculating S_n directly by first defining

$$\begin{aligned} D_m^n &= \sum_{i=1}^{n}\sum_{r=1}^{N} f_x(r)P(X_n = m, X_i = r) \qquad n > 0 \\ D_m^0 &= f_x(m)\pi_m(0). \end{aligned}$$

The D_m^n, $n > 0$ can be calculated recursively as

$$D_m^n = \sum_{s=1}^{N} D_s^{n-1}P_{sm} + f_x(m)\pi_m^n \qquad n > 0. \tag{4.28}$$

Now, S_n can be expressed in terms of D_m^n as follows

$$S_n = \sum_{m=1}^{N} f_x(m)D_m^n. \tag{4.29}$$

$E[\overline{X}(T)^2]$ can be easily calculated by substituting the resulting value of (4.29) into (4.27). This is then substituted into (2.13) to obtain $Var[\overline{X}(T)]$.

The $MSE[\overline{X}(T)]$ can now be obtained effortlessly by substituting the results of (2.13) and (2.12) into (2.14), i.e.

$$MSE(\overline{X}(T)) = Var(\overline{X}(T)) + B^2(\overline{X}(T))$$

4.4 Equilibrium Solutions

In our modeling and analysis of stochastic processes, we assume the systems are ergodic. Ergodicity in stochastic system implies that the time average converges toward the expected number in the system over a long run, that is as $T \to \infty$, $\overline{X}(T) \to E[X]$ and the mean of time average $E[\overline{X}(T)] \to E[X]$, where $E[\overline{X}(T)] = \frac{1}{T}\int_0^T E[X(t)]\ \mathrm{d}t$. When the system is ergodic, the distribution $\overline{X}(T)$ converges towards a normal distribution as $T \to \infty$, i.e., the system approaches equilibrium. Also, as $T \to \infty$, the transition probability $\pi(t)$ converges to an equilibrium probability π. The calculation of equilibrium solutions for complex systems rely on more verstaile numerical methods. However, for simple queueing systems the equilibrium probability can be computed directly using readily available closed form solutions e.g. state reduction method, Gauss-Seidel method, Gauss-Jordan method etc. Each of the methods meets some measure of the computational complexity. However, the state reduction method, proposed by Grassmann [28, 24, 31], is found to be most efficient and numerically stable, as it eliminates the need of subtraction operations and minimizes the round-off errors. This makes it the method of choice for computation of equilibrium probabilities.

4.4.1 The State Reduction Method

The state reduction method efficiently computes the equilibrium probabilities of certain Markov chains. At equilibrium $\pi_j^{'}(t) = 0$, because the rate of change for all $\pi_j(t)$ with respect to t are zero. Therefore, from (4.3) at equilibrium, we get

$$0 = \sum_{i=1}^{N} \pi_i\ a_{ij} \tag{4.30}$$

$$\text{where} \quad \sum_{i=1}^{N} \pi_i = 1 \tag{4.31}$$

where $\pi_i = \lim_{t\to\infty} \pi_i(t)$. In the matrix Q with N states, the state reduction method computes as follows. First, the state reduction method reduces the number of states one at a time until it reaches one state.

$$a_{ij}^{n-1} = a_{ij}^n + a_{in}^n \frac{a_{nj}^n}{\sum_{j=1}^{n-1} a_{nj}^n} \quad for \ \ i,j = 1,2,\ldots,n-1 \ \ and \ \ n = N-1, N-2, \ldots, 2 \tag{4.32}$$

The following $N-1$ equations for π_n are obtained.

$$\pi_n = \sum_{i=1}^{n-1} \pi_i \frac{a_{in}^n}{\sum_{j=1}^{n-1} a_{nj}^n} \tag{4.33}$$

After obtaining a matrix with one state only, one substitutes backwards to find all π_n's in terms of π_1 from (4.33). The π_1 is then easily obtained subject to (4.31). The actual values of π_n are subsequently obtained. The state reduction method only deals with addition operations, thus making it numerically stable and less susceptible to rounding errors.

4.5 Accuracy of Results

The results in this section are generated using VBA in Excel from a system operating under the Windows XP operating system. The values are manipulated as IEEE 64-bit (8-byte) floating-point numbers ranging in value from -1.79769313486231E308 to -4.94065645841247E-324 for negative values and from 4.94065645841247E-324 to 1.79769313486232E308 for positive values. The type declaration for these values is *Double* in VBA. It is important to generate the results that are as accurate as possible. For example, a 1% error in forecasted attendance of an event is reasonably acceptable. The accuracy of the results of our analysis was verified by means of comparing the variance of accumulated total reward with the variance of time average. Another approach is to compare the expectations in the algorithm before and after reallocating the rewards.

4.5.1 The Variance of Accumulated Total Reward

Consider a system simulated from time 0 to T, where $X(t)$ represents the state of the system at time $0 \leq t \leq T$. Furthermore, $I(X(t) = i)$ is 1 if the system at time t is in state i, and zero otherwise. If reward in state i is $r(i)$ per time unit, then the actual rate at which rewards $r(t)$ are accumulated

$$r(t) = \sum_i I(X(t) = i) r(i),$$

such that, $r(t) = r(i)$. According to Grassmann [29], the reward that accumulated during the time 0 to T, i.e.

$$\Upsilon = \int_0^T r(t) \mathrm{d}t,$$

is the accumulated total reward Υ in $(0, T)$. As defined in Section 2.5.1, t_i is the total time during the period $[0, T]$ for which the system contains exactly i customers. Therefore, the proportion of time that exactly i customers were in the system is given as $\frac{t_i}{T}$. The accumulated total reward during simulation from time 0 to time T is

$$\Upsilon = \sum_{i=0}^{N} r_i t_i.$$

If $T \to \infty$, one has

$$Var(\Upsilon) \quad = \quad 2T \int_0^{\infty} Cov(t)\mathrm{d}t, \tag{4.34}$$

here,

$$Cov(t) = Cov(r(0), r(t)) = \sum_{i=0}^{N} \sum_{j=0}^{N} \pi_i p_{ij}(t)(r(0) - \mu_k)(r(t) - \mu_k),$$

and $p_{ij}(t)$ is the probability of making a transition from state i to state j during a time of length t. Recall from Section 2.5.1, $\frac{t_i}{T} \to \pi_i$ as $T \to \infty$. Therefore, we have $\mu_k = \lim_{T\to\infty} \sum_{i=0}^{N} r_i \frac{t_i}{T} = \sum_{i=0}^{N} r_i \pi_i = \frac{\Upsilon}{T} = E(X)$. Grassmann [29] provided a relatively simple way to evaluate $Var(\Upsilon)$ from a set of equilibrium equations (4.35 - 4.37) as follows:

$$0 \quad = \quad \pi_j(r_j - \Upsilon) + \sum_i v^*(i) a_{ij} \qquad for\ all\ j, \tag{4.35}$$

$$c \quad = \quad \sum_i v^*(i), \tag{4.36}$$

$$v(i) \quad = \quad v^*(i) - c\pi_i \qquad for\ all\ i, \tag{4.37}$$

$$\text{Such that,} \quad \sum_i v(i) \quad = \quad 0.$$

After computing the $v(i)$ values as a by-product of state reduction method, one has

$$\int_0^{\infty} Cov(t)\mathrm{d}t \quad = \quad \sum_{i=0}^{N} r_i v(i) + \epsilon_i, \tag{4.38}$$

where ϵ_i is the truncation error. Therefore, (4.34) becomes

$$Var(\Upsilon) \quad = \quad 2T \sum_{i=0}^{N} r_i v(i) + \epsilon_{i,1}, \tag{4.39}$$

to be rewritten as

$$\frac{Var(\Upsilon)}{T} \quad = \quad 2 \sum_{i=0}^{N} r_i v(i) + \epsilon_{i,2}. \tag{4.40}$$

In transient analysis, after the equilibrium is reached, the variance of the time average is expected to tend towards the variance of accumulated total reward over the simulation period $[0, T]$, i.e.,

$$Var[\overline{X}(T)] \to \frac{Var(\Upsilon)}{T}, \tag{4.41}$$

alternatively (as shown in Figure 2.2),

$$T \times Var[\overline{X}(T)] \to Var(\Upsilon). \tag{4.42}$$

Table 4.2 illustrates a test of (4.42) in various models. The table shows closer values for $Var[\overline{X}(T)]$ and $Var(\Upsilon)/T$.

Table 4.2: Evaluation of $Var[\overline{X}(T)]$ against $Var(\Upsilon)/T$ in Various models.

Case	Model	Parameters	T	$Var[\overline{X}(T)]$	$Var(\Upsilon)/T$
1	$M/M/1$	$\lambda = 9, \mu = 10, N = 15$	1024	0.07132273	0.07163776
2	$M/M/2$	$\lambda = 9, \mu = 5, N = 15$	1024	0.064047917	0.064315786
3	$M/M/4$	$\lambda = 9, \mu = 2.5, N = 15$	1024	0.048080084	0.048254325
4	$Periodic$	$\lambda = 1, N = 5$	512	0.003910821	0.00390625

4.5.2 Reallocation of Rewards and Expectations

In the second step, to ascertain the accuracy of the results of our analysis, we reallocate our reward function such that

$$r_i = f_x(i) - \mu_k = f_x(i) - E(X).$$

Here, $E(X)$ is derived using (2.27), as discussed in Section 2.10. The expectations, i.e., $E\left(\overline{X}(T)\right)$ and $E\left(\overline{X}(T)^2\right)$, in our algorithm is recomputed as $E\left(\overline{X}(T) - \mu_k\right)$ and $E\left(\overline{X}(T) - \mu_k\right)^2$ respectively. Therefore, the new mean of a time average is

$$\begin{aligned} E\left(\overline{X}(T) - \mu_k\right) &= E\left(\overline{X}(T)\right) - E\left(\mu_k\right) \\ &= E\left(\overline{X}(T)\right) - E(X) \\ &= B\left(\overline{X}(T)\right), \end{aligned} \tag{4.43}$$

and for $E\left(\overline{X}(T) - \mu_k\right)^2$, we have

$$E\left(\overline{X}(T) - \mu_k\right)^2 = E\left(\overline{X}(T) - E(X)\right)^2$$

substituting (2.14), one has

$$E\left(\overline{X}(T) - \mu_k\right)^2 = MSE\left(\overline{X}(T)\right). \tag{4.44}$$

Therefore, the reallocation of rewards, (4.43) and (4.44), are expected to hold. This is also tested for various experimental models. The results are shown in Table 4.3. One would observe the closeness in corrosponding figures, thought slight difference appear due to round-off errors. The observations in Tables 4.2 and 4.3 confirm the accuracy of our results by showing that (4.42), (4.43) and (4.44) hold true in this study.

Table 4.3: Reallocation of rewards and expectations.

	Model	Parameters	T	$B[\overline{X}(T)]$	$MSE[\overline{X}(T)]$	$E[\overline{X}(T)-\mu]$	$E[\overline{X}(T)-\mu]^2$
1	$M/M/1$	$\lambda=9,\mu=10,N=15$	256	0.035904561	0.281509765	0.035904561	0.282799878
			512	0.01795228	0.142014927	0.017952281	0.14233771
			1024	0.00897614	0.071322476	0.00897614	0.071403301
2	$M/M/2$	$\lambda=9,\mu=5,N=15$	256	0.034724464	0.252976297	0.034724465	0.254183196
			512	0.017362232	0.127559552	0.017362232	0.127861567
			1024	0.008681116	0.064047628	0.008681116	0.064123278
3	$M/M/4$	$\lambda=9,\mu=2.5,N=15$	256	0.033045097	0.190228076	0.033045098	0.191321545
			512	0.016522549	0.095810941	0.016522549	0.096084698
			1024	0.008261274	0.048079697	0.008261274	0.048148333
4	$Periodic$	$\lambda=1,N=10$	256	0.014640388	0.032499111	0.014648485	0.032731183
			512	0.007332258	0.016176148	0.007324201	0.016239426
			1024	0.003660378	0.008069656	0.003662102	0.008088173

Chapter 5

Experimental Studies and Evaluation

In this chapter, we present the results obtained from a set of experimental studies that explore the effect of different conditions on the performance measures (i.e. $Var[\overline{X}(T)]$, the $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$) of a system. First, to recognize the convergence pattern exhibited by the performance measures, we will examine in Section 5.1 the convergence behavior of $E(X(t))$ and $E[\overline{X}(t)]$ for different initial conditions in a $M/M/1/N$ queue and in a periodic system. Further, our analysis consists of five different investigations, i.e., single server systems, multi-server systems, sequential systems, almost periodic systems and almost decomposable systems. The experimental results obtained by using the analytical method are important for understanding the variation in simulation run length from simultion to simulation.

The single-server systems described in Section 5.2 are $M/M/1$ and $M/E_k/1$. In this section, we reviewed the behavior of a $M/M/1$ queue to investigate the optimal starting condition for a single server system. This will be compared to Madansky's findings in [46], and used to verify Madansky's claim that empty-and-idle state is the optimal initial condition for an $M/M/1$ system. In this section, we also present the results concerning the effect of the variability of the service-time distribution on the performance measures of a system. The effect of other factors such as traffic intensity and number of phases for single server systems is also presented in this section. This will enable us to observe the effect of different factors on the performance measures for single-server systems. In Section 5.3, we explore $M/M/c$ queues for optimal initial conditions. The effect of other factors such traffic intensity, system capacity and number of servers for $M/M/c$ queues is also examined in this section. This will enable us to observe the effect of different factors on the performance measures for multi-server systems. In Section 5.4, we first explore sequential queueing systems or queues in tandem for an optimal initial condition. In addition, the first queue is examined seperately for optimal initial condition and to observe its behaviour in comparison to an $M/M/1$ queue. We also examine the effect of other factors such as traffic intensity, system capacity and number of queues for sequential queueing systems. This will enable us to observe the effect of different factors on the performance measures for sequential queueing systems. In Section 5.5, we investigated an inventory model to observe the effect of periodicity on the performance measures for almost periodic system. The queueing network system described in Section 5.6 will

enable us to observe the effect of degree of decomposability on the performance measures for almost decomposable systems.

5.1 Convergence Pattern of Performance Measures

In this section, we examine different convergence behaviours of the performance measures with the help of an $M/M/1/N$ system and an inventory system. The convergence behaviour of $E[\overline{X}(t)]$ and

Table 5.1: Parameters for Convergence Behaviour of Performance Measure of an $M/M/1/N$ System.

N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	Calculated E(X)	Competing Initial Conditions
14	15	9	10	0.9	$5.1109 \approx 5$	$\{0 = empty/mode, 4 = median,$ $5 = close\ to\ mean, 14 = Full\ System\}$

$E[X(t)]$ for an $M/M/1/N$ queue is shown in Figure 5.1 (a) and (b) respectively. The parameters for the $M/M/1/N$ system are given in Table 5.1. It shows that $E[X(t)]$ converges at a faster rate than $E[\overline{X}(t)]$. Same parameters are used to examine the convergence behaviour of $Var[\overline{X}(t)]$ and $Var[X(t)]$ for an $M/M/1/N$ queue, as shown in Figure 5.2 (a) and (b), respectively. In view of the convergence behaviour shown by $Var[\overline{X}(t)]$ and $Var[X(t)]$ curves in Figure 5.2 (a) and (b), we use time averaging because the $Var[\overline{X}(t)]$ decreases over time to give more precise results. Three possible behavioral patterns observed from Figure 5.1 (a) and (b) over time are as follows:

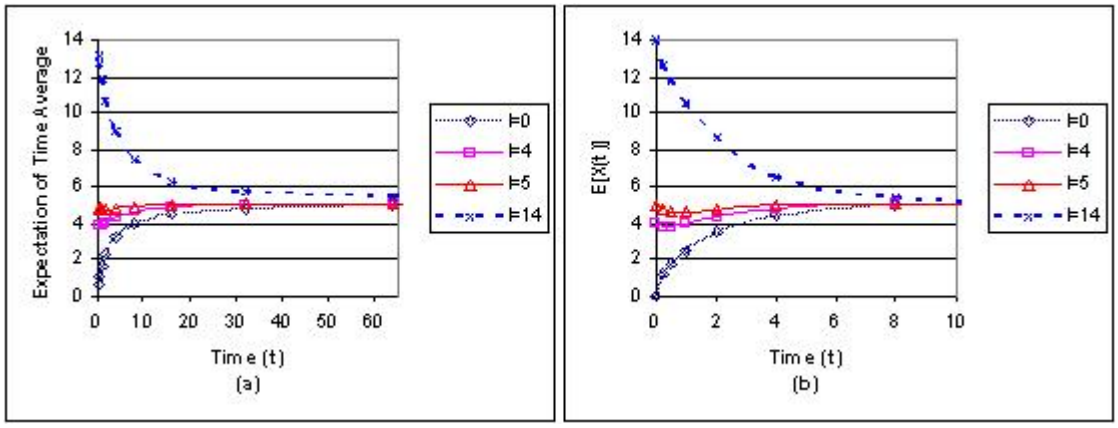


Figure 5.1: Convergence of $E[X(t)]$ and $E[\overline{X}(t)]$ of a $M/M/1/N$ Queue, $\rho = 0.9, N = 14$

1. Monotonically decreasing convergence.

   It is generally observed that if $X(0) > E(X)$, $E[X(t)]$ and $E[\overline{X}(t)]$ are a concave function of T and approach $E(X)$ monotonically from above, e.g. $I = 14$. Since $I > E(X)$ the tendency

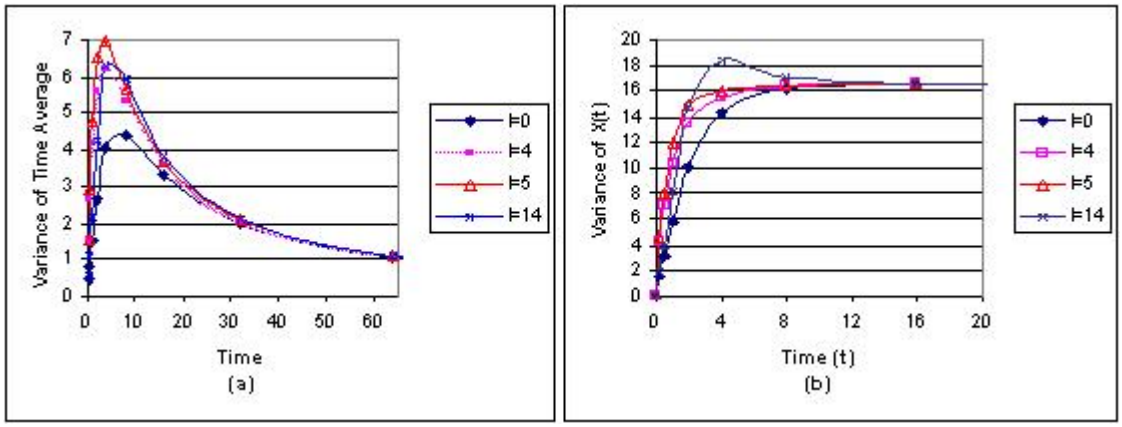


Figure 5.2: Convergence of $Var[X(t)]$ and $Var[\overline{X}(t)]$ of a $M/M/1/N$ Queue, $\rho = 0.9, N = 14$

for the system is to be dominated by difference between the departure and arrival rates $\mu - \lambda$ for some initial period of time. Thus, $E[X(t)]$ and $E[\overline{X}(t)]$ will decrease until they approach $E(X)$.

2. Monotonically increasing convergence.

   If $X(0) < E(X)$, $E[X(t)]$ and $E[\overline{X}(t)]$ are a convex function of T approaching $E(X)$ monotonically from below, e.g., $I = 0$. Since the $E(X)$ of the system has not been reached, the tendency of the system is to be dominated by fewer departures from the system, due to the fact that it is often empty. However, $E(X(T))$ and $E(\overline{X}(T))$ will approach $E(X)$.

3. Non-monotonic convergence.

   If $X(0)$ is close to $E(X)$, the convergence behaviour of $E(X(t))$ and $E(\overline{X}(t))$ is initially influenced by the fact that the system is close to the $E(X)$ at $T = 0$. Since $X(0) \approx E(X) > 0$, the probability for a downward transition (a departure) is more likely than the probability of an upward state transition (i.e. an arrival in the system). Thus, for an initial time period $E[X(t)]$ and $E[\overline{X}(t)]$ decreases for $X(0)$ just below $E(X)$ and then increases monotonically toward $E(X)$ as in the case of $\{I = 4, I = 5\}$ in Figure 5.1.

   Alternatively, if $X(0) > E(X) > 0$ and $\rho > 1$, the convergence behaviour of $E(X(t))$ and $E(\overline{X}(t))$ is initially influenced by the facts that the system starts in $X(0) > E(X)$ at $T = 0$ and the difference between the arrival and departure rates $\lambda - \mu$ for some initial period of time. Since $X(0) > E(X) > 0$ and $\rho > 1$, the probability for an upward transition (an arrival) is more likely than the probability of a downward state transition (i.e. a departure from the system). Thus, for an initial time period $E[X(t)]$ and $E[\overline{X}(t)]$ increases for $X(0) > E(X)$ and then decreases monotonically toward $E(X)$.

Table 5.2: Parameters for Convergence Behaviour of Performance Measure of an Almost Periodic System.

λ	N or *Periodicity*	*#States*	Calculated E(X)	Initial Condition
1	15	15	8.0	$I = 1$
1	20	20	10.5	$I = 1$
1	25	25	13.0	$I = 1$

The selected inventory system represents an almost periodic system. The parameters for examining the convergence behaviour of $E[X(t)]$ and $E[\overline{X}(t)]$ for the inventory system are given in Table 5.2. Figure 5.3 (a) shows the periodic convergence behaviour of $E[X(t)]$, as influenced by increasing periodicities. Figure 5.3 (b) shows that the curve for $E[\overline{X}(t)]$ is more stable than the curve for $E[X(t)]$.

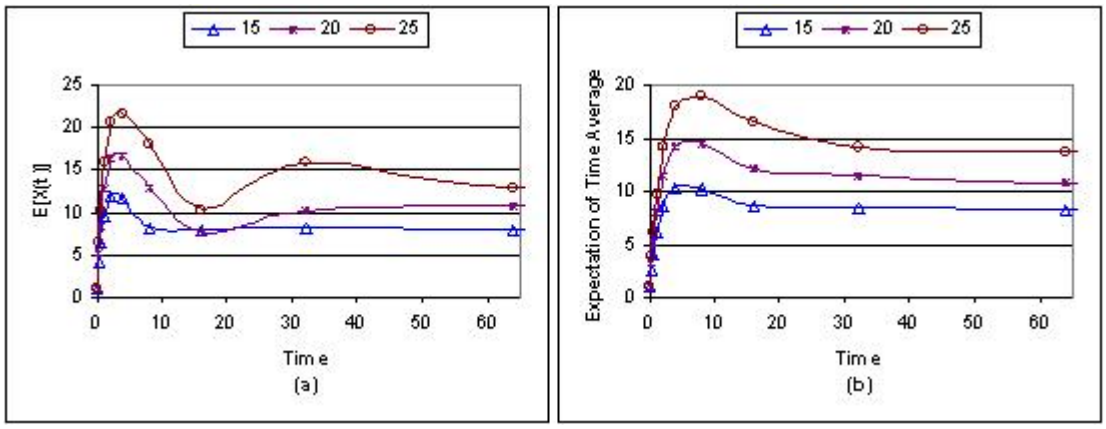


Figure 5.3: Convergence of $E[X(t)]$ and $E[\overline{X}(t)]$ for an Almost Periodic System with $\lambda = 1$.

5.2 Single Server Systems

5.2.1 Optimal Initial Conditions for Single Server Systems

Table 5.3: Parameters for Optimal Initial Condition of an $M/M/1/N$ System

N	*#States*	λ	μ	$\rho = \frac{\lambda}{\mu}$	Calculated E(X)	Competing Initial Conditions
14	15	9	10	0.9	$5.1109 \approx 5$	$\{0 = empty/mode, 4 = median,$ $5 = close\ to\ mean, 14 = Full\ system\}$

We examined a single server system with the parameters given in Table 5.3. It is important to note that the $mode = 0$, $median = 4$ and steady state mean number in the system $E(X) = 5.1109 \approx 5$ remain unchanged for all the experiments in this case. Figure 5.4 (a),(c),(e) show the $Var(\overline{X}(t))$, $Bias(\overline{X}(t))$ and $MSE(\overline{X}(t))$ respectively as a function of $X(0) = i$ (i.e. the effect of starting initial condition) where i may be mean, median or mode of the steady state. Figure 5.4 (b),(d),(f) show the effect of initial conditions from a different perspective. The X axis in Figure 5.4 (b),(d),(f) represents $X(0) = i$ and Y axis gives the $Var(\overline{X}(t))$, $Bias(\overline{X}(t))$ and $MSE(\overline{X}(t))$ respectively. In Figure 5.4 (c) the system was observed to exhibit the minimum bias when started with the initial condition closest to the steady-state mean, i.e., $X(0) = E(X) \approx 5$ and Figure 5.4 (d) shows that for all times the system was observed to exhibit the minimum bias closest to the steady-state mean i.e. $X(0) = E(X) \approx 5$. The bias increases as $X(0)$ moves away from $E(X)$. The plotted value of $Var(\overline{X}(t))$ shown in Figure 5.4 (a) is minimized for the system starting empty-and-idle, i.e., $X(0) = 0$, which is the mode or most frequently visited state. In this case, the variance increases as $X(0)$ moves away from the mode $X(0) = 0$. To resolve the conflicting performance behavior of the bias and the variance, we make use of minimum MSE criterion which takes into account the effects of both measures with required tradeoff and consequently minimizes the initial bias problem. The MSE is smallest for $X(0) = 0$ (i.e. mode) after time $T = 8$. The values of $MSE(\overline{X}(T))$ after $T = 8$ from Figure 5.4 (e) look tied for initial conditions $I = 0, I = 4$ and $I = 5$. We present the computed values of $MSE(\overline{X}(T))$ for $T = 16, 32, 64, 128$ in Table 5.4, based on which we concluded that $X(0) = 0$ is the optimal initial condition. Similarly, we made conclusions for other experiments based on the values obtained from our numerical method. However, the numerical values for other experiments are not presented. Thus, we infer using MSE criterion in $M/M/1$ system that the

Table 5.4: Values of $MSE[\overline{X}(T)]$ for $T = 16, 32, 64, 128$

	16	32	64	128
I=0	3.628203954	2.052343097	1.086187143	0.558097652
I=4	3.633329133	2.053973442	1.086594884	0.558199588
I=5	3.685762674	2.067268042	1.089918613	0.55903052
I=14	5.251155321	2.460736361	1.188286529	0.583622499

MSE is optimized by starting system in empty-and-idle condition, which is the mode. It can be observed from Figure 5.4 (a),(c),(e) that the behavior of MSE curve is initially dominated by the bias of the system and in the long run when bias is negligible the variance dominates the MSE curve. The higher bias in empty starting condition is traded-off for a more desirable lower variance resulting in lower MSE. The earlier findings of Madansky [46] on $M/M/1$ systems is confirmed in Figure 5.4 that MSE is lowest if one starts with $X(0) = 0$. In addition, the results given by Oni

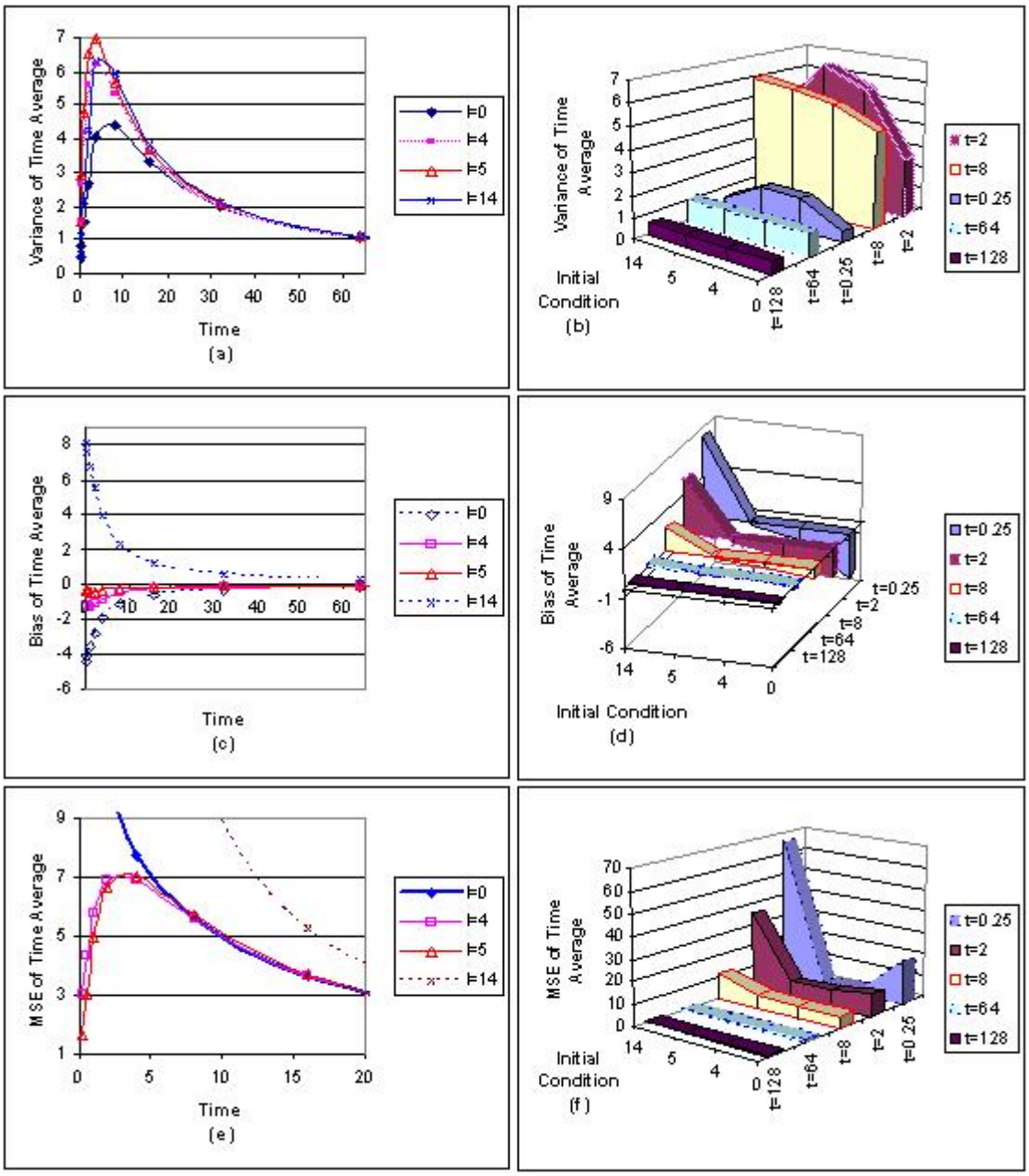


Figure 5.4: Effect of Initial conditions on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$, and $MSE[\overline{X}(T)]$ of an $M/M/1$ queue, $\rho = 0.9$

[52] using MSE criteria, confirm that the optimal initial condition for single servers systems having different service-time distribution (Erlang-k, hyper-exponential) is $X(0) = 0$.

5.2.2 Effect of Traffic Intensities on Single Server Systems

Table 5.5: Parameters for Effect of Traffic Intensities on Single Server Systems, $X(0) = 0$.

$M/M/1/N$ System

N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	E(X)
14	15	1	10	0.1	0.1111
14	15	3	10	0.3	0.4286
14	15	5	10	0.5	0.9995
14	15	7	10	0.7	2.2618

$M/E_k/1/N$ System

k	N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	E(X)
2	13	29	1	10	0.1	0.10833
2	13	29	3	10	0.3	0.3964
2	13	29	5	10	0.5	0.87496
2	13	29	7	10	0.7	1.9077

To show the effect of increasing *traffic intensities* on our measures of performance in an $M/M/1$ system, we plotted Figure 5.5 (a),(c),(e) for the parameters given in Table 5.5. The initial condition is maintained at $X(0) = 0$ for the experiments. ρ can be greater than 1 for finite buffer but $\rho < 1$ for infinite buffer. It is important to note here that for this model $E(X)$ increases with the increase in ρ whereas the mode (0) remains same. The figure shows that $Var[\overline{X}(T)]$ increases with ρ and with the difference between the mode and $E(X)$ because for high values of ρ it is difficult to reach the high probability state (i.e. mode). The $Bias[\overline{X}(T)]$ increases with the difference between $E(X)$ and $X(0)$ because it takes longer to reach $E(X)$. Consequently, $MSE[\overline{X}(T)]$ takes more time to converge for higher values of ρ. The Figure 5.5 (b),(d),(f) also show the same thing from a different perspective. It shows that, as a result of the difference between the initial condition and $E(X)$, the $Bias[\overline{X}(T)]$ is highest at a time close to 0 for all the given traffic intensities and decreases monotonically with time. The $Var[\overline{X}(T)]$ is, however, lowest for times closer to 0 for all given traffic intensities and decreases non-monotonically. As $t \to \infty$, $Var[\overline{X}(T)] \to 0$. Consequently the $MSE[\overline{X}(T)]$, which is dominated by the bias, initially is higher closer to time 0, and decreases monotonically in the long run as a result of the influence of variance. The convergence of the performace measures is slower for higher values of ρ thus indicating that the required length of the simulation run increases with ρ. Figure 5.6 shows three different graphs representing the behaviour of the $Var[\overline{X}(T)]$, the $Bias[\overline{X}(T)]$ and the $MSE[\overline{X}(T)]$ for increasing values of ρ for an $M/E_k/1$ system. The parameters for the $M/E_k/1$ system are given in Table 5.5. The effect of increasing ρ is found similar to an $M/M/1$ system, but the convergence is faster.

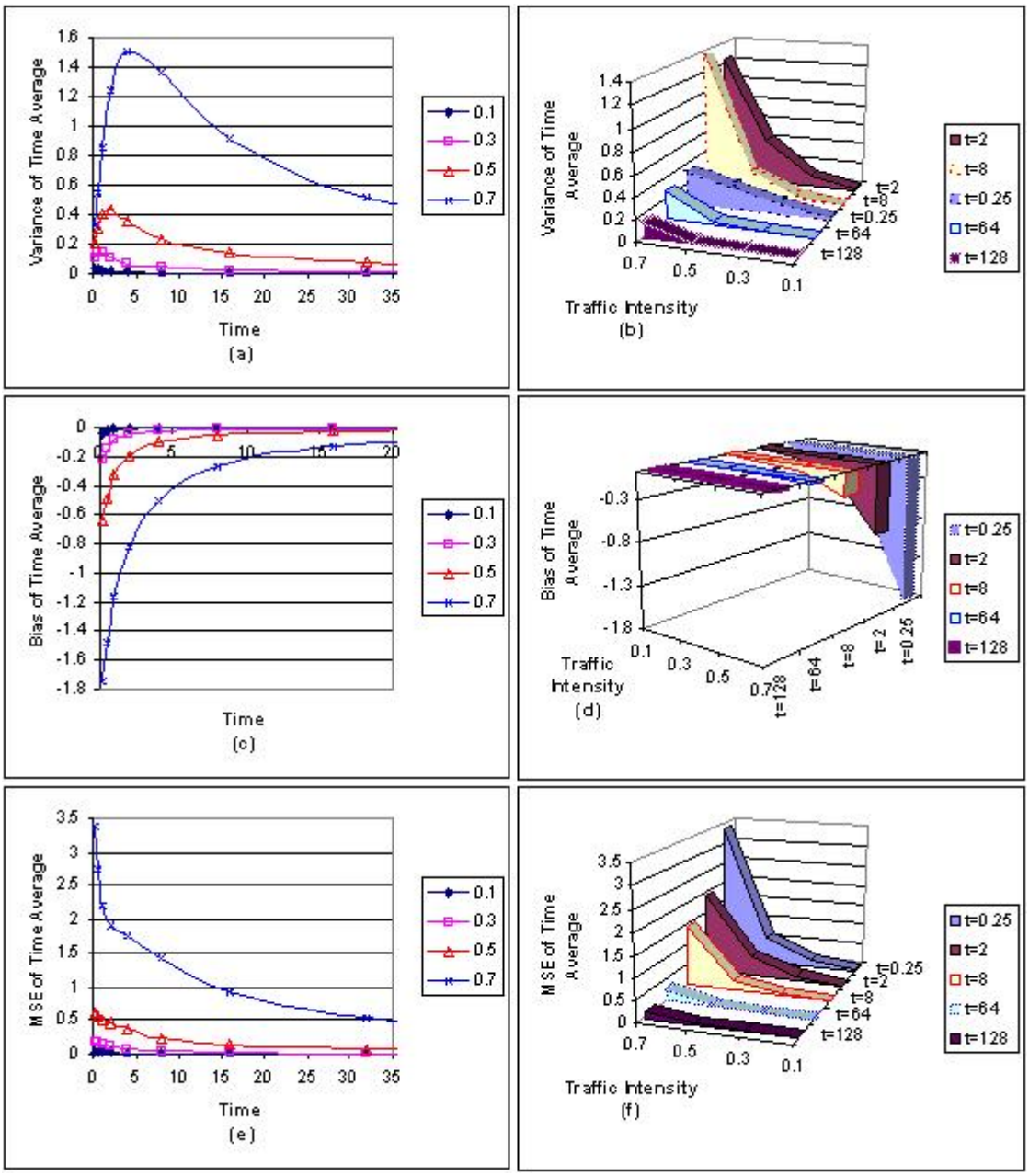


Figure 5.5: Effect of ρ on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of an $M/M/1$ queue

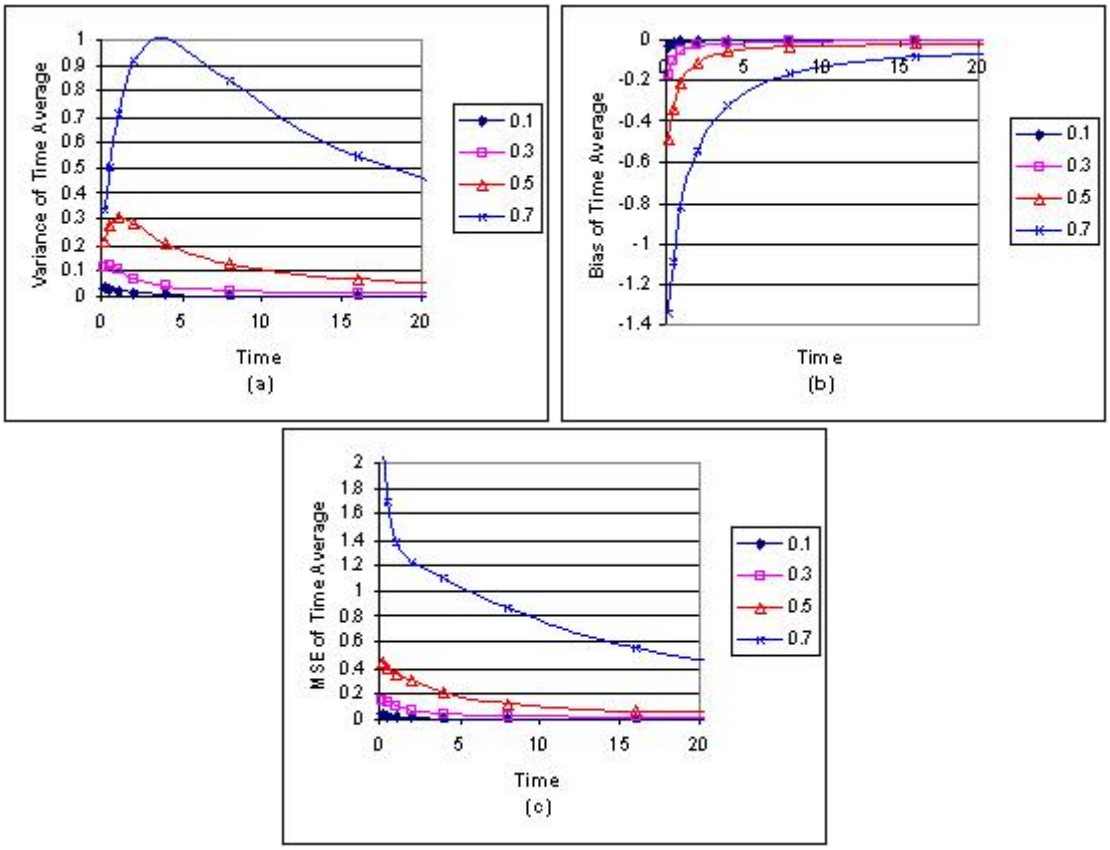


Figure 5.6: Effect of ρ on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of an $M/E_k/1$ queue

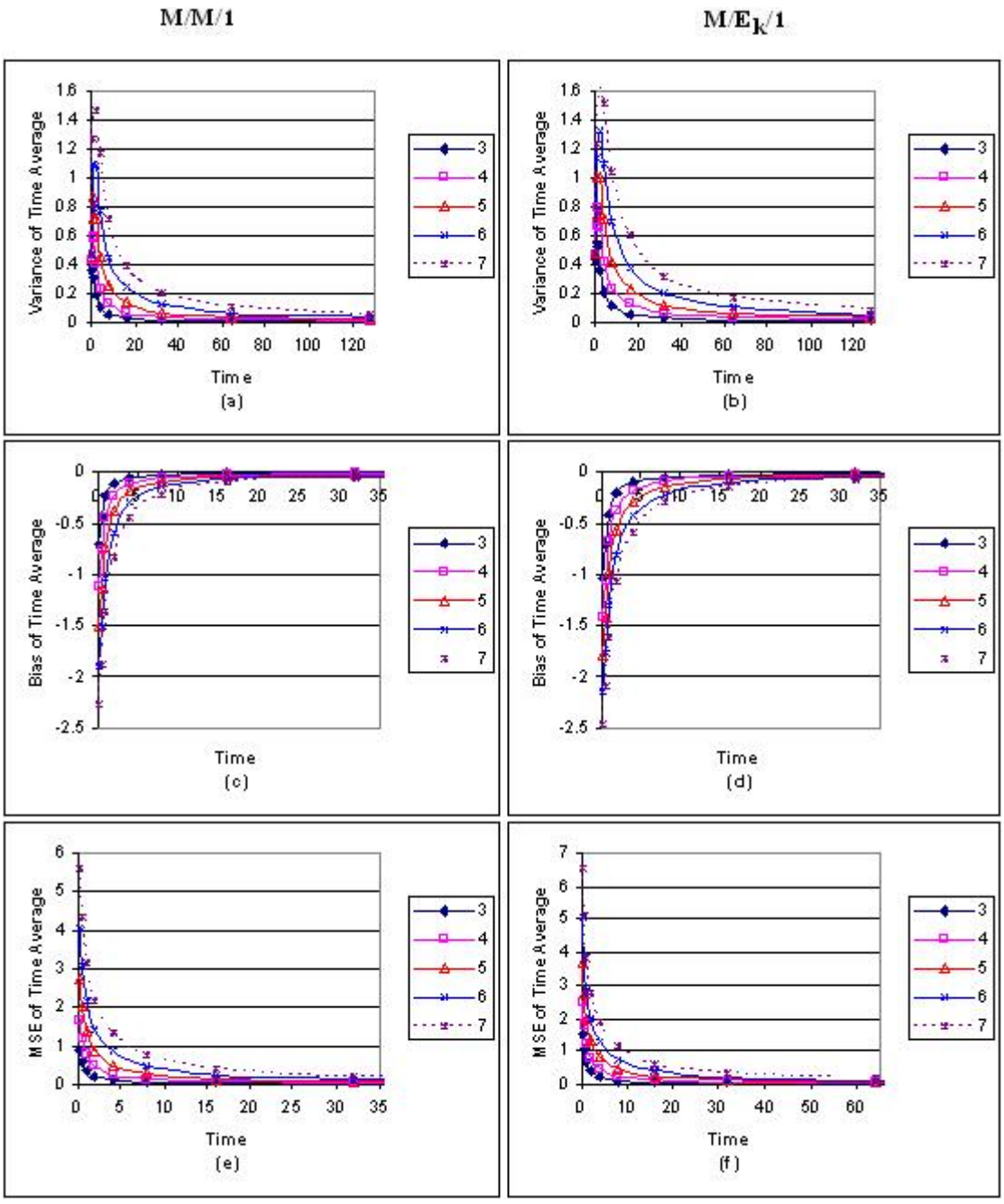


Figure 5.7: Effect of Buffer Size on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of Single Server Systems

Table 5.6: Parameters for Effect of Buffer Size on Single Server Systems, $X(0) = 0$.

$M/M/1/N$ System

N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	E(X)
3	4	9	10	0.9	1.3687
4	5	9	10	0.9	1.7903
5	6	9	10	0.9	2.1948
6	7	9	10	0.9	2.5824
7	8	9	10	0.9	2.9534

$M/E_k/1/N$ System

k	N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	E(X)
2	3	9	9	10	0.9	1.7807
2	4	11	9	10	0.9	2.1722
2	5	13	9	10	0.9	2.5406
2	6	15	9	10	0.9	2.8867
2	7	17	9	10	0.9	3.2112

5.2.3 Effect of Buffer Size on Single Server Systems

To show the effect of increasing *buffer size* on our measures of performance in single server systems, the parameters given in Table 5.6 were used. Figure 5.7 (a),(c),(e) shows the effect of increasing buffer size on an $M/M/1$ system, whereas Figure 5.7 (b),(d),(f) shows the effect of increasing buffer size on an $M/E_k/1$ system. The initial condition is maintained at $X(0) = 0$ for the experiments. It is important to note here that for this model $E(X)$ increases with the increase in buffer size whereas the mode (i.e. 0) remains same. The figure shows that $Var[\overline{X}(T)]$ increases with buffer size and with the difference between the mode and $E(X)$. The $Bias[\overline{X}(T)]$ increases with the difference between $E(X)$ and $X(0)$ because it takes longer to reach $E(X)$. Consequently the $MSE[\overline{X}(T)]$, which is dominated by the bias, initially is higher closer to time 0, and decreases monotonically in the long run as a result of the influence of variance. The convergence of the performace measures is slower for increasing buffer size, thus indicating that the required length of the simulation run increases with buffer size. This time the $M/M/1$ system is observed to be converging faster as compared to the $M/E_k/1$ system.

5.2.4 Effect of Number of Phases on Single Server Systems

Table 5.7: Parameters for Effect of k on an $M/E_k/1/N$ System

k	N	$\#States$	λ	μ	$\rho = \frac{\lambda}{\mu}$	Calculated E(X)	Initial Condition
1	13	15	9	10	0.9	5.1109	$I = 0$
2	13	15	9	10	0.9	4.7519	$I = 0$
3	13	15	9	10	0.9	4.5815	$I = 0$
4	13	15	9	10	0.9	4.4826	$I = 0$
5	13	15	9	10	0.9	4.4181	$I = 0$

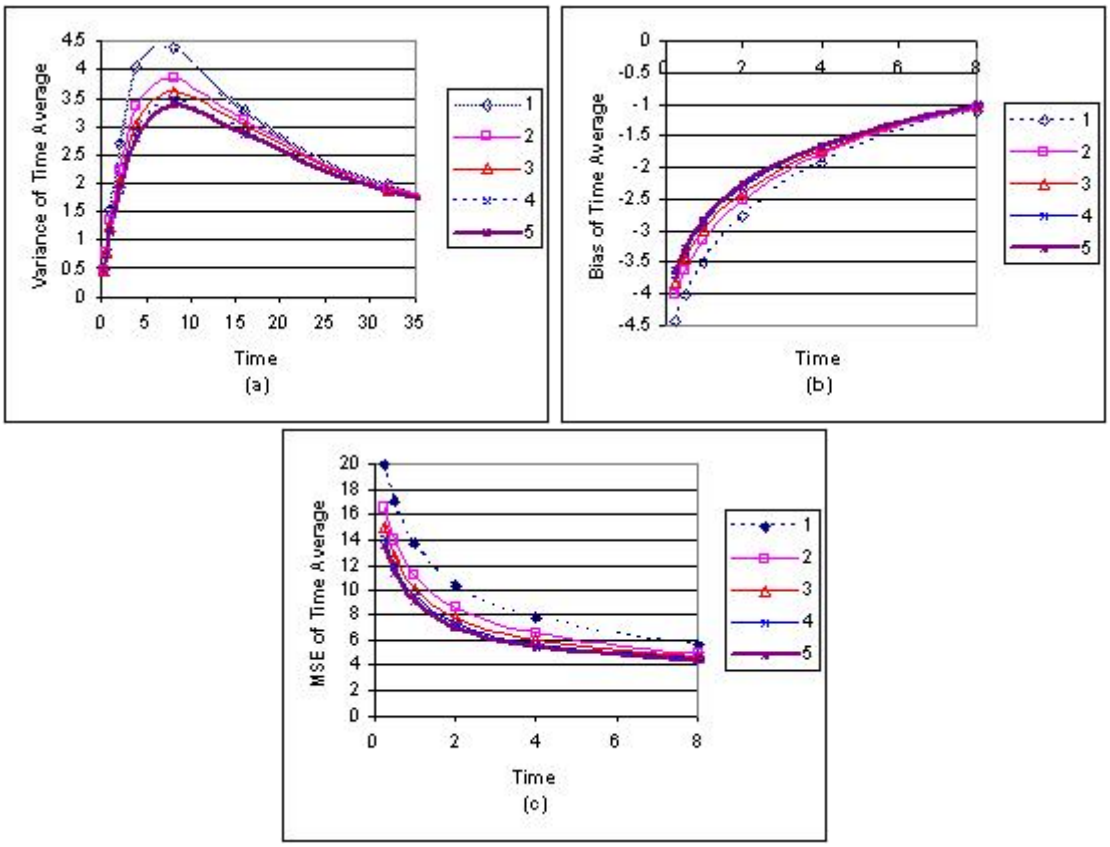


Figure 5.8: Effect of k on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of an $M/E_k/1$ queue

The effect of increasing number of phases for $M/E_k/1$ system, for parameters given in Table 5.7, is depicted in Figure 5.8. It is important to note here that for this model $E(X)$ decreases for increasing values of k whereas the mode (=0) remains same. The difference between the mode and $E(X)$ decreases for increasing values of k and hence the $Var[\overline{X}(T)]$ as shown in Figure 5.8 (a). The $E(X)$ of the system decreases for increasing values of k and hence the $Bias[\overline{X}(T)]$ (see Figure 5.8 (b)). Accordingly, the $MSE[\overline{X}(T)]$ converges faster for increasing values of k.

5.3 Multi-Server Systems

The aim of this section is to explore the optimal initial condition and the convergence pattern exhibited by the performance measures, $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ in $M/M/c$ systems.

5.3.1 Optimal Initial Conditions for $M/M/c$ Queues

Table 5.8: Parameters for Optimal Initial Condition of an $M/M/c/N$ System

c	N	$\#States$	λ	$\mu = \frac{10}{c}$	$\rho = \frac{\lambda}{c\mu}$	Calculated E(X)	Competing Initial Conditions
2	14	15	9	5	0.9	$5.4543 \approx 5$	$\{0 = Empty, 1 = Mode,$ $5 = Median/Close\ to\ Mean,$ $6 = just\ above\ mean,$ $14 = Full\ system\}$
4	14	15	9	2.5	0.9	$6.3189 \approx 6$	$\{0 = Empty, 3 = Mode,$ $6 = Median/Close\ to\ Mean,$ $7 = Just\ above\ mean,$ $14 = Full\ System\}$

In this section, we try to make the results comparable with the results of $M/M/1$ system given in Section 5.2. Therefore, for all the $M/M/c/N$ systems considered we have kept the traffic intensity $\rho = 0.9$ by keeping $\mu = 10/c$ and $\lambda = 9$, where c denotes the number of servers. The paramaters for exploring the optimal initial condition for $M/M/c/N$ systems are given in Table 5.8 and the results are plotted in Figure 5.9. The set of competing initial conditions in this case are $\{0 = Empty, 1 = Mode, 5 = Median\ or\ Close\ to\ Mean, 6 = Just\ above\ mean, 14 = Full\ System\}$ for Figure 5.9 (a),(c),(e); and $\{0 = Empty, 3 = Mode, 6 = Median\ or\ Close\ to\ Mean, 7 = Just\ above\ mean, 14 = Full\ System\}$ for Figure 5.9 (b),(d),(f). The Figure 5.9 (a),(c),(e) shows the effect of starting conditions on $M/M/2$ system with steady state mean $E[X] = 5.4543 \approx 5$, median ($I = 5$), mode

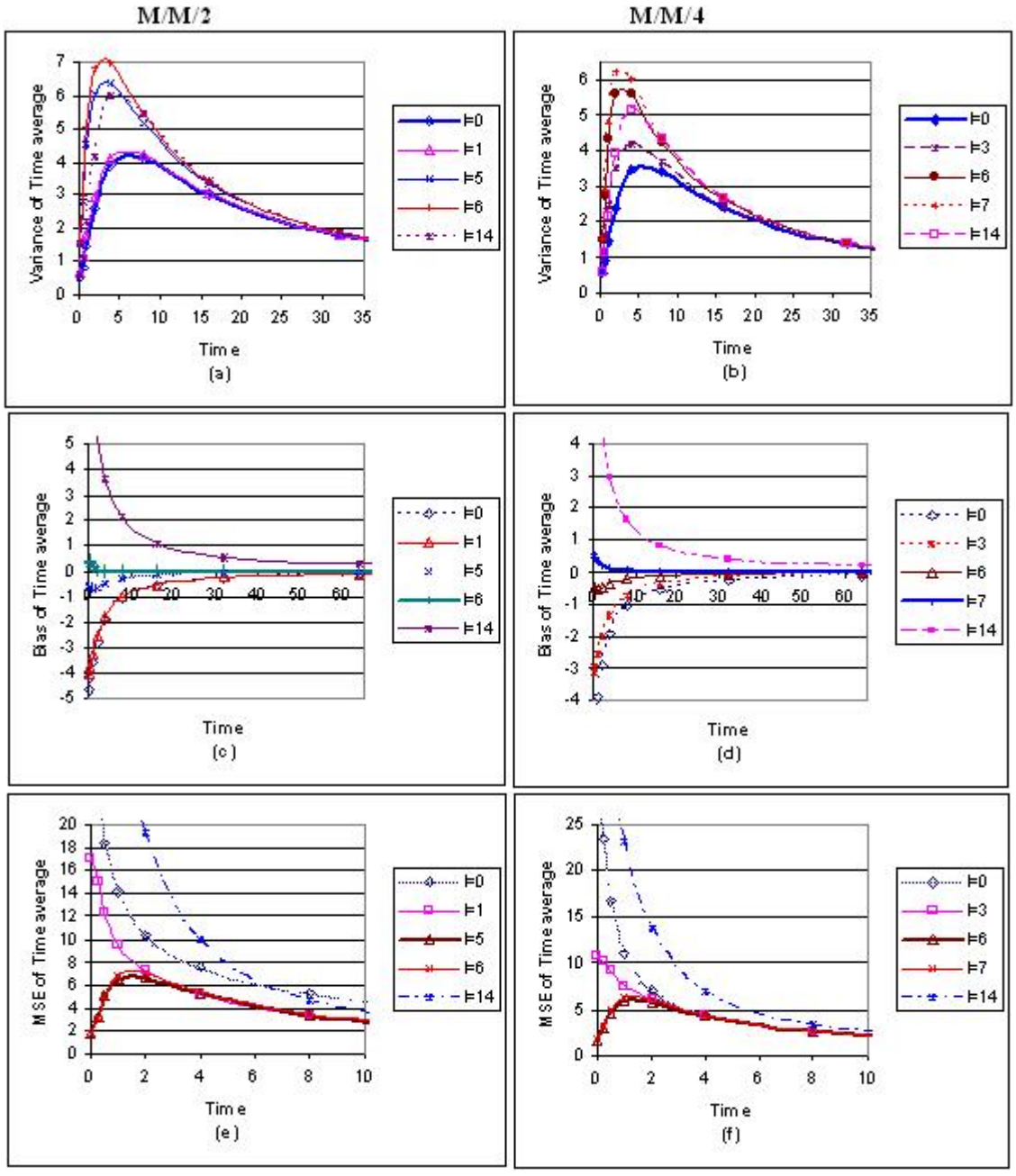


Figure 5.9: Effect of Initial conditions on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of $M/M/2$, $M/M/4$ queues, $\rho = 0.9$

($I = 1$) and full system capacity ($I = 14$), whereas the Figure 5.9 (b),(d),(f) shows the effect of starting conditions on $M/M/4$ system with steady state mean $E[X] = 6.3189 \approx 6$, median ($I = 6$), mode ($I = 3$) and full system capacity ($I = 14$). While initial conditions of $I = 0$ exhibited the smallest variance in Figure 5.9 (a) and (b), a starting condition closest to the mean in Figures 5.9 (c) and (d) showed the smallest bias. The initial condition $I = 14$ in Figures 5.9 (c) and (d) respectively showed the highest bias because it is farthest from respective $E(X)$ for the selected states. Since bias is more important in this case, the MSE is minimized with the system starting close to the steady-state mean. Thus, we conclude that $X(0) \approx E(X)$ is the best initial condition of the conditions examined for $M/M/c$ model when we consider MSE as the measure of the quality of our estimate. Also, as conjectured in Section 3.4, the $M/M/4$ system converges faster than the $M/M/2$ system.

5.3.2 Effect of Traffic Intensity on $M/M/c$ Queue

Table 5.9: Parameters for Effect of Traffic Intensities on an $M/M/2$ Queue

λ	μ	$\rho = \frac{\lambda}{\mu}$	Buffer	N	$\#States$	Calculated E(X)	Initial Condition
1	10	0.1	5	7	8	0.202	$I = 0$
5	10	0.5	5	7	8	1.2932	$I = 0$
7	10	0.7	5	7	8	2.193	$I = 0$
9	10	0.9	5	7	8	3.2376	$I = 0$

The initial condition is not the only factor that determines the convergence pattern, as other factors such as offered workload might also play a role in the convergence behavior of performance measures. Figure 5.10 illustrates the effect of different traffic intensities on the performance measures of a $M/M/c$ queue for parameters given in Table 5.9. The purpose of choosing a smaller buffer size in this experiment as compared to the previous ones is only to create a comprehensible graph. It is important to note here that with the increase in ρ, $E(X)$ also increases, hence the initialization bias. This explains a larger deviation of the $Bias[\overline{X}(T)]$ curves in Figure 5.10 (b) for higher values of ρ and a smaller deviation for $\rho = 0.1$. The behaviour of $Var[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ is observed similar to that of $Bias[\overline{X}(T)]$ in this case. Hence, the simulation run length must be increased with ρ in this case.

5.3.3 Effect of System Capacity on $M/M/c$ Queue

Figure 5.11 depicts the effect of another parameter, the system capacity or buffer size, on the performance measures of a $M/M/c$ queue with two servers. Table 5.10 gives the parameters for

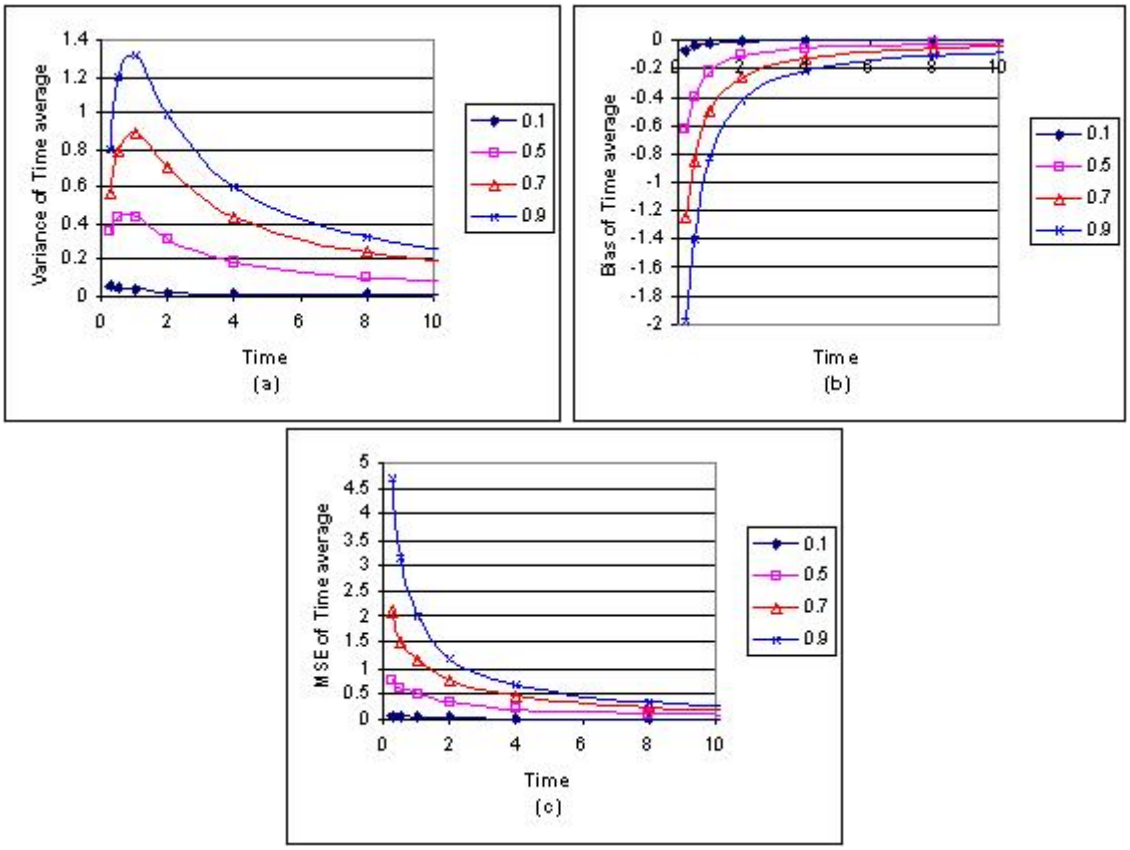


Figure 5.10: Effect of ρ on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of an $M/M/2$ queue

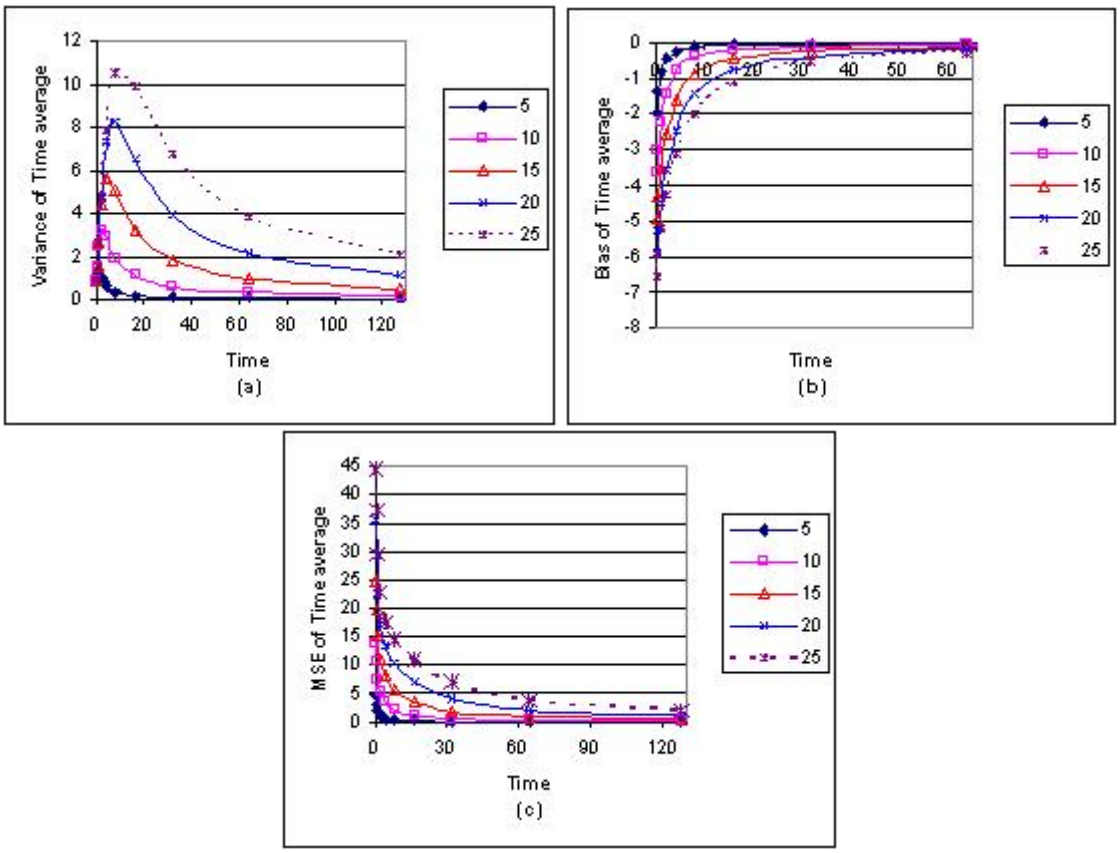


Figure 5.11: Effect of Buffer Size on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ for an $M/M/2$ queue.

Table 5.10: Parameters for Effect of Buffer size on Performance Measures of an $M/M/2$ System

λ	μ	$\rho = \frac{\lambda}{\mu}$	Buffer	N	$\#States$	Calculated E(X)	Initial Condition
9	5	0.9	5	7	8	3.2376	$I = 0$
9	5	0.9	10	12	13	4.8977	$I = 0$
9	5	0.9	15	17	18	6.185	$I = 0$
9	5	0.9	20	22	23	7.1559	$I = 0$
9	5	0.9	25	27	28	7.8689	$I = 0$

this experiment. It can be seen from Table 5.10 that, with the increase in buffer size the $E(X)$ of the system increases. Hence, the $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ increase with buffer size, and so does the simulation run length.

5.3.4 Effect of Increasing Number of Servers in $M/M/c$ Queues

Table 5.11: Parameters for Effect of Servers on Performance Measures of an $M/M/c$ System

c	λ	$\rho = \frac{\lambda}{c\mu}$	N	Calculated E(X)	Initial Condition
2	9	0.9	25	7.6109	$I = 0$
4	9	0.9	25	8.6453	$I = 0$
5	9	0.9	25	9.2108	$I = 0$
10	9	0.9	25	12.1923	$I = 0$

In Figure 5.12, we examine the behavior of a $M/M/c$ queue when the number of servers is increased. The parameters for this experiment are given in Table 5.11. With the increase in the number of servers, the mean/median/mode of system also increase. However, it is important to note here that once the $E(X)$ is reached, with the increase in c it becomes more and more difficult to move away from close to $E(X)$. Hence, as the number of servers is increased, the $Var[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ converge at a faster rate toward respective equilibrium. Consequently, the simulation run length must decrease for systems with more servers.

5.4 Sequential Systems

The aim of this study is to explore the optimal initial condition and the convergence pattern exhibited by the performance measures, $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ in the sequential

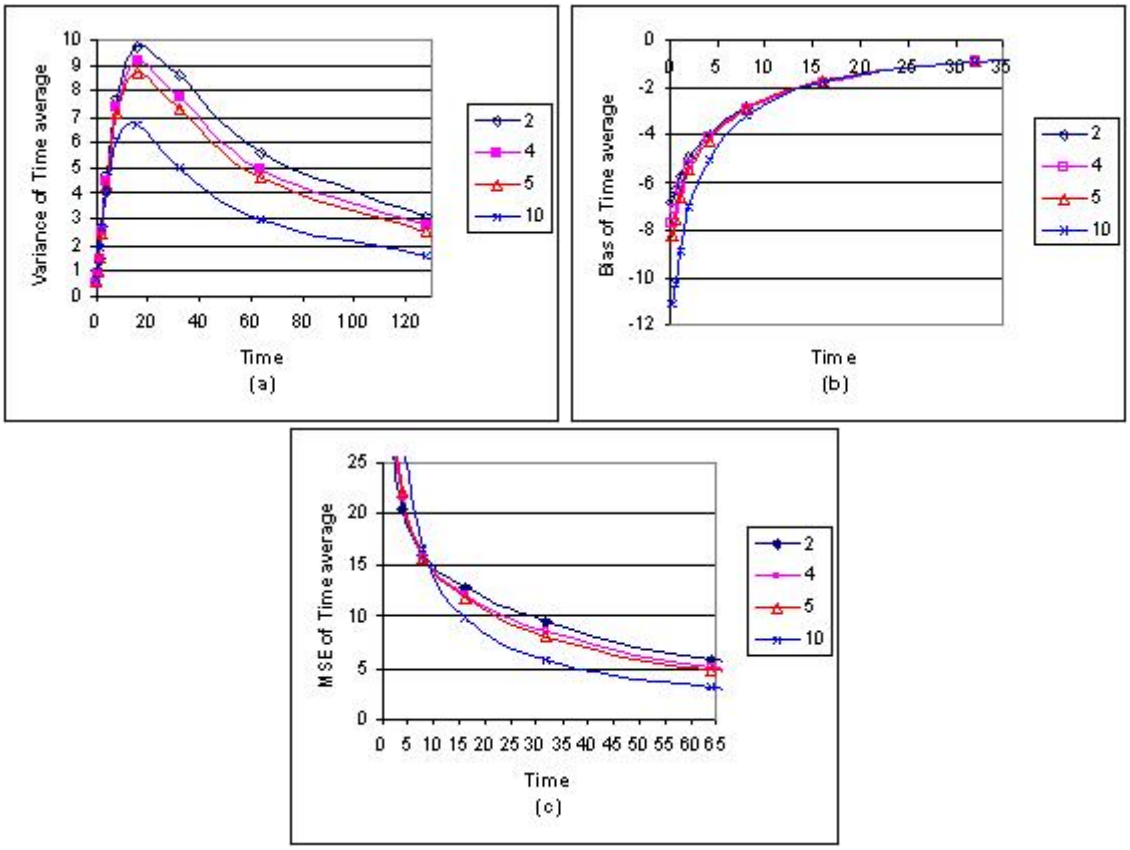


Figure 5.12: Effect of Servers on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of $M/M/c$ queue

queues (with and without blocking). We also investigate the behaviour of the first queue in sequential queueing systems only with blocking because the behaviour of first queue in a sequential queueing system without blocking is identical to that of an $M/M/1$ queue. Here, it is important to discuss the interpretation of different states. The notation $X1, X2$ symbolizes the number of customers in first queue and in second queue respectively for two queues in sequence. Likewise the notation $X1, X2, X3$ represents the number of customers in first, second and in third queue respectively for three queues in sequence. Consequently, $X = X1 + X2$ and $X = X1 + X2 + X3$ represent the total number of customers in sequential queueing system with two and three queues respectively. As a result, the value of X for $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ in different cases is computed differently as follows:

1. For a system having two queues in sequence, $X = X1 + X2$.
2. For a system having three sequential queues, $X = X1 + X2 + X3$.
3. For only the first queue of a system having two or more sequential queues, $X = X1$.

Moreover, a certain number of customers in a system can be represented by more than one state. For example, four customers in a sequential system with two queues can be represented by state $X1, X2 = 2, 2$ or by state $X1, X2 = 4, 0$.

5.4.1 Optimal Initial Conditions for Sequential Queueing System

Table 5.12: Parameters for Optimal Initial Condition for Sequential Queues With Blocking

Two Sequential Queues With Blocking

Fixed Parameters	Initial Condition $X1, X2$
$\lambda = 9$,	$0, 0 = Empty$
$\mu_1 = \mu_2 = 10$,	$2, 2 = Median/Close\ to\ Mean$
$N1 = N2 = 4$,	$4, 0 = Mode/Close\ to\ Mean$
$\#States = 25$,	$4, 4 = Full$
$E(X) = 3.647585$	

Three Sequential Queues With Blocking

Fixed Parameters	Initial Condition $X1, X2, X3$
$\lambda = 9$,	$0, 0, 0 = Empty$
$\mu_1 = \mu_2 = \mu_3 = 10$,	$2, 3, 4 = Median$
$N1 = N2 = N3 = 4$,	$4, 0, 0 = Mode$
$\#States = 125$,	$4, 1, 0 = Close\ to\ Mean$
$E(X) = 5.5433$	$4, 4, 4 = Full$

The effect of initial conditions on the performance measures of sequential queueing systems with blocking is illustrated in Figure 5.13. Figure 5.13 (a),(c),(e) illustrates the effect of starting conditions on a system with two sequential queues with blocking for parameters given in Table 5.12

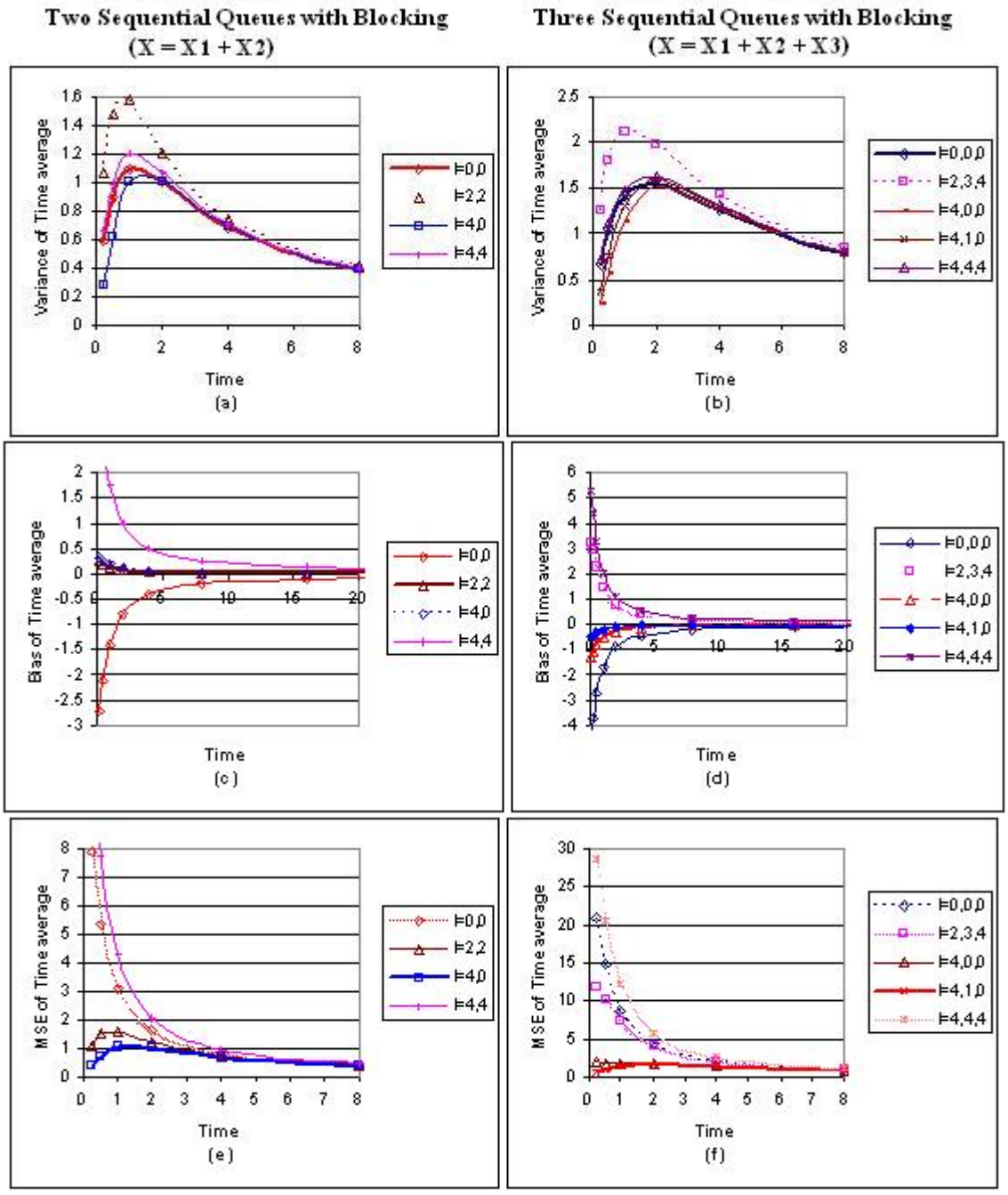


Figure 5.13: Effect of Initial conditions on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Queues with Blocking, $\rho = 0.9$

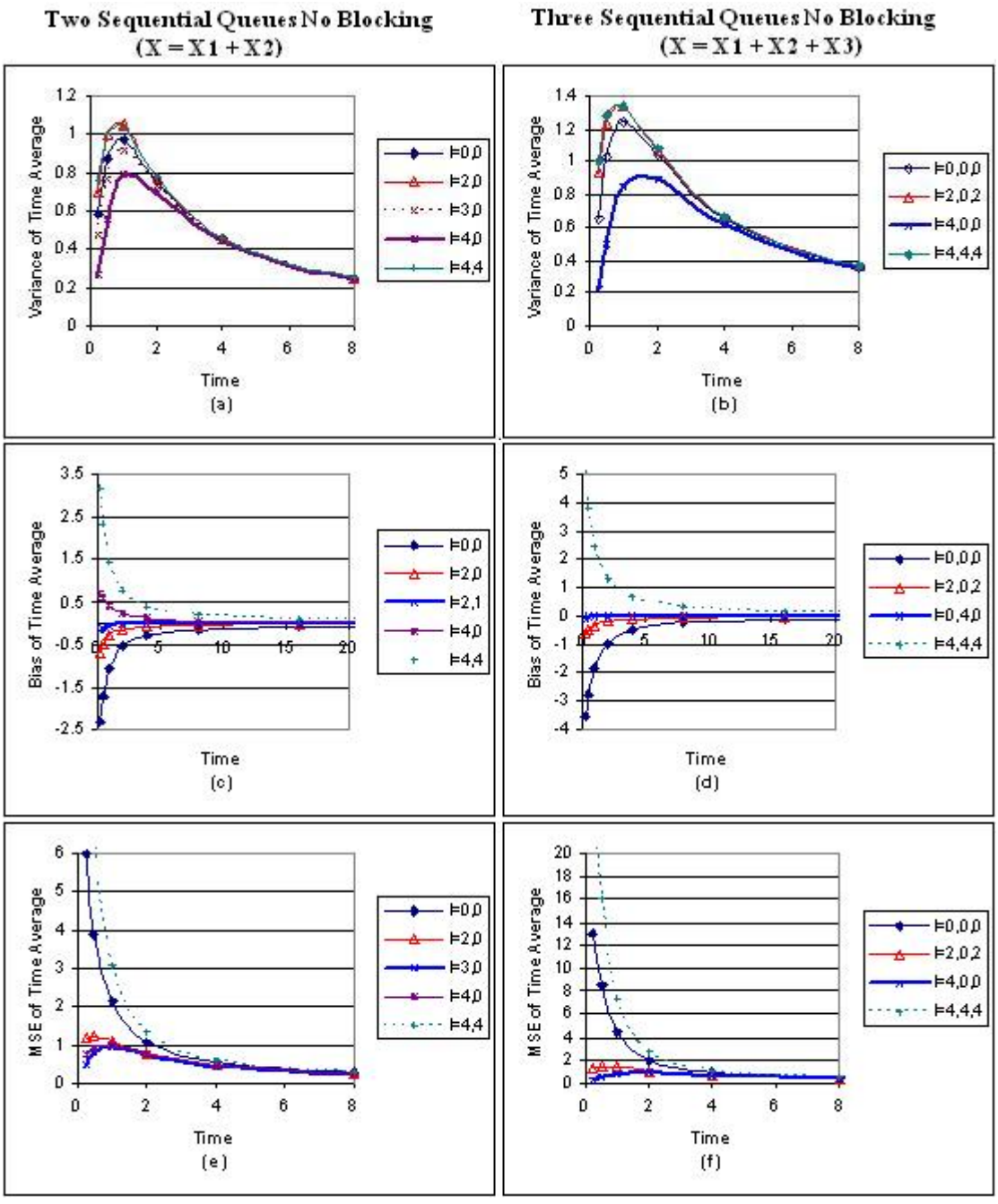


Figure 5.14: Effect of Initial conditions on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Queues without Blocking, $\rho = 0.9$

Table 5.13: Parameters for Optimal Initial Condition for Sequential Queues Without Blocking

Two Sequential Queues Without Blocking

Fixed Parameters	Initial Condition $X1, X2$
$\lambda = 9,$	$0, 0 = Empty$
$\mu_1 = \mu_2 = 10,$	$2, 0 = Median$
$N1 = N2 = 4,$	$2, 1 = Close\ to\ Mean$
$\#States = 25,$	$3, 0 = Close\ to\ Mean$
$E(X) = 3.266993$	$4, 0 = Mode$
	$4, 4 = Full$

Three Sequential Queues Without Blocking

Fixed Parameters	Initial Condition $X1, X2, X3$
$\lambda = 9,$	$0, 0, 0 = Empty$
$\mu_1 = \mu_2 = \mu_3 = 10,$	$0, 4, 0 = Mode$
$N1 = N2 = N3 = 4,$	$2, 0, 2 = Median$
$\#States = 125,$	$4, 0, 0 = Mode$
$E(X) = 4.573585$	$4, 4, 4 = Full$

whereas Figure 5.13 (b),(d),(f) illustrates the effect of starting conditions on a system with three sequential queues with blocking for parameters given in Table 5.12. Here, $X = X1 + X2$ for two sequential queues and $X = X1 + X2 + X3$ for three sequential queues. Among our experiments, the $Bias[\overline{X}(T)]$ is minimized for $X(0) \approx E(X)$ ($X1, X2 = 2, 2$ for two sequential queues and $X1, X2, X3 = 4, 1, 0$ for three sequential queues) and it increases as $X(0)$ moves away from $E(X)$. However, the $Var[\overline{X}(T)]$ is found to be minimized for empty-and-idle starting condition for two and three sequential queues. It is minimized for $X1, X2 = 0, 0$ for two sequential queues and for $X1, X2, X3 = 0, 0, 0$ for three sequential queues. The $MSE[\overline{X}(T)]$ is minimized for the starting condition $X(0) \approx E(X)$ ($X1, X2 = 4, 0$ for two sequential queues and $X1, X2, X3 = 4, 1, 0$ for three sequential queues). Thus, using MSE criteria $X(0) \approx E(X)$ will be considered the most optimal initial condition for sequential servers with blocking.

The effect of initial conditions on the performance measures of sequential queueing systems without blocking is illustrated in Figure 5.14. Figure 5.14 (a),(c),(e) illustrates the effect of starting conditions on a system with two sequential queues without blocking for parameters given in Table 5.13 whereas Figure 5.14 (b),(d),(f) illustrates the effect of starting conditions on a system with three sequential queues without blocking for parameters given in Table 5.13. Here, $X = X1 + X2$ for two sequential queues and $X = X1 + X2 + X3$ for three sequential queues. The $Bias[\overline{X}(T)]$ for two sequential queues is minimized for $X(0) \approx E(X)$ (i.e. $X1, X2 = 2, 1$) and it increases as $X(0)$ moves away from $E(X)$, whereas the $Bias[\overline{X}(T)]$ for three sequential queues is minimized for $X(0) \approx mode$ (i.e. $X1, X2, X3 = 0, 4, 0$) and it increases as $X(0)$ moves away from $mode$. However, the $Var[\overline{X}(T)]$ is found to be minimized for mode in both cases. It is minimized for $X1, X2 = 4, 0$ for two sequential queues and for $X1, X2, X3 = 4, 0, 0$ for three sequential queues. The behaviour of $MSE[\overline{X}(T)]$ in this case is similar to that of $Bias[\overline{X}(T)]$. The $MSE[\overline{X}(T)]$ for two sequential

queues is minimized for $X(0) \approx E(X)$ (i.e. $X1, X2 = 3, 0$), whereas the $MSE[\overline{X}(T)]$ for three sequential queues is minimized for $X(0) \approx mode$ (i.e. $X1, X2, X3 = 4, 0, 0$). Thus, using MSE criteria $X(0) \approx E(X)$ will be considered the most optimal initial condition for two sequential queues without blocking, whereas, for three sequential queue $X(0) \approx mode$ is found to be the optimal initial condition. It is important to note in this case that the *mode* and the $E(X)$ are very close. It is also observed that sequential queueing system without blocking converges faster as compared to the sequential queueing system with blocking.

Table 5.14: Parameters for Optimal Initial Condition for First Queue of Sequential Queueing Systems with Two Queues

First Queue With Blocking

Fixed Parameters	Initial Condition $X1, X2$
$\lambda = 9$,	$0, 0 = Empty$
$\mu_1 = \mu_2 = 10$,	$1, 0$
$N1 = N2 = 4$,	$2, 0 = Close\ to\ Mean$
$\#States = 25$,	$3, 0 = Median$
$E(X) = 2.06823977$	$4, 0 = Mode$
	$4, 4 = Full$

We will now examine the effect of different initial conditions on the performance measures of first queue of a system with two sequential queues with blocking. The parameters are given in Table 5.14 and the results are is shown in Figure 5.15. The behaviour of $Bias[\overline{X}(T)]$ and $Var[\overline{X}(T)]$ is observed similar to an $M/M/1$ system. The $Bias[\overline{X}(T)]$ is minimized for $X1(0) = 2 \approx E(X)$ (here $X1, X2 = 2, 2$) and, the $Var[\overline{X}(T)]$ is minimized for $X1(0) = 4 = mode$ (here $X1, X2 = 4, 4$). However, the $MSE[\overline{X}(T)]$ is minimized for an initial condition, $X1(0) = 3$ (here $X1, X2 = 3, 0$), which is less and close to mode. These results show that the behaviour of first queue in sequential queueing system is very close to that of an $M/M/1$ queue, though not identical.

5.4.2 Effect of Traffic Intensity on Sequential Queueing Systems

Figure 5.16 (a),(c) and (e) illustrate the effect of different traffic intensities on two sequential queues with blocking, for parameters given in Table 5.15, whereas the Figure 5.16 (b),(d) and (f) illustrate the effect of different traffic intensities on the first queue only with blocking. Figure 5.17 (a),(b) and (c) illustrate the effect of different traffic intensities on two sequential queues without blocking for parameters given in Table 5.15. As with the increase in ρ the $E(X)$ of the system also increases

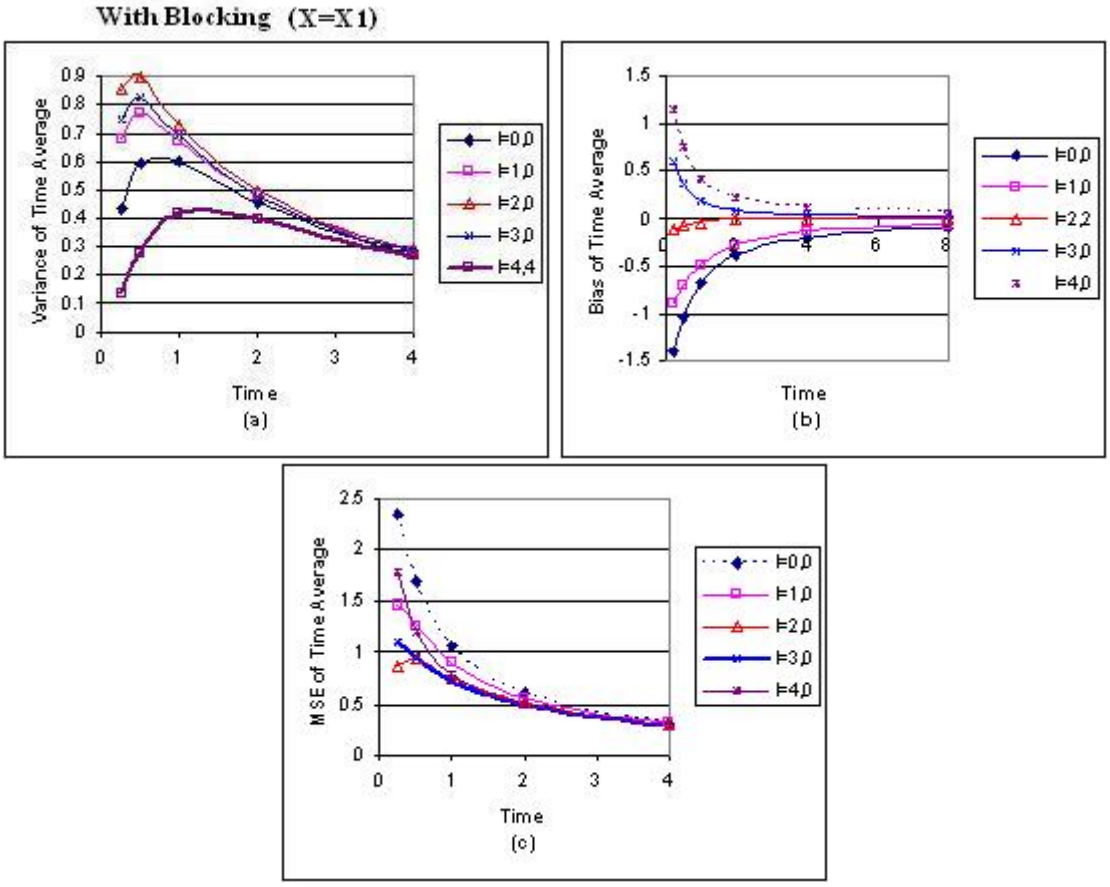


Figure 5.15: Effect of Initial conditions on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ First Queue of Sequential Queues, $\rho = 0.9$

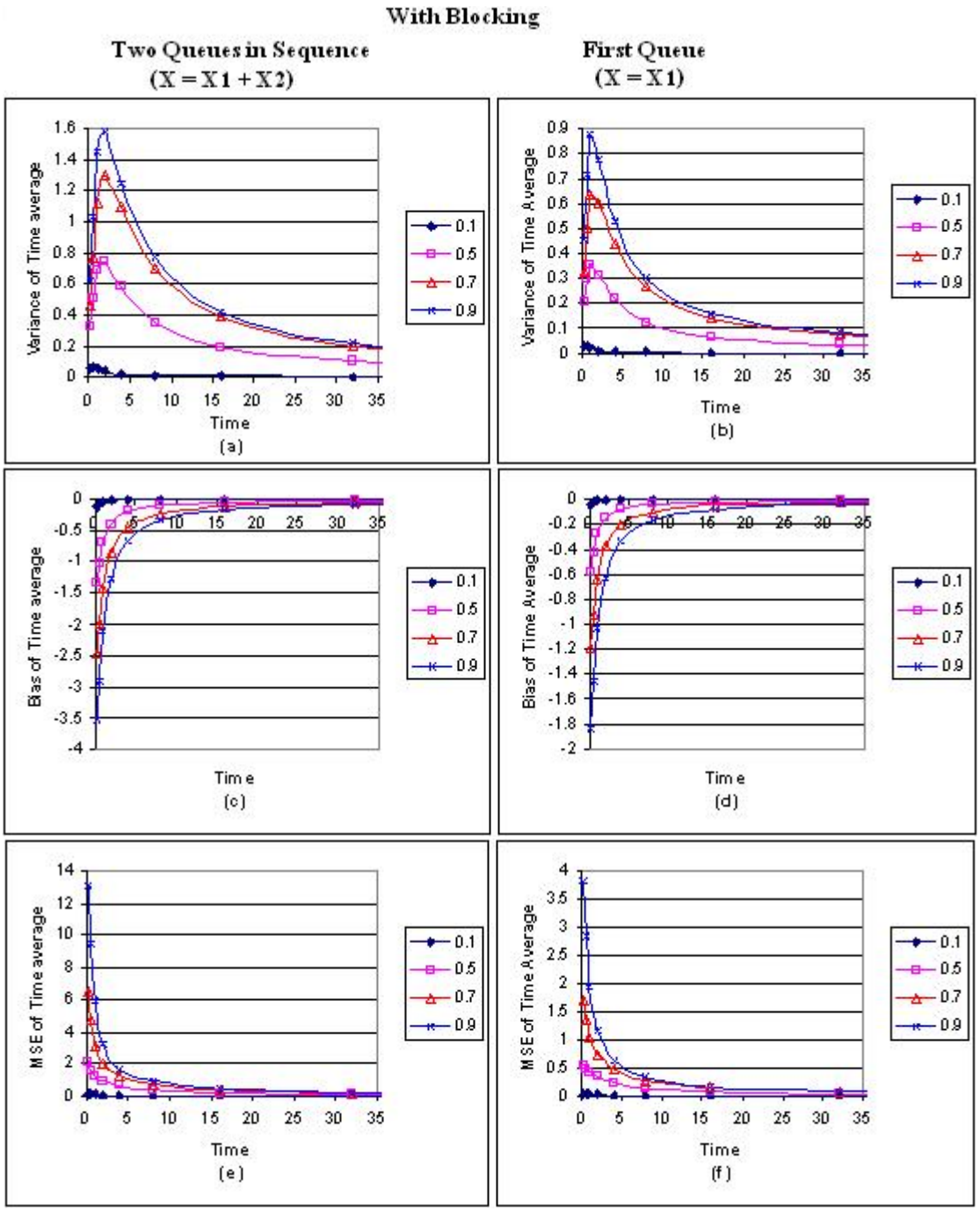


Figure 5.16: Effect of Traffic Intensity on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Queues with Blocking

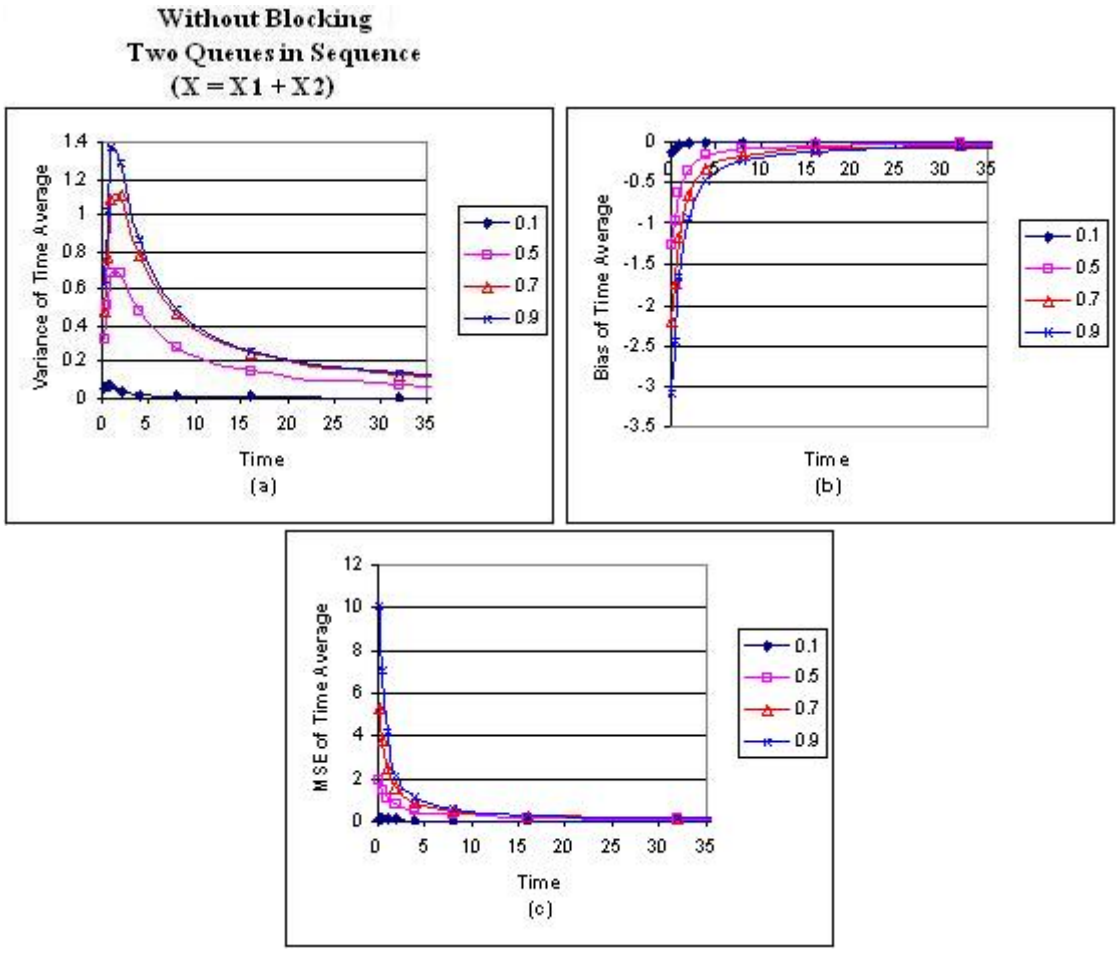


Figure 5.17: Effect of Traffic Intensity on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Queues without Blocking

Table 5.15: Parameters for Effect of Traffic Intensities on Sequential Queueing System

Fixed Parameters	λ	$\rho = \max(\frac{\lambda}{\mu_1}, \frac{\lambda}{\mu_2})$	E(X) Two Queues		E(X) Ist Queue
			Blocking	No Blocking	Blocking
$\mu_1 = \mu_2 = 10$,	1	0.1	0.2222	0.2222	0.1111
$N1 = N2 = 5$,	5	0.5	1.84555	1.782628	0.940298
$I = 0$,	7	0.7	3.18465	2.93118	1.69702
$X1 = 0, X2 = 0$	9	0.9	4.48265	4.031235	2.5194

and the difference between $X(0)$ and $E(X)$ also increases. As a result, the $Bias[\overline{X}(T)]$ is higher for increasing values of ρ. The $Var[\overline{X}(T)]$ is observed lowest for $\rho = 0.1$. The $Var[\overline{X}(T)]$ also increases with ρ. The curve for $MSE[\overline{X}(T)]$ follows the pattern of the curve for $Bias[\overline{X}(T)]$, as $Bias[\overline{X}(T)]$ is more compared to $Var[\overline{X}(T)]$. Consequently, the $MSE[\overline{X}(T)]$ increases with ρ and hence the simulation run length. Similarly, for first queue of sequential queueing system with blocking the $Var[\overline{X}(T)]$, the $Bias[\overline{X}(T)]$ and the $MSE[\overline{X}(T)]$ increase with ρ.

5.4.3 Effect of System Capacity on Sequential Queueing System

Table 5.16: Parameters for Effect of Buffer on Sequential Queueing System

Fixed Parameters	$N1 = N2$	$\#States$	E(X) Two Queues		E(X) Ist Queue
			Blocking	No Blocking	Blocking
$\lambda = 9$,	3	16	2.781	2.47751	1.59487
$\mu_1 = \mu_2 = 10$,	4	25	3.6476	3.26699	2.068
$\rho = \max(\frac{\lambda}{\mu_1}, \frac{\lambda}{\mu_2}) = 0.9$,	5	36	2.51937	4.48265	2.51937
$I = 0$	6	49	5.2862	4.76928	2.949
	7	64	6.0578	5.48073	3.357678

In this section, we examine the effect of increasing system capacity on sequential queueing system by increasing the buffer size for each queue simultaneously. We investigate the sequential queues with and without blocking of the first queue. We also investigate the first queue in case of blocking. The parameters for all the cases are given in Table 5.16. For sequential queueing system with blocking, the effect of increasing buffer size on performance measures of sequential server system with two queues is illustrated in Figure 5.18 (a),(c) and (e). Likewise, Figure 5.18 (b),(d) and (f) shows the effect on performance measures for first queue only. Similarly, the effect of increasing buffer size

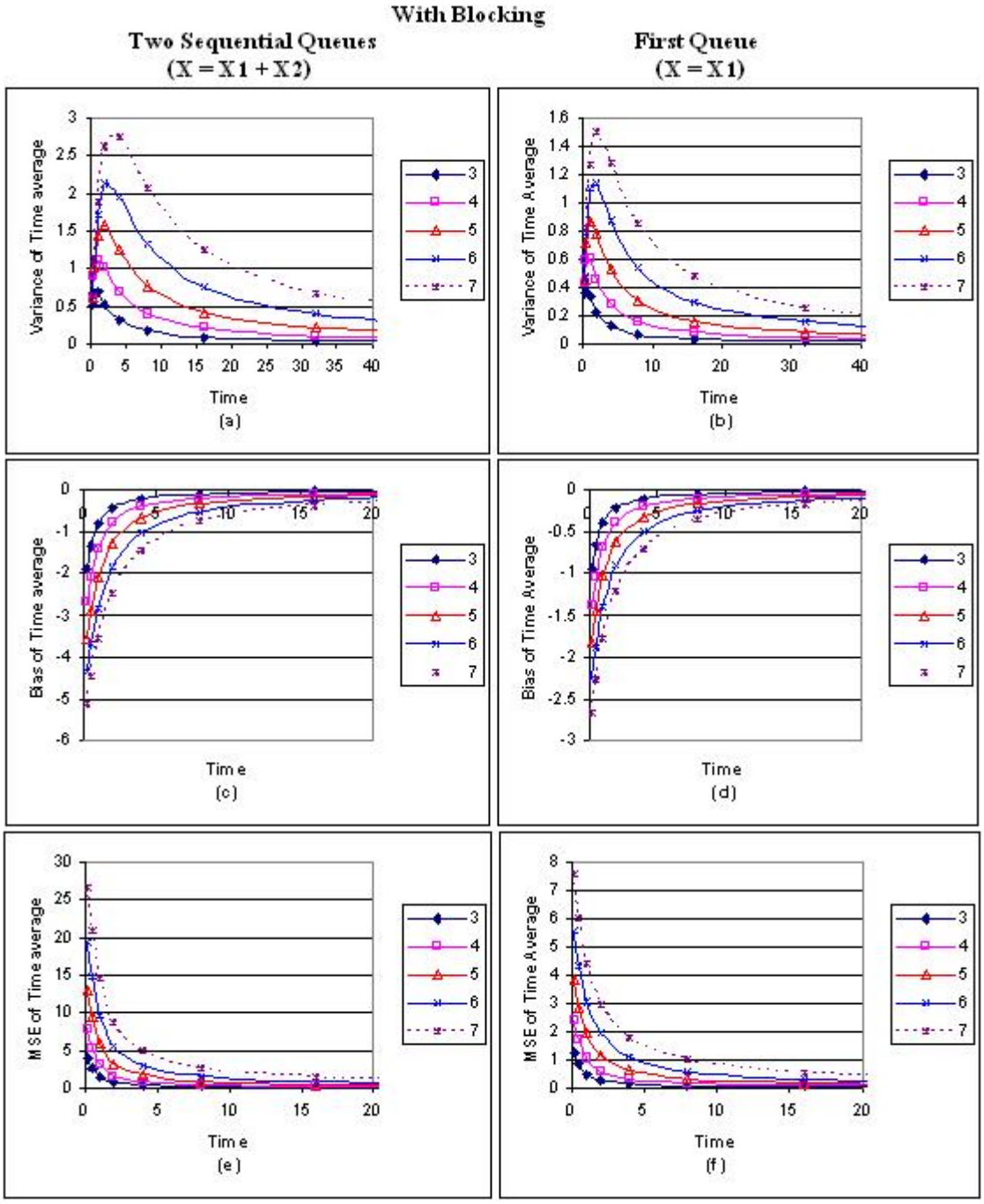


Figure 5.18: Effect of Buffer Size on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Servers with Blocking, $\rho = 0.9$

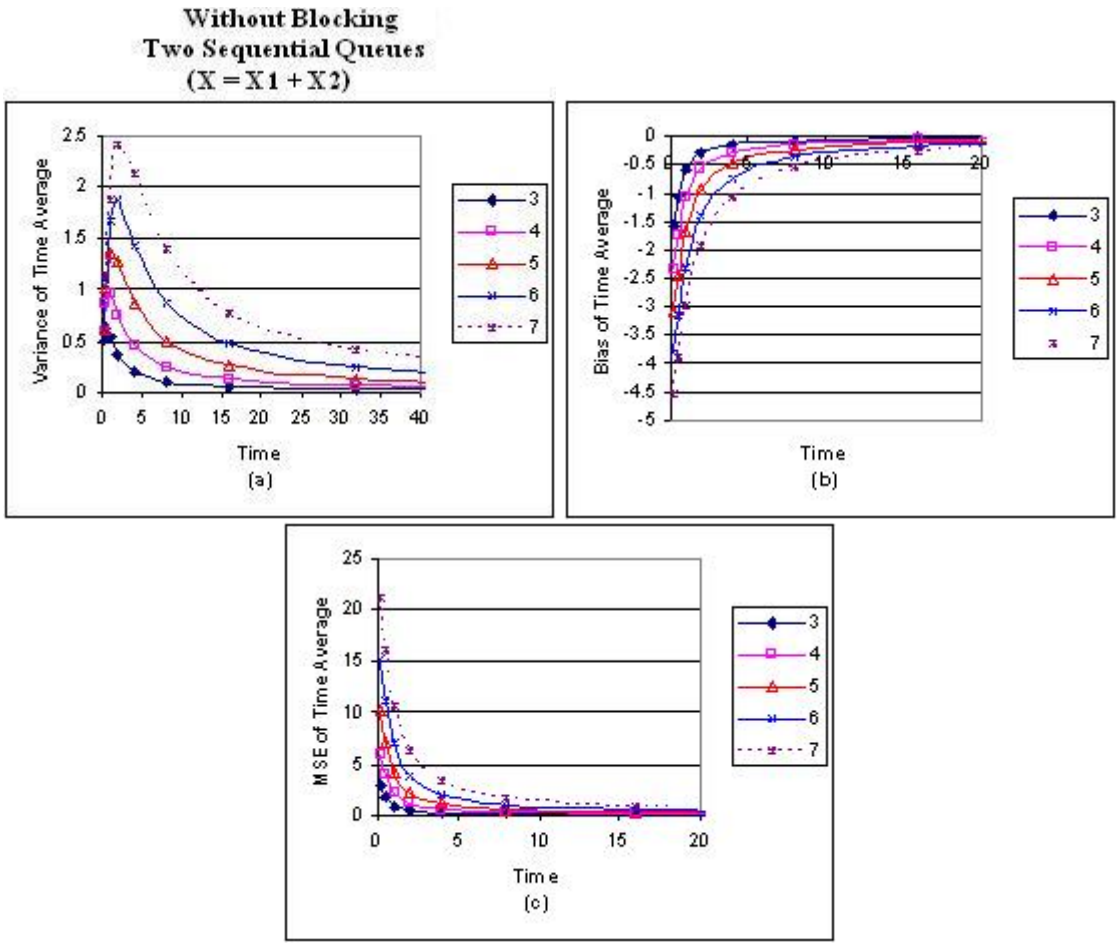


Figure 5.19: Effect of Buffer Size on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential Servers without Blocking, $\rho = 0.9$

on two sequential queues without blocking is shown in Figure 5.19 (a),(b) and (c). Similar to an $M/M/c/N$ system, the effect of increasing buffer size on sequential queues system and first queue is obvious because the size of the sequential queues system increases with the increase in buffer size. The behaviour of first queue is found similar to that of an $M/M/1$ queue or an $M/E_k/1$ queue. As a result, by increasing the buffer size the $E(X)$ of system also increases, which essentially increases the $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ and consequently the simulation run length.

5.4.4 Effect of Increasing Queues In Sequential Queueing System

Table 5.17: Parameters for Effect of Increasing Queues on Sequential Queueing System

Fixed Parameters	$\#Queues$	$\#States$	Calculated E(X)	
			With Blocking	Without Blocking
$\lambda = 9$,	2	9	1.884325	1.665036
$\mu_1 = \mu_2 = \cdots = 10$,	3	27	2.8522	2.29796
$\rho = \max(\frac{\lambda}{\mu_1}, \frac{\lambda}{\mu_2}, \cdots = 0.9)$,	4	81	3.828	2.86472
$N1 = N2 = \cdots = 2, I = 0$	5	243	4.8094	3.3839

The parameters used to examine the effect of increasing number of queues in a sequential queueing system, with and without blocking of first queue, are given in Table 5.17. The $Bias[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ for sequential servers system is shown in Figure 5.20. Figure 5.20 (a), (c) and (e) show the effect of increasing number of queues in a sequential queueing system with blocking, while Figure 5.20 (b), (d) and (f) show the effect of increasing number of queues in a sequential queueing system without blocking. By increasing the numbers of queues the $E(X)$ of the system also increases, therefore increasing $Bias[\overline{X}(T)]$, $Var[\overline{X}(T)]$, $MSE[\overline{X}(T)]$ and the length of a simulation run.

5.5 Almost Periodic Systems

The effect of periodicity is illustrated in Figure 5.21 by using an inventory system. The parameters to examine the effect of periodicity on an inventory system are given in Table 5.18. Figure 5.21 (a), (b) and (c) show the curves for $Bias[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ respectively for different reorder quantities ranging from 5 to 25 in steps of 5. The initial condition of $X(0) = 1$ is maintained for all reorder quantities. The $E(X)$ of system increases with reorder quantities (or periodicity). As a result, the $Bias[\overline{X}(T)]$ takes longer to converge for increasing reorder quantities (or periodicity). The influence of periodicity can be observed in the $Bias[\overline{X}(T)]$ curves by their

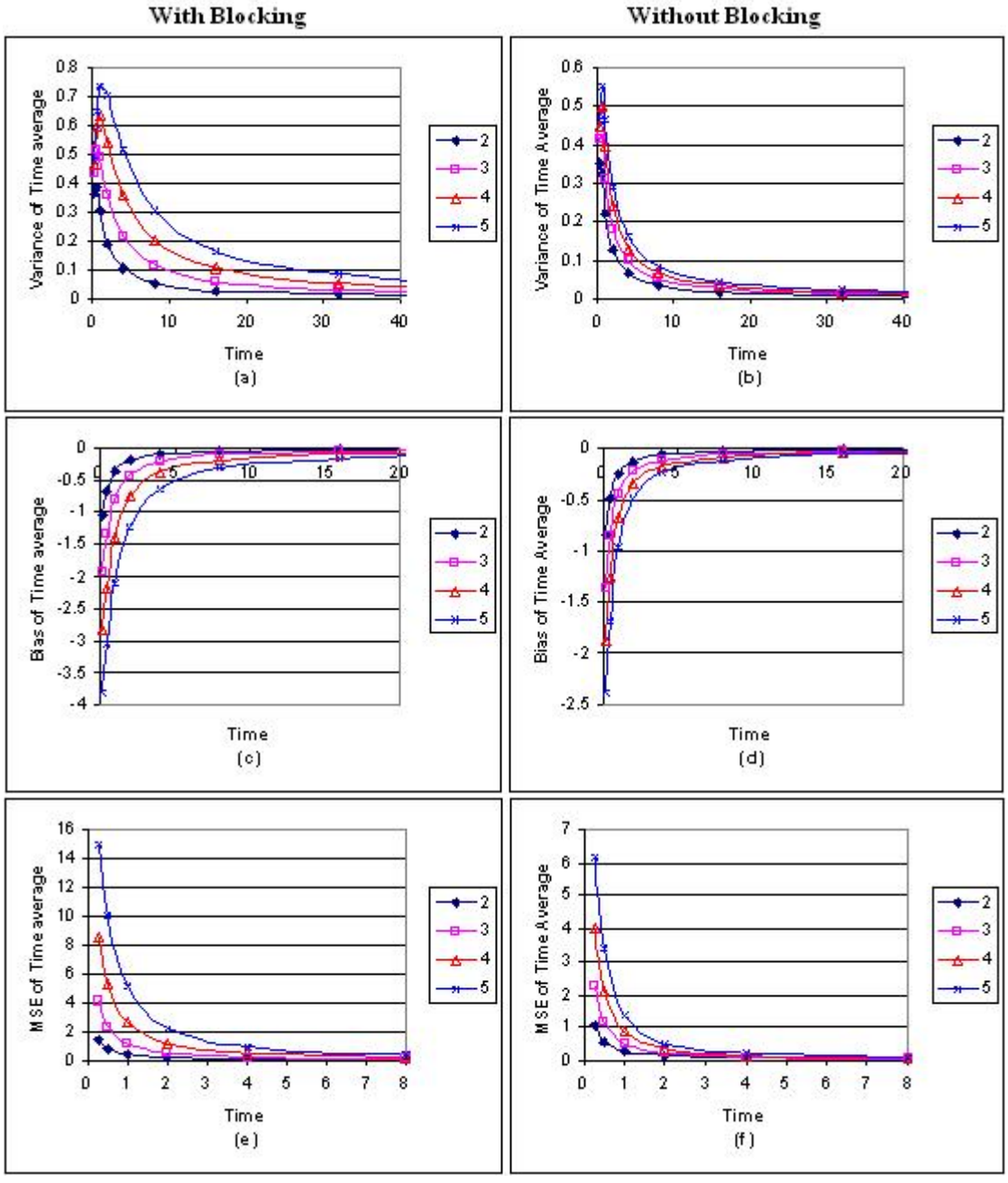


Figure 5.20: Effect of Servers on $MSE[\overline{X}(T)]$, $Var[\overline{X}(T)]$ and $Bias[\overline{X}(T)]$ of Sequential queue, $\rho = 0.9$

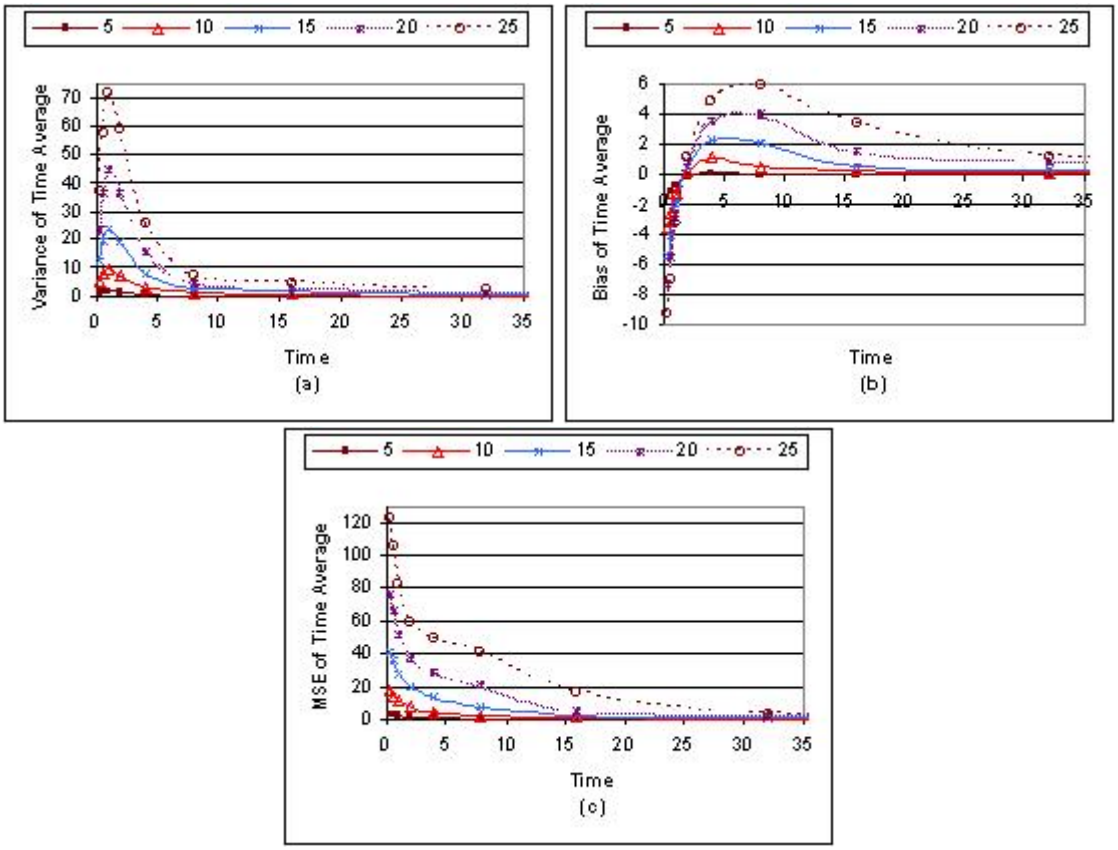


Figure 5.21: Effect of Periodicity on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of an Almost Periodic System, $X(0) = 1$

Table 5.18: Parameters for Effect of Periodicity on an Inventory System

Fixed Parameters	$\#Items$	$\#States$	Calculated E(X)
$\lambda = 1, I = 1$	5	5	3.0
	10	10	5.5
	15	15	8.0
	20	20	10.5
	25	25	13.0

turn from negative to positive. Any state (including $E(X)$) in an almost periodic system is a mode state, as each state has equal probability of being visited. Therefore, larger the reorder quantity (or periodicity) is, the larger is the variance. Hence, the $Var[\overline{X}(T)]$ takes longer to converge with increasing periodicity. The variance curve also appear to have some sort of periodicity. Furthermore, $MSE[\overline{X}(T)]$ takes longer to converge for increasing values of reorder quantities (or periodicity). The effect of periodicity observed in the curves for the $Bias[\overline{X}(T)]$ and the $Var[\overline{X}(T)]$ is also observed in the curves for the $MSE[\overline{X}(T)]$.

5.6 Queueing Network Systems

Table 5.19: Parameters for Effect of Degree of Decomposability on a Closed Queueing Network System

Case I (Almost Decomposable System)

Fixed Parameters	$\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$	Calculated E(X)
$\lambda_{12} = \lambda_{21} = 10,$	1	3.3333
$N1 = N2 = N3 = 5,$	0.5	3.3333
$X1, X2, X3 = 5, 0, 0$	0.1	3.3333

Case II (Semi-decomposable System)

Fixed Parameters	$\lambda_{31} = \lambda_{32}$	Calculated E(X)
$\lambda_{12} = \lambda_{21} = 10,$	1	3.3333
$\lambda_{13} = \lambda_{23} = 1,$	0.5	1.75
$N1 = N2 = N3 = 5,$	0.1	0.222
$X1, X2, X3 = 5, 0, 0$		

The effect of degree of decomposability on a closed queueing network system is shown in Figure 5.22. The figure gives the curves for $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ for different transitions rates to and from queue 3. The parameters for this experiment are given in Table 5.19. The initial state for the experiments is $X1, X2, X3 = 5, 0, 0$ i.e., the system starts with all the customers in first queue. However, we are only interested in the number of customers in

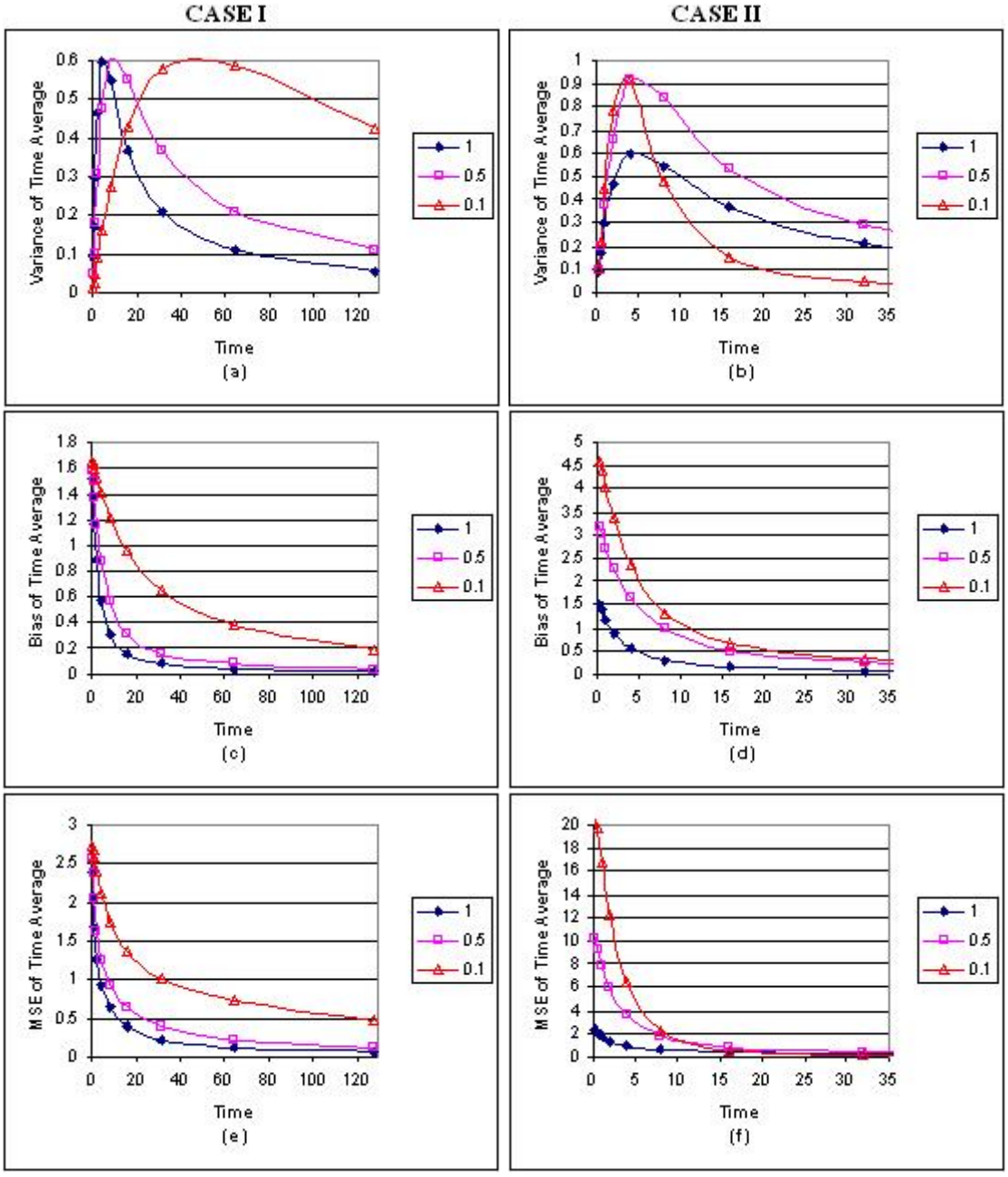


Figure 5.22: Effect of Decomposability on $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ of a Queueing Network System

queue 1 and in queue 2 combined i.e. $X = X1 + X2$. As a result, $E(X)$ is $E(X1) + E(X2)$ or $5 - E(X3)$. We are dealing with two cases in this experiment. In the first case (see Table 5.19), $\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$ varies from $0.1, \ldots, 1$. Other parameters are $\lambda_{12} = \lambda_{21} = 10$ and $N = 5$. The curves for $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ for the first case are shown in Figure 5.22 (a),(c) and (e). Since the transition rates of going back and forth between queue 3 and other queues are same, the $E(X)$ of system for different rates will remain same. As a result, the $Bias[\overline{X}(T)]$ values for different conditions, close to $t = 0$ (see 5.22 (c)), are close to each other. For decreasing transition rates, more and more customers tend to remain in queue 3 (i.e. $E(X3)$ increases) at a nearly constant rate, thus increasing the decomposability and $Bias[\overline{X}(T)]$ over time at nearly constant rate. As the transition rates of going back and forth between queue 3 and other queues are same (i.e. $\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$) in this case, the $E[\overline{X}(T)]$ for higher transition rates will tend to remain more close to $E(X)$ (here $X = X1 + X2$) than for lower transition rates. Consequently, the $Bias[\overline{X}(T)]$ is smaller for higher transition rate when the decomposability is less. That is to say, when $\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$ decreases the bias increases. For the same reason, similar behaviour is observed in the curves for $Var[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$. In the first case we observed that $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ are minimized for lower degree of decomposability and increase with the degree of decomposability at a near constant rate.

The curves for $Var[\overline{X}(T)]$, $Bias[\overline{X}(T)]$ and $MSE[\overline{X}(T)]$ for the second case are shown in Figure 5.22 (b),(d) and (f). The parameters for the second case (see Table 5.19) are $\lambda_{12} = \lambda_{21} = 10$ and $N = 5$. The transitions rates $\lambda_{13} = \lambda_{23} = 1$ whereas $\lambda_{31} = \lambda_{32}$ vary from $0.1, \ldots, 1$. While the rate of arrival to Queue 3 from Queue 1 and Queue 2 is constant, only the rate of departure from Queue 3 to Queue 1 and Queue 2 varies. Therefore, the decomposability of the system increases with the decreasing transition rates λ_{31} and λ_{32} while $\lambda_{13} = \lambda_{23} = 1$ remain unchanged. In this case the $E(X)$ (here $X = X1 + X2$) decreases with increase in the degree of decomposability. As a result, the bias for different conditions close to $t = 0$ are different. The $Bias[\overline{X}(T)]$ close to $t = 0$ is least for $\lambda_{31} = \lambda_{32} = 1$ and is highest for $\lambda_{31} = \lambda_{32} = 0.1$. Once a customer arrives in Queue 3, for lower rate of departure, more and more customers tend to remain in Queue 3. As, a result the variance increase for decreasing rates. The variance, however, is least for highest degree of decomposability because the $E(X)$ in this case is so low that the variation around $E(X)$ would also be low. As opposed to our previous case, the $MSE[\overline{X}(T)]$ in this case is minimized for a lowest departure rate from Queue 3, i.e., highest decomposability.

Chapter 6

Conclusions and Future Research

6.1 What We Did

We used time averages over the period of a simulation as an estimator. The precision of an estimate is of concern when estimating the expectation. An informative estimator must be accompanied by confidence bounds. This question is usually answered by attempting to estimate the standard error of the estimator. The bias estimates the expected difference between a parameter and its estimator while the variance determines the precision of the estimator. Moreover, the distribution of an estimator approaches a normal distribution, and we need the expectation and the variance to define a normal distribution. The assumption here is that the asymptotic variance is finite. As a result, the bias and the variance of an estimator naturally lead to such confidence bounds. Thus, bias and variance are a fundamental feature of an error estimator. In addition, to measure the deviation and dispersion around the true value of the parameter, the MSE combines the effect of the bias and the variance. Therefore, three performance measures, the bias, the variance and the MSE of the time average, are used to measure the quality of an estimator. Wilson and Pritsker [71] also used these three measure to evaluate the startup policies in simulation experiments.

We use a Markovian Event System for our research because it is simple. As the basis to support our line of reasoning for the experiments, we discussed and analyzed the transient behavior of estimators. The transient characterstics examined are useful for studying the finite time properties of systems represented by such models. We selected models of various stochastic systems for close investigation. We built the Markovian models of the systems to be evaluated. The criteria for selecting the models are to lend support to our conjecture that if there are states which greatly influence the observed parameters, these states must be reached soon, that is, after few events and/or with high rate events. Otherwise, the covariance, the bias and the variance all increase. The models selected for close investigation along with their properties and the reason for selection are as follows:

1. The $M/M/1/N$ model is the simplest model.

2. An $M/M/c$ model is selected to examine the behaviour of performance measure in multi-

server systems. Another reason for using this model is to study the faster convergence versus slower convergence depending on the rates of two matrices.

3. The Sequential Queueing Systems with 2 and 3 queues are selected, as they will require 2 and 3 state variables. In addition, the first queue of a sequential system is also examined independently for comparison with an $M/M/1$ queue.

4. An $M/E_k/1/N$ model is selected to investigate the impact of variability of the service-time distribution on the performance measures of single-server systems. Moreover, this single server system requires 2 state variables.

5. An inventory system is selected to examine the effect of periodicity on the bias, the variance and the MSE of time average for almost periodic systems.

6. A queueing network model is selected to experiment with almost decomposable systems to explore the effect of degree of decomposability on the bias, the variance and the MSE of time average.

We calculated the essential statistics such as expectation, variance, bias and MSE for the selected systems. For this purpose we used the algorithm given in [27].

6.2 Summary of Thesis Results

A frequent objective of a simulation is to find the expected rewards $E(X)$ per time unit in equilibrium given the rewards in state i is $r(i)$ per time unit. In this research study, we focussed on the following performance measures the variance, the bias and the MSE of the time average (denoted by $\overline{X}(T)$). These measures are important for measuring the quality of an estimate of the expectation $E(X)$ by an estimator over time. Each measure has its own contribution to the analysis of an estimator. Under the assumption of normality, the variance is important for construction of confidence intervals or testing hypotheses for a point estimate. The bias estimates a certain precision in the estimation of $E(X)$. The MSE estimates the closeness and consistency of an estimate to $E(X)$ by combining the effect of bias and variance. The difference between $E[X(T)]$ and $E[\overline{X}(T)]$ is due to the rate of change of these measures which affects the needed length of a simulation run for estimation of $E(X)$. A graphical approach is taken to visually classify the convergence pattern of these measures into three types i.e. monotonic convergence, non-monotonic convergence and periodic convergence. The empirical results obtained examine the general characteristics of different experimental models, under different conditions. These findings are important as they are useful to analyze the performance of a system in estimation of $E(X)$. We largely focus on the behavior of the models under different system capacities, traffic intensities, number of queues, number of servers

and service-time distribution. The observed behavior of performance measures from the studies provide for the conclusions drawn regarding the system performance. We primarily observed that the measures of interest generally converge, though at different rates. In all cases of our experiments the variance is observed to increase rapidly in the early period of simulation. It reaches its maximum after a while and, then starts decreasing slowly. In contrast, earlier in simulation, the bias is generally large as compared to the variance. Since the formulation of MSE includes a square of the bias, the behavior of MSE in the earlier periods is dominated by the bias. As time passes, the bias approaches 0 rapidly. Hence, in the long run behavior of the MSE is dominated by the variance. The convergence behavior of these measures in complex systems such as $M/M/c$, sequential system and closed queueing network system is observed to be similar to an $M/M/1$ and $M/E_k/1$ queue. However, their rates of convergence differ.

The results of this thesis can be used as guidelines in many simulations studies, even though they may not yet answer specific questions about specific simulations. The results obtained from the experimentation are relevant to queueing models, as most of our experimental models involve queues. The results of the experiments show the following:

- Variance of time average converges to zero in long run, but slowly.
- Three types of convergence patterns are exhibited in the $M/M/1$ system: monotonically decreasing convergence for $X(0) > E(X)$, monotonically increasing convergence for $X(0) < E(X)$ and non-monotonically convergence for $X(0) \approx E(X)$. However, an inventory system exhibited a periodic convergence.
- We investigated an $M/M/1$ system for optimal initial condition. The results for the optimal initial condition for a $M/M/1$ queue confirms Madansky's finding [46] that MSE was optimized by starting the system in empty-and-idle state. The bias is minimized for initial condition close to $E(X)$. The variance is minimized for initial state representing the mode $(X(0) = 0)$ of the system. We investigate the effect of ρ on single-server systems. The times required by the performance measures to converge increase with ρ. Similar results are observed for $M/E_k/1$ system. We investigated the effect of increasing buffer size on single-server systems. The results show that the time required by the performance measures to converge increase with buffer size. We investigated the effect of the number of phases on performance measures of an $M/E_k/1$ system. The variance, bias and MSE decrease with increasing number of phases for $M/E_k/1$ system. For $M/M/1$ systems we also observed that bias dominates for small T, variance dominates for large T.
- In multi-server systems, the optimal initial condition based only on bias is close to the $E(X)$ which is different from optimal initial condition based on variance of time average. Results indicated that optimal initial condition based on MSE in $M/M/c$ model is closest to steady-

state mean. Like the $M/M/1$ system, the time required by the performance measures to converge increase with ρ and system capacity. We investigated the effect of increasing number of server (i.e. c) in $M/M/c$ systems. The variance is observed to be smaller in $M/M/c$ systems as compared to an $M/M/1$ system. The results indicate that the convergence of MSE and variance becomes faster with the increase in number of servers (i.e. c).

- Recall that for two queues in sequence the notation $X1, X2$ symbolizes the number of customers in first queue and in second queue respectively. Likewise, the notation $X1, X2, X3$ represents the number of customers in first, second and in third queue respectively for three queues in sequence. Therefore, an empty-and-idle initial condition for two queues and three queues in sequence is denoted by $X1, X2 = 0, 0$ and $X1, X2, X3 = 0, 0, 0$ respectively. More generally, $X = X1 + X2$ and $X = X1 + X2 + X3$ represent the total number of customers in sequential queueing systems with two and three queues respectively. However, in both the cases if one is concerned with the first queue only then one will consider $X1$ only and ignore $X2$, that is, $X = X1$. If one is concerned with first queue only then one has $X = X1 = 2$ when $X1, X2 = 2, 2$ and $X = X1 = 4$ when $X1, X2 = 4, 0$. The measures of interest are accordingly computed. In addition, in sequential queueing systems where blocking is allowed, a queue is blocked until there a room for a customer in the following queue. In contrast, in sequential queueing systems where blocking is not allowed, a customers leaves the system when the following queue is full.

  - In **sequential queueing systems with blocking**, the optimal initial condition ($X = X1 + X2$ for two queues in sequence and $X = X1 + X2 + X3$ for three queues in sequence) based only on bias is close to the $E(X)$, whereas the optimal initial condition based only on variance is the empty-and-idle condition. Results indicated that optimal initial condition based on MSE is the *mode* which is also close to the steady-state mean. The time required by the performance measures to converge increase with ρ, buffer size and number of queues.

  - We further investigated the behaviour of the **first queue** only for sequential queueing system **with blocking**. The optimal initial condition for the first queue based only on the bias is close to $E(X)$, whereas the optimal initial condition based only on variance is *mode*. The optimal initial condition for the first queue based on MSE is found to be the *median* which is more close to $E(X)$ than the *mode*. The time required by the performance measures to converge increase with ρ and buffer size.

  - In **sequential queueing systems without blocking**, the optimal initial condition ($X = X1 + X2$ for two queues in sequence and $X = X1 + X2 + X3$ for three queues in sequence) based only on bias is close to the $E(X)$, whereas the optimal initial condition

based on variance is the *mode* of the system. Results indicated that optimal initial condition based on MSE is close to steady-state mean. The time required by the performance measures to converge increase with ρ, buffer size and number of queues.

- The selected inventory system represents an almost periodic system where the value of N serves as a means of measuring the periodicity. In almost periodic systems, the time required by performance measures to converge increases with periodicity.

- The selected closed queueing system represents an almost decomposable system. To recall, we are interested only in the number of customers in the first and second queue combined, i.e, $X = X1 + X2$. In the first case (i.e., almost decomposable system) where the rates of going back and forth between queue 3 and other queues are same (i.e. $\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$), the degree of decomposability increases for decreasing rates. Therefore, the system change toward equilibrium also become slower in the same proportion (because $\lambda_{13} = \lambda_{31} = \lambda_{23} = \lambda_{32}$) of decreasing transition rates. Hence, the time required by performance measures to converge increases with degree of decomposability as examined in first case.

  In the second case (i.e., semi-decomposable system) the rates for arriving in queue 3 from other two queues are same (i.e. $\lambda_{13} = \lambda_{23}$) and only the rates for leaving queue 3 are varied to change the degree of decomposability. The customers in this case will arrive in queue 3 at a specified rate, but will leave queue 3 at the same or lower rate. This increases the decomposability of the system and decreases the $E(X)$ ($X = X1 + X2$) of the system. The results indicate that bias takes longer to converge with the increase in degree of decomposability. However, the curves for variance and MSE converge faster for high degree of decomposability (i.e. low rates) because the $E(X)$ is low.

  The results in both the cases indicate that variance is more important than bias. Also, the results of first case indicate that an intial condition close to $E(X)$ is optimal when the part of the system of concern is favourably affected by decomposability.

The contributions of this research in view of set-out objectives are summarized are as following

- In $M/M/1$ and $M/E_k/1$ systems, the optimal initial condition is independent of the service-time distribution. However, it depends on the measure of interest. The variance increases with the difference between mode and $E(X)$, whereas the bias increases with the difference between $E(X)$ and $X(0)$. Therefore, the time taken by measures to converge increase with ρ and, decreases for increasing number of phases.

- In some cases (e.g., see Figure 5.4), the initial condition does not matter in the long run.

- The behaviour of a sequential queueing system with blocking also represents the behaviour of an $M/M/1$ queue, whereas the system without blocking as a whole does not behave like

an $M/M/1$ queue.

- In $M/M/c$ system, the importance of the bias increases with c.
- One needs to pay more attention to high traffic intensities, due to the fact that highly utilized systems take longer to converge.
- The simulation run increases with number of states in almost periodic systems.
- The simulation run length increases with degree of decomposability in almost decomposable systems as observed in first case. However, considering only the degree of decomposability is not sufficient and, one needs to pay attention to the part of a system that will be favorably or unfavorably affected.

6.3 Possible Future Research Studies

This thesis gives useful insight on the behavior of the variance, the bias and the MSE for queueing and non-queueing systems, depending on the structural properties of a system. Further research should compare the estimates obtained using different methods such as batch means method, independent replications method and a single long run to find a preferred method for estimation.

In almost periodic systems, we increased periodicity by increasing N which also increased the number of states in system. The effect of periodicty should be further investigated.

Despite the fact that this study investigates challenging systems like multi-sever systems, sequential systems and closed queueing network system, this study should be extended to more complex stochastic systems including non-Markovian systems before drawing a general conclusion on the behavior of performance measures. Building a model for more complex systems may be a bottleneck experienced in this case.

Another research direction is to investigate a problem using actual simulation methods (e.g. Monte Carlo Simulation) and making consequential comparisons to the analytical results. This will enable one to apply analytical methods more confidently.

Bibliography

[1] P. A. Aad, V. Moorsel, L. A. Kant, and W. H. Sanders. Computation of the Asymptotic Bias and Variance for Simulation of Markov Reward Models. In *SS '96: Proceedings of the 29th Annual Simulation Symposium (SS '96)*, page 173, Washington, DC, USA, 1996. IEEE Computer Society.

[2] C. Alexopoulos and D. Goldsman. To Batch or Not to Batch? *ACM Trans. Model. Comput. Simul.*, 14(1):76–114, 2004.

[3] J. Banks, J. S. Carson II, and B. L. Nelson. *Discrete-Event System Simulation*. Prentice Hall, New Jersey, 1996.

[4] N. Blomqvist. Serial Correlation in a Simple Dam Process. *Operations Research*, 21(4):966–973, (Jul - Aug) 1973.

[5] C. R. Cash, B. L. Nelson, D. G. Dippold, J. M. Long, and W. P. Pollard. Evaluation of Tests for Initial-Condition Bias. In *WSC '92: Proceedings of the 24th Conference on Winter Simulation*, pages 577–585, New York, NY, USA, 1992. ACM Press.

[6] R. C. H. Cheng. A Note on the Effect of Initial Conditions on a Simulation Run. *Operational Research Quarterly*, 27(2, Part 2):467–470, (1976).

[7] C. Chien. Batch Size Selection for The Batch Means Method. In *WSC '94: Proceedings of the 26th Conference on Winter Simulation*, pages 345–352, San Diego, CA, USA, 1994. Society for Computer Simulation International.

[8] R.W. Conway. Some Tactical Problems in Digital Simulation. *Management Science*, 10(1):47–61, October 1963.

[9] P. J. Courtois. *Decomposability: Queueing And Computer Systems Applications*. New York: Academic Press, 1977.

[10] D. J. Daley. The Correlation Structure of Output Process of Some Single Server Queueing Systems. *The Annals of Mathematical Statistics*, 39(3):1007–1019, June 1968.

[11] D.J. Daley. Monte Carlo Estimation of the Mean Queue Size in a Stationary GI/M/1 Queue. *Operations Research*, 16(5):1002–1005, Sep - Oct 1968.

[12] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 4(2):209–223, April 1996.

[13] G. S. Fishman. Bias Considerations in Simulation Experiments. *Operations Research*, 20(4):785–790, (Jul. - Aug., 1972).

[14] A. V. Gafarian and C. J. Ancker Jr. Mean Value Estimation from Digital Computer Simulation. *Operations Research*, 14(1):25–44, (Jan. - Feb., 1966).

[15] A. V. Gafarian, C. J. Ancker Jr, and T. Morisaku. The Problem of the Initial Transient in Digital Computer Simulation. In *WSC '76: Proceedings of the 76 Bicentennial Conference on Winter Simulation*, pages 49–51. Winter Simulation Conference, 1976.

[16] A.V. Gafarian and C.J. Ancker Jr. Mean Value Estimation from Digital Computer Simulation. *Operations Research*, 14(1):25–44, Jan - Feb 1966.

[17] P. W. Glynn and D. L. Iglehart. A New Initial Bias Deletion Rule. In *WSC '87: Proceedings of the 19th Conference on Winter Simulation*, pages 318–319, New York, NY, USA, 1987. ACM Press.

[18] D. Goldsman and B. W. Schmeiser. Computational Efficiency of Batching Methods. In *WSC '97: Proceedings of the 29th Conference on Winter Simulation*, pages 202–207, New York, NY, USA, 1997. ACM Press.

[19] W. K. Grassmann. Relation Between Bias and Variance of Time Averages in Simulation. Unpublished.

[20] W. K. Grassmann. Transient and Equilibrium Probabilities and Their Interpretation. Unpublished.

[21] W. K. Grassmann. Transient Solutions in Markovian Queues. *Computers and Operations Research*, 4:47–53, 1977.

[22] W. K. Grassmann. Initial Bias and Estimation Error in Discrete Event Simulation. In *WSC '82: Proceedings of the 14th Conference on Winter Simulation*, pages 377–384. Winter Simulation Conference, 1982.

[23] W. K. Grassmann. Markov Modelling. *WSC '83: Proceedings of the 15th conference on Winter Simulation*, pages 613–619, 1983.

[24] W. K. Grassmann. The Factorization of Queueing Equations and Their Interpretation. *J. Opns. Res. Soc.*, 30(11):132–139, 1987.

[25] W. K. Grassmann. Finding Transient Solutions in Markovian Event Systems through Randomization. *Numerical Solution of Markov Chains*, pages 357–371, 1991. W.J. Stewart, editor. Marcel Dekker.

[26] W. K. Grassmann. Lecture notes in Computer Science 858. Unpublished, 2006.

[27] W. K. Grassmann. Means and Variances of Time Averages in Markovian Environments. *European Journal of Operational Research*, 31(1):132–139, Jul 1987.

[28] W. K. Grassmann and M.L. Chaudhry. A New Method to Solve Steady State Queueing Equations. *Naval Res. Log. Quart.*, 29(3):461–473, 1982.

[29] W. K. Grassmann and J. Luo. Simulating Markov-Reward Processes With Rare Events. *ACM Trans. Model. Comput. Simul.*, 15(2):138–154, 2005.

[30] W. K. Grassmann and D. A. Stanford. *Matrix Analytic Methods. In Computational Probability.* Kluwer Academic Publishers, Norwell, MA, 2000.

[31] W. K. Grassmann, M. I. Taksar, and D. P. Heyman. Regenerative Analysis and Steady state Distribution for Markov chains. *Operations Research*, 33(5):1107–1116, (Sep. - Oct., 1985).

[32] W.K. Grassmann. *Stochastic Systems for Management.* North Holland, New York, 1981.

[33] W.K. Grassmann. *Computational Probability: Challenges and Limitations. In Computational Probability.* Kluwer Publishers, 2000.

[34] D. Gross and C.M. Harris. *Fundamentals of Queueing Theory.* John Wiley and Sons, 3rd edition, 1998. pages 76-77.

[35] G. Hadley and T.M. Whitin. *Analysis of Inventory Systems.* Prentice-Hall, Eaglewood Cliffs, NJ, 1963.

[36] B. Hajek and L. He. On Variations of Queue Response for Inputs with the Same Mean and Autocorrelation Function. *IEEE/ACM Transactions on Networking*, 6(5):588–598, October 1998.

[37] G. B. Hazen and A. A. B. Pritsker. Formulas for the Variance of the Sample Mean in Finite State Markov Processes. *Journal of Statistical Computation and Simulation*, 12:25–50, 1980.

[38] Hillier and Lieberman. *Introduction to Operations Research.* Tata McGraw-Hill Publishing Company Limited, New Delhi, 2001.

[39] J.H. Jenkins. On the Correlation Structure of the Process of the $M/E_\lambda/1$ Queue. *Journal of the Royal Statistical Society Series B-Statistical Methodology*, 28(2):336–344, 1966.

[40] K. Kang and D. Goldsman. The Correlation Between Mean and Variance Estimators. In *Proceedings of the 1985 Winter Simulation Conference*, pages 211–216, 1985.

[41] E. Kreyszig. *Advanced Engineering Mathematics*. John Wiley & Sons, 2004.

[42] A. M. Law and W. D. Kelton. Confidence Intervals for Steady-State Simulations, II: A Survey of Sequential Procedures. *Management Science*, 28(5):550–562, May 1982.

[43] A.M. Law and W.D. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill New York, 2000. Third Edition.

[44] J. Li. Correlated Arrival Traffic Models. Master's thesis, Department of Computer Science, University of Saskatchewan, 1998.

[45] S. Lock. Some Experimental Designs for Determining Run-Lengths in Simulation. Master's thesis, Department of Computer Science, University of Saskatchewan, 1988.

[46] A. Madansky. Optimal Initial Conditions for a Simulation Problem. *Operations Research*, 24:572–577, 1976.

[47] M. A. Marsan, G. Conte, and G. Balbo. A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Trans. Comput. Syst.*, 2(2):93–122, 1984.

[48] P. M. Morse. *Queues, Inventories and Maintenance*. John Wiley & Sons, Inc., 1967.

[49] P. M. Morse. Stochastic Properties of Waiting Lines. *Journal of the Operations Research Society of America*, 3(3):255–261, (Aug., 1955).

[50] D. H. Ockerman and D. Goldsman. The Impact of Transients on Simulation Variance Estimators. In *Proceedings of the 1997 Winter Simulation Conference*, pages 234–239, 1997.

[51] A. R. Odoni and E. Roth. An Empirical Investigation of the Transient Behaviour of Stationary Queueing Systems. *Operations Research*, 31:432–455, 1983.

[52] O. A. Oni. Initial Bias in the Simulation of Markovian Event Systems. Master's thesis, Department of Computer Science, University of Saskatchewan, 2003.

[53] E. Parzen. *Stochastic Processes*. Society for Industrial and Applied Mathematics, Philadelphia PA USA, 1999.

[54] B. Plateau and K. Atif. A Methodology for Solving Markov Models of Parallel Systems. *IEEE Journal on Software Engineering*, 17(10):1093–1108, Aug 1991.

[55] A. A. B. Pritsker and C. D. Pegden. *Introduction to Simulation and SLAM.* John Wiley and Sons, 1979.

[56] J. F. Reynolds. Some Theorems on the Transient Covariance of Markov Chains. *Journal of Applied Probability*, 9:214–218, 1972.

[57] J. F. Reynolds. Covariance Structure of Queues and Related Processes - Survey of Recent Work. *Advances in Applied Probability*, 7(2):383–415, 1975.

[58] S. Robinson. A Statistical Process Control Approach for Estimating the Warm-up Period. In *Proceedings of the 2002 Winter Simulation Conference*, pages 439–446, 2002.

[59] B. Schmeiser and W. T. Song. Correlation Among Estimators of the Variance of the Sample Mean. In *WSC '87: Proceedings of the 19th Conference on Winter Simulation*, pages 309–317, New York, NY, USA, 1987. ACM Press.

[60] B. W. Schmeiser and W. T. Song. Batching Methods in Simulation Output Analysis: What We Know and What We Don't. In *WSC '96: Proceedings of the 28th Conference on Winter Simulation*, pages 122–127, New York, NY, USA, 1996. ACM Press.

[61] L. W. Schruben. *Graphical Simulation Modeling and Analysis.* Boyd & Fraser Publishing Company, 1995.

[62] R. E. Shannon. *Systems Simulation the Art and Science.* Prentice-Hall, Inc., 1975.

[63] M. Sherman. On Batch Means in the Simulation and Statistics Communities. In *Proceedings of the 1995 Winter Simulation Conference*, pages 297–301, 1995.

[64] H. A. Simon and A. Ando. Aggregation of Variables in Dynamic Systems. *Econometrica*, 29(2):111–138, Apr., 1961.

[65] D. A. Stanford, B. Pagurek, and C. M. Woodside. Optimal Prediction of Queue Lengths and Delays in GI/M/m Multiserver Queues. *Operations Research*, 32(4):809–817, (Jul. - Aug., 1984).

[66] D. A. Stanford, B. Pagurek, and C. M. Woodside. Optimal Prediction of Times and Queue Lengths in the M/G/1 Queue. *The Journal of the Operational Research Society*, 39(6):585–593, (Jun., 1988).

[67] D. A. Stanford, B. Pagurek, and C. M. Woodside. Optimal Prediction of Times and Queue Lengths in the GI/M/1 Queue. *Operations Research*, 31(2):322–337, (Mar. - Apr., 1983).

[68] W. Whitt. Simulation Run Length Planning. In *WSC '89: Proceedings of the 21st Conference on Winter Simulation*, pages 106–112, New York, NY, USA, 1989. ACM Press.

[69] W. Whitt. The Efficiency of One Long Run Versus Independent Replication in Steady-State Simulation. *Manage. Sci.*, 37(6):645–666, 1991.

[70] J. R. Wilson and A. A. B. Pritsker. A Survey of Research on the Simulation Startup Problem. *Simulation*, 31:55–58, 1978.

[71] J. R. Wilson and A. A. B. Pritsker. Evaluation of Startup Policies in Simulation Experiments. *Simulation*, 31:79–89, 1978.

[72] C. M. Woodside, B. Pagurek, and G. F. Newell. A Diffusion Approximation for Correlation in Queues. *J. Appl. Prob*, 17:1033–1047, 1980.

[73] Y. Yeh and B. W. Schmeiser. On the MSE Robustness of Batching Estimators. In *WSC '01: Proceedings of the 33nd Conference on Winter Simulation*, pages 344–347, Washington, DC, USA, 2001. IEEE Computer Society.